Software Giveaway with Purchase of this Magazine
See bound-in card behind front cover

HOME COMPUTER™ magazine

FOCUSING EXCLUSIVELY ON ●APPLE ●COMMODORE ●IBM ●TEXAS INSTRUMENTS

Vol.4 No.2

$3.00 in USA
$3.50 in Canada

Bouncing Ball Graphics for the Apple
Pixel Tricks for the Commodore 64
Multi-Color Secrets of the TI-99/4A
VIC-20 Image Editor
Animation on the IBM PCjr

FROGO for LOGOphiles

Your Electronic Home Secretary

Computer Gaming:
Hungry Cannibals &
Board Warfare

Continuing 99'er Magazine's Complete Coverage of the TI-99/4A

HOME COMPUTER GRAPHICS

0 12509 06
477465

FREE Bound-in Home Computer Digest

REMEMBER.

ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

All the hits your computer is missing.

If you thought you'd never find fun games for your hardworking home computer, happy days are here. Because now ATARISOFT™ has all the great hits…Pac-Man,[1] Donkey Kong[2] by Nintendo;[2] Centipede,™ Defender;[3] Joust;[3] Jungle Hunt;[4] Moon Patrol;[3] Pole Position;[5] Galaxian;[1] Ms. Pac-Man;[1] and Battlezone.™

And we've got them for all the hit computers …Apple, IBM, Commodore 64, Vic-20, Colecovision;* and TI 99/4A. We've got Pac-Man, Centipede and Defender for Intellivision too.

So dust off your joystick and ask your dealer for all the ATARISOFT hits. It's the software your hardware's been waiting for.

ATARISOFT.™

All the hits your computer is missing.

HOME COMPUTER magazine

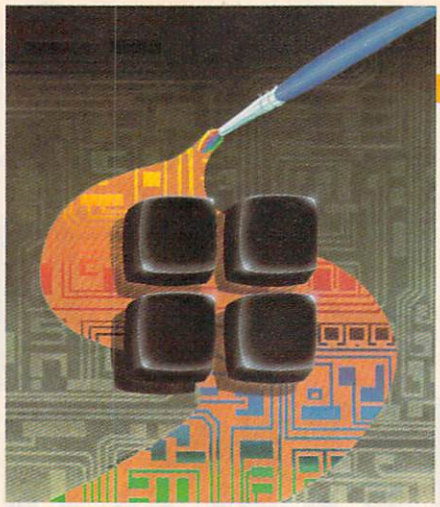
Home Computer Magazine (ISSN 0747-055X) is published monthly by Emerald Valley Publishing Co., P.O. Box 5537, Eugene, OR 97405. The editorial office is located at 1500 Valley River Drive, Suite 250, Eugene, OR 97401 (Tel. 503-485-8796). Subscription rates in U.S. and its possessions are $25 for one year, $45 for two years, and $63 for three years. In Canada and Mexico add $7 per year. Other foreign countries $43 for one year surface mail. Inquire for air delivery. Single copy price in U.S. and its possessions is $3.00, and $3.50 in Canada and Mexico. Foreign subscription payment should be in United States funds drawn on a U.S. bank. Second-class postage paid at Eugene, OR 97401, and Columbia, MO 65201.

POSTMASTER: Send all address changes to **Home Computer Magazine**, P. O. Box 5537, Eugene, OR 97405. Subscribers should send all correspondence about subscriptions to above address.

Address all editorial correspondence to the Editor at **Home Computer Magazine**, 1500 Valley River Drive, Suite 250, Eugene, OR 97401. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage and self-addressed envelope. Not responsible for lost manuscripts, photos, or program media. Opinions expressed by the authors are not necessarily those of **Home Computer Magazine**. All mail directed to the "Letters to the Editor" column will be treated as unconditionally assigned for publication, copyright purposes, and use in any other publication or brochure, and are subject to **Home Computer Magazine's** unrestricted right to edit and comment. **Home Computer Magazine** assumes no liability for errors in articles or advertisements. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by **Home Computer Magazine** or the publisher unless explicitly stated.

Membership in ABC applied for and pending.

Publisher/Editor-in-Chief Gary M. Kaplan
Executive Editor David G. Brader
Managing Editor Walter Hego
Production Editor Sharyn Lyon
Sr. Technical Editor William Balthrop
Technical Editors
Peter Bloch, Robert Keller, G.R. Michaels, Patricia Swift, John Thrasher, Roger Wood
Feature & Review Editor Greg Roberts
Manuscript Editors
Dana Campbell, Joan Killough-Miller, Erin O'Connor
Contributing Editors
Michael Brownsworth, William M. Goodman, Henry Gorman, Jr., Richard Haller, S. T. Holl, Roger Kirchner, Ted Martino
Editorial Assistants
Deborah Amity, Rhea Grundy
Production Manager Norman Winney, Jr.
Creative Director Gei-Lei Gom
Photography Brion Marquez
Design Coordinator Eric Martin
Production & Design
Larry Fisk, Larenzo, Barbara Mickelson, Mohsen Zarrabi
Typesetting
Kathy Garcia, Louisa Hooven, Julienne Laabs
Administration & Finance
Pat Kaplan, Bob Karau
Accounting
Mark Andersen, Maureen Connolly, Dorothy Dwyer
Customer Relations **Tel. (503) 341-1029**
Sharon Hinshaw, Denise McGuire, Susan Reed, Kaine Thompson
Dealer Sales & Distribution **Tel. (503) 341-1032**
Ann Crenshaw, Michael Flagg, Paula Hoiland
Advertising **Tel. (503) 341-1049, 1050, 1053**
Irene Alderman, Lynn Ballheim, Charisse Loritz, Carol O'Brien
Main Switchboard **Tel. (503) 485-8796**



Outside HCM

This month's colorful cover painting depicts the rainbow of graphics potential awaiting you within a home computer's integrated circuits. All it takes to release this silicon palette is a little help from your friend, *Home Computer Magazine*. Just press one of the magical graphic keys and begin our visually exciting journey into the realm of the computer artist.

Inside HCM

Spring—and the Earth takes on a fresh new look as its multi-hued vegetation blossoms into a veritable rainbow of color. This month, we bring the same breath of fresh air to your home computer—transforming it from a monochrome information processor to a colorful showcase of lively computer graphics.

Inside this issue of *Home Computer Magazine* an impressive portfolio awaits. You'll find graphics techniques, tutorials, and product reviews for a wide range of artistic talent—from budding Renoirs to the fingerpaint set.

There's no better way to launch a discussion of computer graphics than with sprites—those marvelous video imps that dart across the screens of both TI-99/4A and Commodore 64 home computers. In *Double Your Pleasure, Double Your Fun* we show users of these machines how to expand their high-resolution graphics repertoire with some spritely never-before-revealed tricks of the programming trade.

Apple users haven't been left out of the picture either. Anyone can learn Apple graphics programming with *Follow the Bouncing Ball: An Apple Graphics Tutorial*. In this issue we also present Part 2 of *3D-IIe: Apple Graphics in Three Dimensions*.

Fans of the VIC-20 don't despair! Nowhere is it written that users of a 5K machine must be excluded from this issue's artistic visions. You can try out some major motion of your own—but first tune in to *Building Your Character: A Graphics Editor for the VIC-20*.

And to round out our theme, we present you with several reviews of commercial graphics packages for all the home machines we cover—utilities that can save you a lot of time, and make design work with the computer that much easier.

For those days when spring showers are busy creating summer flowers, we've got several games ready to key-in and play—computer entertainment such as *Cannibals, Sea of States, Tablut,* and *San Francisco Tourist*. Nowhere else will you find as varied a magazine mix of arcade action, educational benefits, and logical skill development.

If number-crunching is your game, we've got just the menu. Youngsters who are getting down to basics can try *Elementary Addition and Subtraction,* while those with more advanced math skills can sink their teeth into *Matrix Muncher,* an easy way to solve simultaneous equations with numerous unknowns.

To help our readers who are more interested in learning music than math, there's *Musical Mystery Words*—where players must identify the letter names of musical notes displayed on the screen and played through their TV. These notes, in turn, spell out the sought-after mystery word. A sound way to learn music, improve spelling skills, and have fun!

Getting the sound out of your home computer doesn't have to remain a mystery any longer when *Jr. Sounds Off* in this issue. And lest the sound of music distract you from your annual spring file cleaning, our *Electronic Home Secretary* offers some valuable organizational help. So get it all together with this versatile program that makes directories, lists, and inventories. Just don't expect it to make coffee. . .

LOGO lovers can be list makers too—after reading *Files in LOGO*. The article takes you through an interactive graphics program that includes data file access like that in BASIC. On the lighter side of LOGO, there's *FROGO*—a frog-leaping game that will help you to understand the rows and columns, speed, direction, and x and y coordinates of the LOGO language. And rounding off this month's coverage of turtles, frogs, and the like, we learn from an outspoken father-daughter team just how effectively *LOGO Spans the Generation Gap*.

Until next month, have fun reading, learning and RUNing HCM

Your Guide to Typing in Programs from *HCM*

Within these pages is a software bonanza: entertainment, education, home and business applications, utilities, and tutorials—just for you. All you need to do is type them into your computer. *HCM* has taken most of the strain out of this process with:

- Typeset listings with numbers in boldface type.
- A bold, double vertical bar separating the line numbers from the program statements in BASIC listings.
- A vertical background grid to aid entry of the spaces.

Look at the Key-in Reference (Figure 1 below) to see how each character actually appears in the listing. By checking any questionable characters with the Key-in Reference, you can reduce errors to a minimum.

Figure 1: KEY-IN-REFERENCE

Before You Begin

Since *HCM* publishes programs for several different computers, the first thing you should do is make sure that you are looking at the listing designed for your machine. If, for example, you have an Apple IIe, make sure you look for the following black bar above the listing: APPLE II Series The computer model name will likewise appear on each subsequent page of each listing, so always look for the name before you begin typing from a new page of listings.

Before you begin typing in the program, you will want to set up a system to save your program. Whether you are using a cassette or diskette storage system, now is the time to be certain it is properly connected, powered up, and loaded with a blank cassette or an initialized disk. As you type in your program, you should get in the habit of saving your work after every twenty or so lines.

One of the most common errors in entering a listing is typing one symbol for another. These transpositions include substituting the letter O for the number 0, the letter I for the number 1, the letter S for the $, and the uppercase B for the number 8. This last error is especially likely when working in hexadecimal numbers which are composed of 0-9 and the uppercase letters A-F.

The listings in *HCM* are always the same number of characters wide, but, the number of characters put on any line of the video display will vary from computer to computer. Don't try to make your listing *look like* the type-set listing—instead make sure you key in the listings character for character and space for space.

A Special Note on Listings for the C-64 and VIC-20

Commodore uses more than 90 special symbols to represent various keyboard operations: for instance, the symbol [♥] in a program represents the operation of holding down the [SHIFT] key and pressing the key which has CLR on its upper half (second key from the right on the top row). This operation clears the screen.

Rather than reproducing these symbols, *HCM*'s listings include key-stroke instructions between two hands with pointing fingers. For example, when you find ☛ SHIFT CLR ☚ in an *HCM* listing, you will know to hold down the [SHIFT] key and press the key with CLR on it.

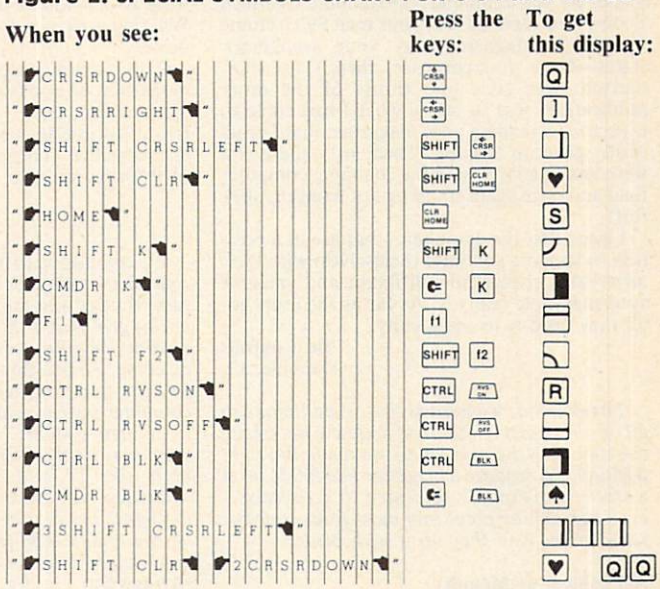
A number is included if you need to repeat the operation: ☛ 8SHIFT CRSRLEFT ☚ tells you to hold the [SHIFT] key down and press the cursor left key (on the bottom right of the keyboard) eight times.

When you come to the hand symbols, remember:

- Each operation is enclosed in its own set of hand symbols.
- If any key action requires you to press two keys, press the control key or the Commodore key or the shift key. *first* and hold it down before pressing the second key.
- Not to enter spaces or anything else *within* a set of hands.
- Everything between a pair of hand symbols is set in a different typeface.

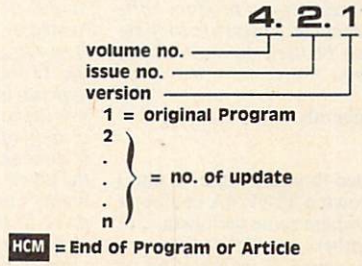
In Figure 2 below, we have included a chart showing you a representative sample of the symbols that appear when you use keystrokes enclosed by the hand symbols.

Figure 2: SPECIAL SYMBOLS CHART FOR C-64 and VIC 20



Program Identification

Each program header (the first few lines of the program) contains information giving the language the program is written in (e.g., TI Extended BASIC, Applesoft, etc.) and any special system components that are required (special memory cards, Speech Synthesizer, etc.). The first two digits of the version number tell you in which volume and issue of *HCM* the program *initially* appeared. The third digit of the version number indicates the version of the program. When a program initially appears, in *HCM*, it is version 1. Any subsequent revisions to the program if later published in the magazine or in the software available on magnetic medium from *HCM* will bear a revised version number.



HCM

Letters to the Editor

Happy with 2 Machines

Dear Sir:

It is with a sigh of relief that I write this letter. I was afraid that since the pull-out of Texas Instruments from the home computer market, you would have to put your magazine to rest. How surprised and relieved I was to find your journal alive, well and from all appearances, better than ever.

I began my journey into home computers with a TI-99. Due in large part to your excellent efforts, I was able in a short time to learn much about programming. Several months after my purchase of the 99, a new computer arrived at my place of work: an Apple IIe attached to a pulmonary function machine. Because of this and the fact that TI decided to fold its home computer operations, I recently purchased an Apple-compatible Franklin Ace 1200.

When I began to search for a new journal to match my new computer, the standard by which I compared them all was your own 99'er Home Computer Magazine. As your trademark states—Once you compare, there's no comparison. The style and thrust of the other publications was so below what I had come to expect from reading your magazine, that I practically gave up looking. Then, lo!—there you were again, adapting to the changing computer field and once again standing out from the slick fluff.

I hope that the companies that are in a position to support you (and themselves) with their advertising realize how different and valuable your magazine really is. As far as the users go, all they need is to see a copy.

Sid Lunford
Danville, Ill.

Thanks, Sid, we needed that. And thanks to all of our readers both loyal and new for taking the time to write and tell us we made the right decision. To produce a magazine like HCM takes a staff that is totally dedicated to excellence—even beyond the point that most folks consider reasonable. And they do it with pride.

A Problem of Memory

Dear Sir:

For Christmas this past year I bought the family a Commodore VIC-20 home computer. Now I am wondering if I made a mistake. It seems that there is very little in the way of support for this toy while there is lots of support for the more expensive Commodore C-64 home computer. Did I waste my money or am I looking in the wrong places?

Arnold Bainbridge
Elk Grove Village, Ill.

The VIC-20 was designed with only 5K of RAM. The fact that it is extremely difficult to write very sophisticated programs in so limited a memory area, has caused most current third-party Commodore developers to produce software for the bigger machine. If you are looking for programs that can be used on the VIC-20, HCM is now your best source.

A Question of Peripherals

Dear Sir:

I recently subscribed to your magazine and I like it very much. I own a TI-99/4A computer and would like to purchase some peripherals for it. The thing that bothers me is that there appears to be no good source of information on what peripherals are compatible with my computer. I can't even find any technical information which would allow me to figure this out for myself. I would like to see you write a series of articles in laymen's language on the differences in interfaces and how they work.

Perhaps you could also help me with some questions I have concerning my computer. I would like to replace my TI computer with an IBM PC later on, and would like to buy my peripherals with this in mind. Specific questions I have are as follows: Can I use an IBM color monitor directly with my TI-99/4A? Will the TI Impact printer work with an IBM PC? Will the IBM printer work with my TI? If not, what has to be done to convert one or the other? An ad in your magazine says they are both Epson MX80s, but are they exact copies of one another? Can I put one double-density double-sided disk drive (or perhaps two half-height drives) in my TI Peripheral Expansion System box and have them work properly? Will the TI electronics work with a double-density double-sided drive? Will single- density disks run on a double-density drive?

I have not only looked extensively for magazines or books which provide this type of information, but I have asked many of my friends for the answers,and they don't know the answers either. They would like to have this type of information also.

John Paulson, jr.
Endwell, NY

We are aware of the many readers that are changing systems for one reason or another. The concerns that you express are shared by everyone in this group and should also be considered by anyone buying new peripherals for their machine. To boil down and paraphrase your letter, the question becomes, "If I upgrade my computer system, which peripherals can I re-use in my new system?" First, any peripheral that interfaces via the RS232 standard (supported on both the TI and IBM machines) will be something that can be kept. Generally, printers, modems, and plotters fall in this area. Certainly, the TI and IBM (Epson MX80) printers will work interchangeably (although internal switch settings may require changing due to software differences).

Any monitor that uses a composite color input signal will work with different computers that have this video output. The TI color monitor, for instance, works well with the IBM Pjr. (Both computers have a composite color output.) The IBM PC color monitor, however, is not a composite—but rather the RGB type—so it will not work with your 99/4A.

Your questions on disk drives cannot be answered quite so easily. Both the TI and IBM machines can utilize double-sided disk drives (except for the early model TI systems with stand-alone disk controllers that connected to the right side of the console). The TI disk drive controller, however, does not support double-density recording while the IBM does. You may connect a double-density drive to the TI but it will be treated by the controller as a single-density drive. Regarding the installation of two disk drives in the TI Peripheral Expansion System box, you may run into power supply problems. The system was designed to support the power requirements of only one standard disk drive. Unless specially designed (low-power) disk drives are used, or the power supply is beefed up (by rebuilding with higher-current voltage regulators), stick to one drive in the box.

Your search for a source of information obviously ends with Home Computer Magazine. Watch each issue for techniques and answers to your computing needs. In particular, check out our Home Computer Tech Notes feature.

A Commodore Tip

Dear Sir:

I have a little tip to pass along to the other Commodore C-64 users that read Home Computer Magazine. Maybe you can print it in the Letters to the Editor section of your magazine.

I found that the Commodore BASIC INPUT statement works different from my TI-99/4A and maybe it is causing a problem for others. Take the following example:

10 INPUT "Enter string"; A$

If you just press the return key, the previous value of A$ is retained (from the last time A$ was used in the program!). I thought it would automatically be emptied when the return key was pressed. This can come in handy sometimes, but other times you can clear it in the program as follows:

10 A$="":INPUT "Enter string"; A$

which clears the variable first. This will work on numeric variables too.

Mike Adler
Santee, CA

Thanks for the tip, Mike. I know that will help any TI'ers out there who are trying to learn about the C-64.

Grounds for Trouble

Dear Sir:

I have a problem with my computer that you may be able to help with:

I have a home computer that sometimes, when I am in the middle of a program or a pre-made cassette game, will suddenly just stop and the screen will go blank or produce a mass of unrecognizable characters and sounds. What is causing this?

Ken Marshall
El Toro, CA

Ken, there could be any number of things going wrong to cause your computer to go "bananas" as they say. Some of the more common causes are related to heat and poor connections. First, check under, over, and around the main console to make sure there is good air flow space. Second, with the unit unplugged, clean all the connections to the computer (including plug-in cartridges) by unplugging and replugging them several times followed by carefully reseating each connection. One final suggestion that works in some cases (usually in an older home that uses two-wire AC circuits) is to run a good ground wire from a copper water pipe to the ground connection of the power plug (if you are not sure of how to identify this connection, call an electrician). If all else fails, send the unit to the repair depot closest to your home.

Let Your Fingers Do Debugging

Dear Sir:

I was more than pleased to receive the first issue of your renewed magazine. I just love it and as you had mentioned, there is still enough for us, users of the TI-99/4A.

In Vol 4 #1 of HCM, there is a game in Extended BASIC called *Meltdown*. I typed and checked it over and over again and I am now sure there is no typing mistake in it.

When I RUN it, right at the beginning I get "BAD VALUE IN 490." As I told you, I checked it from the first line to the very last line and there is absolutely no typing error.

I am more than anxious to read the Debugs on Display in next month's magazine to see what the problem was. Thank you for an excellent magazine.

Camille Morin
Verdun Quebec Canada

Fully realizing that you checked your typing extremely carefully, Camille, we recommend that you recheck every character, space, and comma in lines 2020-2110 (the DATA statements). The bad value message means that the HCHAR portion of line 490 has received a value in one of its variables (either R2 or C2) that is outside its range of exceptance.

To discover what the bad value actually is, enter "PRINT C2,R2" and press [ENTER] immediately after you see the Bad value message appear on the screen. Then compare the actual numbers printed for the variables to the allowable values for the HCHAR statement. If you still can't locate the typing error, don't forget that you can get that program and several others ON TAPE™ or ON DISK™ for only $3.95! Something to consider. . .

We Do Peanuts Right!

Dear Sir:

I just purchased a new IBM PCjr with 128K of memory and a disk drive. Now I am reading your Home Computer Magazine to compare it with other magazines which just cover the Peanut. It seems to me that your coverage is not trivial. By this, I mean that I found really useful information in your pages. Especially helpful was the ten-or-so-page article that got inside the Peanut's shell! I immediately went back to my dealer and bought the Cartridge BASIC package and the disk operating system.

Will you be telling readers about software packages that will work on both the PC and the PCjr? I am in a position that I would like to do some word processing at home sometimes. Can the diskettes be used in both machines or will I loose my data if I swap a disk back and forth? Thanks in advance for the help, and for a neat magazine!

J.V. Smith
Royal Oak, MI.

Yes, Mr. Smith, you can swap diskettes between the IBM PC and the PCjr. You should not loose any data as long as you follow the manuals for using the disk operating system. We will be exploring the use of software packages on both the PC and the Junior. In an upcoming issue, we will show you how a word processor (EasyWriter II) works on the little brother of PC. There are many surprises ahead for readers as HCM ventures farther into Big Blue country.

Assembly Needs

Dear Sir:

I have the TI-99/4A computer and it is equipped with the expansion box that has one disk drive, the RS232 card, and the 32K memory expansion. I also have a Banana printer attached.

I want to upgrade it to use the TMS9900 Assembly Language. However, I am confused as to what I need to do this. With the peripherals I now have, would the Assembler/Editor module be sufficient or should I get the Mini-Memory or both?

I would appreciate your advise on how to upgrade my set with the appropriate software and I would also appreciate your advise on where to obtain good instructional material on learning the use of the TMS9900 Assembly Language. I have a pretty good knowledge of TI BASIC & Extended BASIC, but I know very little about assembly language. Also is there any information of how to use the CALL LOAD, CALL INIT, CALL LINK, and CALL PEEK subroutines listed in the TI Extended BASIC manual? The material in the manual is very shallow.

Larry D. Dodgens
Oakwood, GA.

You've got it all, Larry! At the very least, the TI-99/4A must have a disk system with one drive and the memory expansion. Having a printer, although not a system requirement, is a practical neccessity.

The TI Editor/Assembler package, by the way, is much more than just a cartridge. It contains a three-ring binder, a 470 page manual, and two floppy diskettes. The cartridge works with the diskettes and both are required for the Editor/Assembler to function. The manual gives more information on the use of the LOAD, INIT, LINK, and PEEK subroutines in Extended BASIC. The manual is designed as a reference volume, not a tutorial. Assembly language is one of the most difficult languages to learn without a solid background in computer science. There is a need to have a basic understanding of the computer's internal architecture, which often is covered as an ancillary topic when learning about assembly language. It is possible, with great dedication, for a non-computer specialist to learn assembly language, but there is no easy way.

The TI Mini-Memory cartridge may be used with the TI-99/4A by itself (with the possible addition of a cassette player). It too makes use of assembly language, although, on a much more rudimentary level (which for some folks, makes it even harder to master) and its documentation often refers to the Editor/Assembler manual for clarification.

The best information that I have seen on the TMS9900 Assembly Language for beginners is in our own magazine over the past three years. A great place to start is Chapter Five of The Best of 99'er, *over thirty pages of tutorials and programs just on this subject.*

Taping It Right

Dear Sir:

I am going to start taping programs from your articles onto cassettes and am confused as to how to do it. I understand the part about transfering the program from computer to cassette, but I don't know how much space to leave between programs. If I have a digital counter, how many counts do I leave in between programs, and how do I know when a program is at the end before adding that space? Do I have to listen to it out loud and then remember when the computer noise ends? Also, will the computer stop automatically when the program you want read is over, or do you have to stop the tape recorder manually? I am just wondering if the computer would otherwise read all the programs on the cassette before stopping. Also, approximately how many programs (in line length) will fit on a 10-minute tape? A 30-minute tape? A

Continued on p. 32

HCM Review Criteria

Each month, *HCM* reviews software packages for the IBM PC and PCjr, Apple II, II+ and IIe,TI- 99/4A, and Commodore 64 and VIC-20 computers.These reviews take a detailed look at the quality of commercially available third-party software for these home computers.

At the beginning of each review, a review-at-a-glance box provides the user with an instant assessment of the program. Each software item will be evaluated, where relevant, with the criteria below.

- **Performance**—how well the activity responds to the player's commands; how well the sound effects, music, or speech are integrated with the software.
- **Documentation**—the quality of the printed matter that comes with the software: whether the instructions are clear and comprehensive; whether the machine configuration requirements are spelled out. Information such as how to load the program, use the keyboard, and restart the activity contributes to the documentation rating, as do tips on performance peculiarities.
- **Engrossment**—whether the game or activity has that intangible quality that holds the player on the edge of his seat while the hours tick by unnoticed.
- **Ease of Use**—the degree to which a user can interact with the software without outside help; the ease and effectiveness of error-handling features; whether the actual reading level of the activity is appropriate for the suggested audience.

Education-Specific Criteria

Educational software may also be evaluated in the following areas:

- **Concept Presentation**—whether the concepts are presented clearly, in logical order, and in enough depth for the learner to be able to apply the learning to other situations.
- **Rewards**—whether the audio-visual rewards are motivating and whether they are appropriate to the activity.
- **Graphics**—rates the quality of the graphics and whether they enhance or detract from the educational purposes of the activity.

On Screen

By Gary M. Kaplan
Publisher & Editor-in-Chief

T'was early Superbowl Sunday
And all through the house,
Not a creature was stirring,
Not even a mouse.

Came time for the game,
And America tuned in—
To an annual tradition,
Watching the better team win.

Beer, cars, and computers
Were part of that day,
As costly commercials
Entered the fray. . .

Suds for the thirsty,
Major motion for the driven,
Less paper-work to drown us,
And happier times to live in. . .

When all of a sudden
There arose such a clamor,
As the runner in red shorts
Flung her mean hammer.

And shattered the big screen
For all America to see
that 1984 would
A revolutionary year be. . .

"The personal computer industry will experience a "soft upheaval" during the remainder of 1984—rapidly transiting from its seven-year "infancy cycle" to its next seven-year "adolescence cycle."

The now-famous "Big Brother" commercial which marked the debut of Apple's Macintosh computer started me thinking...Not about "a computer for the rest of us" or Apple's alternative to an IBM-dominated world, but to the revolutionary year, 1984, itself.

Revolutions come in all shapes and sizes. They are, however, all characterized by an upheaval from within. The personal computer industry will experience a "soft upheaval" during the remainder of 1984—rapidly transiting from its seven-year "infancy cycle" to its next seven-year "adolescence cycle." The rest of the decade should indeed be an interesting time as we watch the industry grow to maturity.

1984 is the year that the "productivity home computer" finally emerges. With the entrance of the PCjr, the repositioning of the expanded Apple II family, and the announcement of a new Commodore standard (IBM-compatible or otherwise), the home segment of the personal computer market is now being populated by an ever-increasing number of 128K-byte machines. These more expensive systems will probably overtake the installed base of 64K-and-under machines by the end of 1985. This increase in resident memory, coupled with smaller and denser mass storage devices at lower prices, means more sophisticated and easier-to-run programs for home users.

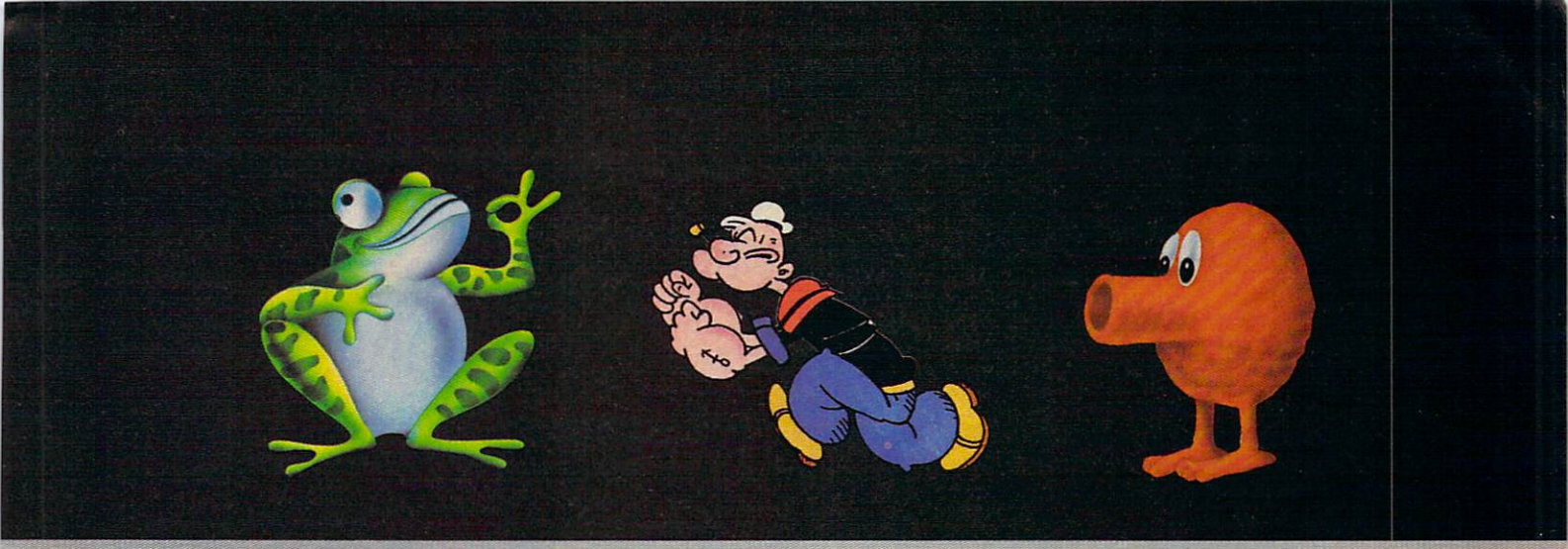
The first phase of software's new adolescence has begun. We've already seen entertainment software virtually explode in 1982, education software in 1983, and now productivity software in 1984. In terms of product development, we're witnessing the abandonment of the "hit title syndrome." Instead of scrambling for the microcomputer rights to blockbuster arcade and movie titles, savvy players are concentrating on building up stables of less flashy software packages that have been designed for longevity—to produce sales month after month, and year after year.

For the most part, changes in the status quo during 1984 are coming from major product *refinements* rather than product *breakthroughs*. We are already seeing a steady stream of new software products that are both more powerful, and easier for beginners to use. These refined packages allow you to "paint" the screen, so what you *see* is what you *get*. Packages with windows, icons, and keyboard-alternates such as the mouse, touch-pad, and light-pen will increasingly find their way into systems used in the home. And as 256K RAM chips become available toward the end of 1984, the memory environment necessary for major software breakthroughs will finally materialize.

In the manufacturing, marketing, and distribution of home computer products, 1984 is already manifesting itself as the year of focussed operations, shake-outs, and consolidations. Many of the firms who used to "do it all"—develop, manufacture, and market their own software lines—have now, by economic necessity, chosen to be either a development house (designing, programming, and documenting the software), or a publisher (handling the manufacturing, marketing, and arranging for distribution).

It is also the pivotal year for product distribution. The industry is undergoing a cross-pollination of delivery channels—with computer specialty stores and chains supplementing the mass merchandisers, and electronic delivery systems gaining in acceptance. And as broader distribution dynamics and a higher ante for playing the game take effect, watch for some smaller firms to band together (sharing marketing and distribution expenses), and for more of the larger firms to buy up their cash-strapped competitors.

For companies who find themselves locked out from one or more of the new distribution channels, mail order advertising in computer publications will be crucial. And we at *Home Computer Magazine* feel proud to be contributing to the growth of this dynamic industry by providing a premium medium for advertisers and consumers alike.



THE STARS OF THE ARCADE SHINE ON TI99/4A.™

Your TI99/4A is a great computer system. And one of the things that's great about it is it can play three of the greatest Arcade Action games ever. Frogger,*™ Popeye,® and Q*bert,™ from Parker Brothers.

The award-winning FROGGER is one of the top selling Arcade Action games of all time. With graphics that are nothing less than ribbitting and game play that gets tougher as you get better.

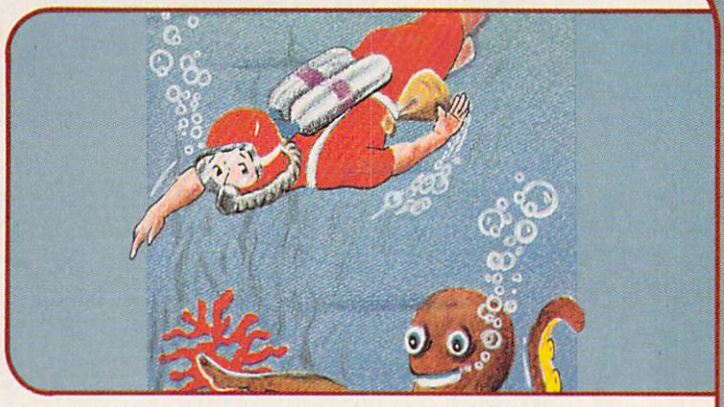
And POPEYE has you running through three screens of non-stop action, where you try to capture Olive Oyl's heart while avoiding untold dangers, including Brutus and the Sea Hag.

As for Q*BERT, he's irresistible. Jumping from cube to cube, trying to avoid an army of nasty critters, he's jumped into the hearts and minds of millions.

Frogger, Popeye, and Q*bert, from Parker Brothers' Arcade Action Series. They make your TI99/4A computer feel as close to the arcade as you can get.

®PARKER BROTHERS

HOME COMPUTER™ magazine



FEATURES

12. Sea of States
Match states and capitals while diving for sunken treasure. *by Joseph Phillips*

17. San Francisco Tourist
A country tourist negotiates some urban curves. *by the HCM Staff*

20. Building Your Character: A Graphics Editor for the VIC-20
Give your characters some personality. *by John P. Thrasher*

20. Quick Pixel Tricks: A Graphics Editor for the C-64
Throw away your paper graphs and create on the screen. *by John P. Thrasher*

34. Follow the Bouncing Ball
On the rebound with graphics fundamentals. *by Randy Wilson*

39. 3D-IIE: Apple Graphics in Three Dimensions, Part II
Create your own 3-D shapes with last month's display program. *by M.D. Brownsworth*

42. Double Your Color, Double Your Fun
Sprites try on the layered look. *by W.K. Balthrop*

51. Musical Mystery Words
No instruments necessary for this music lesson. *by the HCM Staff*

100. Matrix Muncher
Simultaneous equations are a snap, even with numerous unknowns. *by the HCM Staff*

113. Elementary Addition and Subtraction
Basic math skills for youngsters. *by Mark DeWese and HCM staff*

116. IBM Animation: Controlling the Pallet on the PCjr
Drawing circles around the competition. *by W.K. Balthrop*

126. Jr. Sounds Off
Access the Peanut's special sound enhancements. *by W.K. Balthrop*

132. The Electronic Home Secretary
Its applications possibilities are endless. *by Malladi Subbaiah and the HCM Staff*

LOGO TIMES™

81. Files in LOGO
Muller's last challenge is answered. *by Roger Kirchner and the HCM Staff*

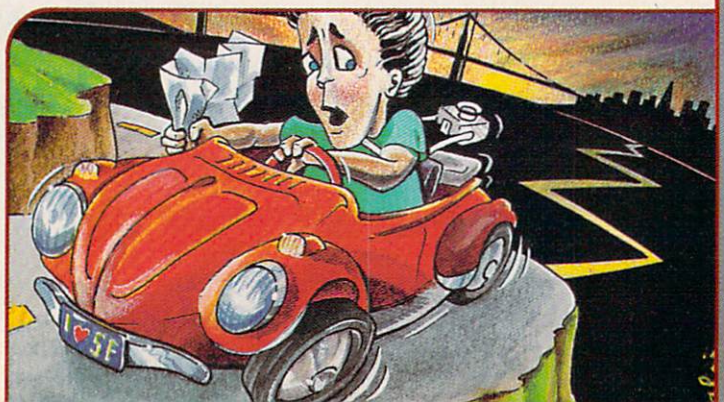
83. LOGO Spans the Generation Gap: A Review of Commodore LOGO
Perspectives from a neophyte teen and a professional programmer. *by Richard and Sarah Haller*

84. FROGO: LOGO Invades the Arcade
FROGO helps with tiles, sprites, and recursive procedures. *by Ted Barnicoat*

PRODUCT REVIEWS

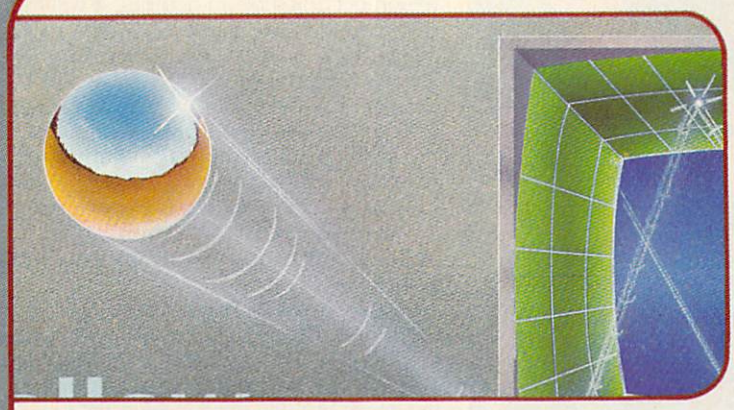
26. Pinball Construction Set
Pinball wizards can literally fine-tune their game. *A Review*

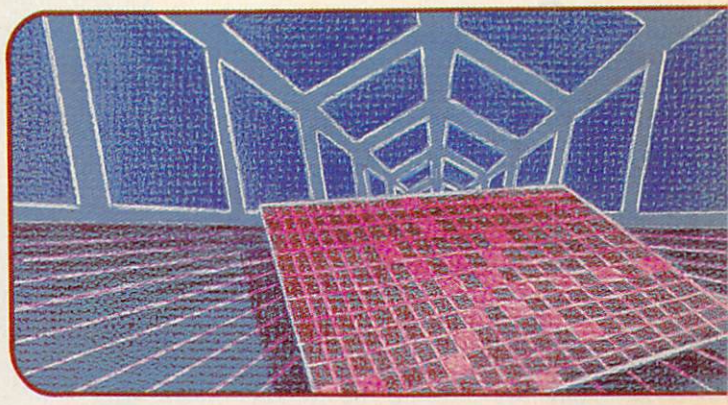
33. Spritemaster 64
Animation brings your imagination to life. *A Review*



CONTENTS

VOLUME 4 NUMBER 2





57. Miner 2049'er — A Review
Trapped in a radioactive mine.

65. Mouser — A Review
A challenge to build a better mousetrap.

66. Attack of the Mutant Camels — A Review
Cosmic cameloids at their best.

67. Computer War — A Review
Save the world from thermonuclear destruction.

68. Castle Wolfenstein — A Review
Steal the Nazis' plans and escape their castle prison.

73. Zork — A Review
Let your mind wander with this classic adventure.

74. Donkey Kong — A Review
Get ready for gorilla warfare—in your home.

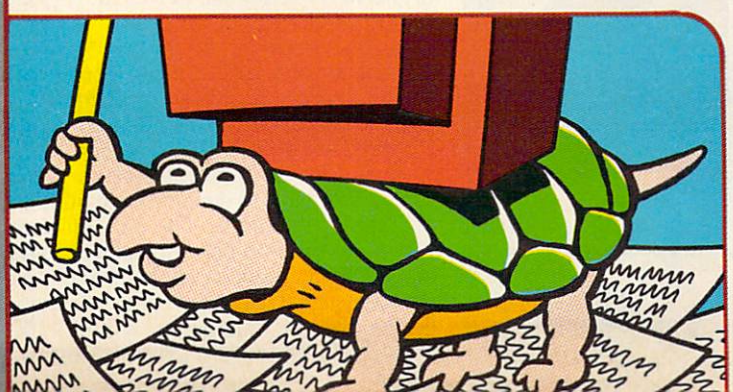
74. Tag Tom and Fire — A Review
Finally, games for preschoolers.

75. Q*bert — A Review
This little guy keeps the action hopping.

76. TI-Maze — A Review
An amazing 3-D game.

97. Animation Creation — A Review
Creating video flip-cards for motion.

115. Adventures in Math — A Review
Who says math drills have to be dull?



122. Spellakazam — A Review
Race through an alphabetical maze of color and letters.

129. Let's Get Graphic: Reviews of Graphics Code Generator and Graphics Designer
Modern day canvas and paintbrush.

154. PowerPad — A Review
Staying in touch with the touch-tablet market.

155. A Conversation with Robert Ransom
HCM interviews the maker of PowerPad

GAMEWARE BUFFET™

58. Tablut
A 14th-century lapboard game reappears on your computer screen.
by James J. Mulligan and the HCM Staff

59. Cannibals
If the sharks don't eat you, the natives will.
by Carl Carrozza and the HCM Staff

DEPARTMENTS

4. Inside/Outside HCM
5. How to Type in HCM Programs
6. Letters to the Editor
8. On Screen
30. Group Grapevine
107. HCM Product News
146. HCM Classifieds
161. DeBUGS On DisPLAY
161. Index to Advertisers

Home Computer Tech Notes:
90. Commodore
91. IBM
98. TI
99. Apple

HOME COMPUTER™ DIGEST
News and Happenings in the Home Computer World
Bound in between Pages 66 & 67

Sea of States

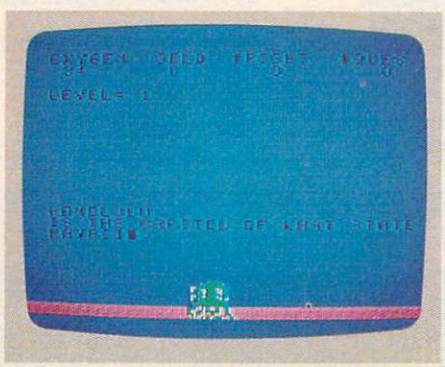
by Joseph Phillips
and the HCM Staff

Sea of States is an educational game that tests your knowledge of states and their corresponding capital cities. You play a world class scuba diver who has inherited a map from your uncle, an old sea captain. The map shows two quadrants where several Spanish galleons, laden with treasure chests full of gold, floundered and sank off the coast of Bermuda. With this knowledge, you lead an expedition to reclaim this long-forgotten treasure.

Once you arrive at quadrant 1 (level 1), you make your first dive. You can find your position in the quadrant at any time by pressing the M key, which displays an 8 x 8 grid and your relative position within that grid. This function is very useful because it lets you keep track of where you are, and allows you to make a systematic exploration of the search area. The map also helps you to avoid dangerous areas you encountered previously (where sharks abound) and to remember where you found important items, such as diving bells.

You move about in the search area by pressing the keys: N = north; S = south; E = east; and W = west. When the M option is selected, the top of the displayed map is labeled N, and other directions are labeled relative to N.

When you move from one position to another within the search area, you will encounter a number of different sea creatures, along with sunken wrecks and diving bells. Some of these creatures will be harmless and serve only to make you expel a unit of oxygen. Others, however, are quite treacherous. Sharks, for example, will force you to lose large amounts of oxygen, along with some of the gold you may have salvaged. Sunken wrecks, on the other hand, provide opposite results. When you happen upon a sunken wreck, you may receive extra units of oxygen and add to your cache of gold.



Beware the Octopus

The slyest of all the undersea creatures are the octopi. The first time you enter a region occupied by an octopus, it will ask you to match the name of a state to its capital city, or vice versa. Type in your answer and press the [RETURN] key. If the answer is correct you will be rewarded with gold and oxygen. If you are incorrect, the dark creature will become annoyed, grab you, and shake you, forcing you to lose oxygen and gold. If you happen upon the same octopus's lair more than once, you will lose more of your oxygen and gold.

Once you have answered at least five questions correctly on the first level you will need to find a diving bell, which is your passage to the second level. Once in the diving bell you need to press the U key to indicate "UP." If you have answered at least five questions correctly, you will be carried to the second level. On the second level you will need to increase your number of right answers to at least ten before you can use the diving bell again. If you have answered at least ten questions correctly throughout the game you can enter a diving bell and press U again. This time you will be carried to the surface where a helicopter will pick you up. If at any time during the game you try to go up by pressing U without being in a diving bell, you will acquire the bends and be eaten by sharks. Not a pretty fate.

The amount of oxygen remaining, number of gold pieces collected, number of questions answered correctly, and total number of questions asked are tabulated after each encounter and displayed at the top of the screen. If you run out of oxygen, you automatically lose the game. To successfully complete *Sea of States*, you must answer at least five questions on level one, find a diving bell and go up, increase your score to at least ten correct answers on level two, and again resurface in a diving bell.

Sea of States for the TI-99/4A

The TI-99/4A version of *Sea of States* uses the sprite capabilities of Extended BASIC to bring animation to your screen. Almost all of the graphics used in this program are made up of sprites. At sprite magnification 4, each sprite is composed of four characters, and each character is twice its normal size. This results in sprites 32 pixels wide and 32 pixels high.

Because *Sea of States* uses so many sprites, you have to define a large number of graphics characters. The most efficient way to do this is to place the pattern data in DATA statements and then READ the data into variables, which can be assigned to specific characters. Lines 1160

Getting down to TI™ BASICS

Fun & Games with Your TI-99/4A™

(Muncy) Eases the beginner into game development with simple programs, then gradually introduces sound, music, and other special features. Each program includes a description of its operation, a picture, plus instructions for making any game more challenging. **#6412, $8.95**

Arts & Graphics with Your TI-99/4A™

(Thompson) Picks up where **Fun & Games with Your TI-99/4A** leaves off, with more advanced static graphics and animation programs. Each program statement is described and illustrated with a picture of what the program is doing. No additional accessories or memory is needed. **#6407, $8.95**

Stimulating Simulations for the TI-99/4A™

(Engel) From the popular **Stimulating Simulations** series, this book presents 12 unique TI-99/4A games programs that provide hours of chills and challenges. Among the games are: Devil's Dungeon, Lost Treasure, Forest Fire, Monster Chase, and Diamond Thief, to name a few. **#6404, $7.50**

TI™ BASIC Computer Programs for the Home

(Sternberg) TI specific version of the best-selling **BASIC Computer Programs for the Home.** Programs help you with home financial management, auto maintenance, scheduling, recreational pastimes, and more. Programs are fully documented. For novice or experienced users. **#6402, $14.95**

I Speak BASIC to My TI-99/4A™

(Jones) New in the widely acclaimed **I Speak BASIC** Educational series, this provides an excellent introduction to BASIC programming & operation of the TI-99/4A.

The Student Text is an excellent at-home instruction manual, including dozens of programming examples and exercises. Educators can order Teacher's Manual or Classroom Sets. Order on school letterhead. **Student Text #6173, $9.75**

Introduction to TI™ BASIC

(Inman/Zamora/Albrecht) Three of the country's top programming experts show you how to use and program your TI computer. Also covers TI extended BASIC and sprite graphics for those ready to move beyond BASIC. Includes review question/answer sections. **#6401, $14.95**

Order by Phone
1-800-631-0856
operator TI44
In NJ call (201) 393-6315

HAYDEN

TI is a trademark of Texas Instruments and is not affiliated with Hayden Book Company.

Dept. TI44 • Hayden Book Company
10 Mulholland Dr. • Hasbrouck Hts., NJ 07604

Please send me the book(s) indicated below by code number. If I am not completely satisfied I may return the book(s) undamaged, within 10 days for a complete refund. I **am enclosing $2.00 to cover postage and handling.**

☐ Enclosed is my check or money order
☐ Bill my ☐ Visa ☐ MasterCard

Name

Address

City

State/Zip

Visa/MasterCard # Expires

Signature

Residents of NJ and CA must add sales tax.
Prices subject to change.

Sea of States. . . from p. 12

through 1310 contain the data for the graphics characters. The number preceding each line of character data specifies the character to which that pattern data is to be assigned. Lines 1250 through 1270 do not have this number because they are special cases in which the pattern of a character must be changed later in the program. In these situations it is easier to fix the character code within the program segment that loads the pattern, rather than write a subroutine to read both pattern and character number, as was done for the other patterns.

Several of the pattern descriptions for the sprites are approximately 64 characters long. In Extended BASIC you can define more than one graphics character with a single CALL CHAR statement. You will notice that some lines define the four patterns for a sprite separately. This is done when the majority of the sprite pattern is blank. A blank character is represented by all zeros. Any trailing zeros in a pattern description can be dropped, and if the whole character is to be blank, a single zero will suffice to assign zeros to the whole character. You can see this in characters 108, 109, 110, and 111 in line 1280. Character 108 is the only character with a pattern. The leading zeros must be included in its pattern description, but the next three characters can be defined with a single zero, because they are all blank. You could have put all four characters into the same definition, but you would have had to fill each of them with zeros, creating longer data statements that would waste a lot of memory.

The flexibility of placing the character number before the pattern defintion makes this code very efficient. Line 200 contains a subroutine that reads most of the patterns. The loop FOR D=1 TO 18 . . . NEXT D reads the data and places it into the character pattern table with the CALL CHAR statement.

The three-dimensional array SEA(, ,) is used to keep track of the contents of the sea bottom. The first two dimensions keep track of the 8 x 8 grid that makes up the sea floor map area. The third dimension specifies the level you're on. This lets the computer keep accurate records of the contents of each section of the sea floor on both levels.

SEA OF STATES (TI-99/4A)
Explanation of the Program

Line Nos.	
100-170	Program header.
180-200	Initialize program variables and graphics characters.
210-250	Display the title screen.
260-290	Initialize a new game; input player's name.
300-310	Start a new game.
320	Branch to appropriate subroutines, depending on what is on the sea bottom.
330-360	Subroutine to update the amount of oxygen used and display the score. Input the next command.
370-390	Subroutine to draw map.
400-430	Check to see if diver is at edge of the map area.
440-470	Subroutines to handle minor encounters with starfish, sea shells, and coral reefs.
480-500	Adjust oxygen and gold found or lost.
510-610	Octopus catches player. If it's the player's first time in the area, the octopus asks a question.
620-630	Encounter sharks.
640	Find a diving bell.
650	Find a sunken wreck.
660-690	Subroutine to move the objects that the diver encounters to the center of the screen.
700-780	Diver makes attempt at going up to the surface without a diving bell.
790-830	Diver goes to the second level.
840-940	Diver successfully finishes the game and surfaces from the second level; sequence to control helicopter rescue.
950-1000	End of the game; option to play again.
1010-1060	Subroutine to handle the diver when he reaches the edge of the map area.
1070	Time delay subroutine.
1080-1150	Data containing the states and their capitals.
1160-1310	Data containing graphics character patterns.

TI-99/4A

```
100 REM ******************
110 REM * SEA OF STATES *
120 REM ******************
130 REM BY JOSEPH PHILLIPS
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM TI EXTENDED BASIC
170 REM
180 CALL CLEAR :: RANDOMIZE :: DIM SEA(
8,8,2),ST$(50,2),S(7):: CALL SCREEN
(5)
190 DEF SB=SEA(S(5),S(6),S(7))
200 RESTORE 1160 :: FOR D=1 TO 18 :: RE
AD A,A$ :: CALL CHAR(A,A$):: NEXT D
 :: CALL COLOR(9,8,8):: CALL HCHAR(
1,1,96,192):: CALL MAGNIFY(4)
210 CALL SPRITE(#1,132,12,20,48,#2,140,
15,23,48,#3,140,15,23,60):: GOSUB 1
070 :: CALL SPRITE(#3,136,15,40,70,
2,2)
220 CALL SOUND(100,-6,0):: RESTORE 1080
 :: CALL MOTION(#1,0,-1,#2,0,-1)::
DISPLAY AT(10,6):"THE SEA OF STATES
"
```

TI-99/4A

```
230 FOR A=1 TO 50 :: READ ST$(A,1),ST$(
A,2):: NEXT A :: DISPLAY AT(12,1):"
YOU ARE AT THE BOTTOM OF THESEA IN
SEARCH OF GOLD!"
240 GOSUB 1070 :: CALL DELSPRITE(#1,#2)
250 DISPLAY AT(16,1):"YOU WILL ALSO FIN
D:":" :" SHARKS (HATED & FEARED)":"
UNDER WATER WRECKS":" VERY SMART OC
TOPI":" AND MUCH MORE"
260 FOR A=1 TO 8 :: FOR B=1 TO 8 :: SEA
(A,B,1)=INT(TAN(RND*1.37)):: SEA(A,
B,2)=INT(TAN(RND*1.35)):: NEXT B ::
 NEXT A
270 CALL DELSPRITE(#3):: SEA(INT(RND*8)
+1,INT(RND*8)+1,1)=4 :: SEA(INT(RND
*8)+1,INT(RND*8)+1,2)=4
280 CALL HCHAR(10,1,32,512):: DISPLAY A
T(10,1):"YOUR NAME?" :: ACCEPT AT(1
0,12)SIZE(8):N$ :: S(7)=1 :: S(1)=3
6 :: S(5)=INT(RND*8)+1
290 S(6)=INT(RND*8)+1 :: DISPLAY AT(15,
1):N$;", YOU'RE DIVING:" :: GOSUB 1
070 :: CALL CLEAR :: CALL COLOR(9,2
,10)
300 DISPLAY AT(2,1):"OXYGEN GOLD #RIG
HT #QUES. 36 0 0
0 ":" :"LEVEL= 1" :: CALL HCHAR(24,1
,96,32)
310 RESTORE 1250 :: READ A$ :: CALL CHA
R(140,A$):: CALL SPRITE(#2,136,15,1
60,100)
320 CALL DELSPRITE(#1,#3,#4):: CALL HCH
AR(15,1,32,256):: ON INT(SB+1)GOTO
440,510,620,630,640,650
330 IF SB<>4 THEN CALL DELSPRITE(#1)
340 S(1)=S(1)-1 :: IF S(1)<1 THEN 950 E
LSE IF S(1)<10 THEN DISPLAY AT(19,5
)BEEP:"! LOW OXYGEN !"
350 FOR A=1 TO 4 :: DISPLAY AT(3,A*8-7)
:S(A):: NEXT A :: DISPLAY AT(5,7):S
(7):: DISPLAY BEEP AT(20,1):"WHAT N
OW ";N$;"?":"N S E W UP MAP"
360 CALL KEY(0,A,B):: IF A>87 OR A<69 T
HEN 360 ELSE ON A-68 GOTO 420,360,3
60,360,360,360,360,360,370,400,360,
360,360,360,410,360,700,360,430
370 REM DRAW MAP
380 CALL HCHAR(13,1,32,288):: FOR A=1 T
O 8 :: CALL HCHAR(7,A*2+8,A+48):: C
ALL HCHAR(24-A*2,9,A+64):: NEXT A
390 CALL HCHAR(6,17,78):: CALL HCHAR(24
-S(5)*2,S(6)*2+8,88):: GOSUB 1070 :
: CALL HCHAR(6,1,32,544):: GOTO 350
400 IF S(5)<8 THEN S(5)=S(5)+1 :: GOTO
320 ELSE 1010
410 IF S(5)>1 THEN S(5)=S(5)-1 :: GOTO
320 ELSE 1010
420 IF S(6)<8 THEN S(6)=S(6)+1 :: GOTO
320 ELSE 1010
430 IF S(6)>1 THEN S(6)=S(6)-1 :: GOTO
320 ELSE 1010
440 IF SB=0 THEN ON INT(RND*3+1)GOTO 45
0,460,470 ELSE ON SB*10 GOTO 450,46
0,470
450 A$="THERE IS NOTHING HERE BUT SAN
D AND SEA SHELLS." :: C=16 :: SH=11
2 :: GOSUB 660 :: GOTO 330
460 A$="SMALL STARFISH ATTACK YOU." ::
C=16 :: SH=104 :: GOSUB 660 :: GOTO
 330
470 A$="THE CORAL IS ATTRACTIVE BUT THE
RE IS NOTHING OF VALUE." :: C=10 ::
 SH=108 :: GOSUB 660 :: GOTO 330
480 IF SB=5 THEN D=2 ELSE D=1/SB
490 A=INT(RND*10*D)+1 :: B=INT(RND*10*D
)+1 :: GOSUB 1070 :: DISPLAY AT(15,
1):A;"TURNS OF OXYGEN AND":B;"GOLD
PIECES":
```

Continued on p. 102

Animation, in the computer context, is the illusion created when an object springs to life and exhibits qualities of movement usually found only in the real world. *Sea of States* uses this technique to send a helicopter flying across the video screen. Let's look at how the Commodore 64 version of the program implements this effect.

The helicopter is a *sprite*. Sprites are used for animation because they can be moved one pixel at a time, resulting in very smooth motion across the screen. The trick to creating animation is to define a series of sprites (as many as you need to create the illusion), and then interchange them at regular intervals. The sprites you define should all be oriented in the same position, but each one should be slightly different to create a series that simulates movement.

For example, a helicopter's most obvious movement is that of the main and rear rotor blades. By defining just two sprites with rotors in different positions and then interchanging them, we were able to create a very effective animated sequence.

Interchanging sprites can be handled in more than one way, but perhaps the best way is not to change the sprite at all. This may sound confusing, but remember that sprite definitions are much more easily interchanged than the values of the sprite registers. It is possible to create several different sprite patterns and display them all using only one set of sprite registers. We do this by enabling one sprite, then changing the sprite definition pointer to point to another sprite definition in memory. The following BASIC lines illustrate this concept:

```
10 POKE 53248,25:POKE 53249,75
20 POKE 2040,240
30 POKE 53287,1:POKE 53269,1
40 FOR I=25 TO 200:FOR J=1 TO
   10:NEXT
50 IF PEEK(2040)=240 THEN POKE
   2040,241: GOTO 70
60 POKE 2040,240
70 POKE 53248,I:NEXT
80 END
```

Lines 10 through 30 are preparatory. Line 10 sets the starting X and Y pixel positions at 25 and 75 respectively. Line 20 assigns the sprite definition starting at address 15336 (64*240) as the initial sprite to be displayed. Address 2040 is the pointer for sprite #0. Line 30 sets the sprite color to white (1) and turns on sprite #0.

Lines 40 through 70 contain the programming loop that moves sprite #0 across the screen and interchanges the sprite definition pointer value to create the animation. In line 40, the FOR-NEXT loop that uses I for its control variable keeps track of the pixel positions for movement. The loop with the control variable J controls the speed of movement across the screen. You can alter the speed by making this loop smaller or larger.

Lines 50 and 60 work together to interchange the value of the sprite definition pointer. This part may be expanded with the use of a count variable to allow an interchange of more than two sprite definitions. Line 70 updates sprite #0's X register value in order to move the sprite across the screen from left to right.

Good animation really does liven up game programs. With this short BASIC program and a little bit of practice, you can install exciting animation sequences into your own programs.

SEA OF STATES (C-64)
Explanation of the Program

Line Nos.	
100-160	Header.
170-240	Initialization for programmable graphics.
250-260	Read in character and sprite data.
270-280	Initialize variables and arrays.
290-950	Character and sprite definitions.
960-1090	State and capital data.
1100-1130	Initialization.
1140-1240	Randomly select game characters of levels 1 and 2.
1250-1390	Main program loop.
1400-1560	Display program header.
1570-1630	Get keyboard input subroutine.
1640-1680	Display score routine.
1690-1700	Clear middle of screen routine.
1710-1760	Display map subroutine.
1770-1780	Clear keyboard buffer routine.
1790-1940	Found octopus routine.
1950-2250	Message subroutines.
2260-2740	Completed levels subroutine.
2750-2820	Routine to display loss and gain of gold and oxygen.
2830-2900	End of game routine.

COMMODORE 64

```
100 REM  **********************
110 REM  *  SEA OF STATES  *
120 REM  **********************
130 REM  BY JOSEPH PHILLIPS AND THE HCM
     STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  C64 BASIC
170 POKE 52,132:POKE 56,132:CLR
180 POKE 56334,PEEK(56334)AND254
190 POKE 1,PEEK(1)AND251
200 FOR I=0TO1023:POKE I+47104,PEEK(I+5
    3248):NEXT
210 POKE 1,PEEK(1)OR4
220 POKE 56334,PEEK(56334)OR1:POKE 5657
    8,PEEK(56578)OR3
230 POKE 56576,(PEEK(56576)AND252)OR1:P
    OKE 648,132
240 POKE 53272,(PEEK(53272)AND240)OR14:
    POKE 53281,6:POKE 53280,6
250 FOR I=47616 TO 47615+(2*8):READ A:P
    OKE I,A:NEXT
260 FOR I=48128 TO 48127+(16*64):READ A
    :POKE I,A:NEXT
270 DIM ST$(50,2),SEA(8,8,2),S(7)
```

COMMODORE 64

```
280 B1=34811:B2=34809
290 FOR I=1TO50:READ ST$(I,1),ST$(I,2):
    NEXT
300 DATA 56,56,16,84,146,16,40,198,255,
    255,255,255,255,255,255,255
310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
320 DATA 0,0,0,0,0,0,31,248,0,31,254,0,32
    ,15,230,64,7
330 DATA 254,31,255,239,31,252,66,32,0,
    64,64,0,32,0,0,16
340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,15,192
    ,16,15
360 DATA 224,0,16,112,16,32,60,8,0,14,0
    ,0,7,230,0,7
370 DATA 254,0,15,239,15,252,66,15,248,
    192,16,0,128,32,1,128
380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
390 DATA 0,0,0,0,0,0,0,0,0,16,2,0,48,2,0,
    16,7
400 DATA 252,56,10,0,16,18,24,16,34,24,
    16,255,255,252,255,255
410 DATA 248,255,255,240,127,255,224,63
    ,255,192,31,255,128,0,0,0
420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
430 DATA 0,0,0,0,8,0,0,28,0,0,62,0,0,12
    7,0,0
440 DATA 255,128,1,156,192,1,156,192,1,
    255,192,0,227,128,224,127
450 DATA 0,191,255,254,191,207,57,164,3
    6,141,36,36,69,34,36,37
460 DATA 66,20,70,68,36,36,136,24,8,0,0
    ,0,0,0,0,0,0
470 DATA 0,0,0,0,0,0,62,0,0,34,0,0,127,0,1,
    255,192,3
480 DATA 255,224,7,243,240,15,225,248,1
    5,192,248,31,192,252,31,225
490 DATA 252,31,243,252,15,255,248,15,2
    55,248,7,255,240,3,255,224
500 DATA 1,255,192,0,127,0,0,0,0,0,0,0,
    0,0,0,0
510 DATA 0,0,0,0,0,0,0,0,0,0,0
520 DATA 0,0,0,0,0,0,8,0,0,4,0,0,14,1,1
    92,4
530 DATA 8,160,4,24,144,4,120,136,228,2
    48,252,127,255,248,63,255
540 DATA 248,31,239,240,15,207,224,7,22
    7,192,0,0,0,0,0,0,0
550 DATA 0,0,0,0,0,0,0,0
560 DATA 0,0,0,0,0,56,0,0,124,0,1,254,0,3
    ,255,0,7
570 DATA 255,0,15,255,128,15,255,128,24
    ,63,128,0,15,192,0,3
580 DATA 192,0,1,192,0,1,192,70,0,192,1
    27,128,64,127,224,192
590 DATA 63,241,192,31,255,192,15,255,1
    92,3,255,128,0,0,0,0
```

Continued on p. 102

JOYSTICKS BUILT TO LAST!

MADE SPECIFICALLY FOR TI 99/4A
NO ADAPTORS NECESSARY

SPECIAL MAIL ORDER OFFERING

Limited Offer Only

$19.95 EACH PAIR

MODEL J1020
Includes two joysticks with common cord for TI 99/4A

WORKS BETTER & LASTS LONGER

If you've tried to use other joysticks on your 99/4A, odds are they have not worked properly and/or broken down in a short period of time. Super Stik™ is the best joystick on the market made specifically for the TI 99/4A Home Computer with no adaptors necessary. What's more, Super Stik is made to last longer and work better than any other . . . backed by a one year limited warranty. No other joystick can match this claim.

WHY SUPER STIK IS BETTER

As mentioned, Super Stik is made specifically for the TI 99/4A home computer. This means that there is no need for special adaptors like other joysticks. A common 5' cord is included with each joystick pair.

The Super Stik Model J1020 has "built-in" isolation electronics that prevents "Crosstalk" between the two units. Without this isolation, joysticks will not function properly on the TI 99/4A.

J1020 units feature a simple, yet durable design with fewer moving parts. The housing is designed to completely stop handle movement before damage can occur to the contacts, providing a unique "positive stop" action. The assembly withstands a breaking strength of more than 50 lbs. and has been tested to more than 500 continuous hours without failure.

ARCADE FEEL & PERFORMANCE

Users of the Super Stik love its feel and performance. It features a long, comfort contoured handle, 8-way 45° angle control, and handle mounted firing button.

SUPER STIK MODEL J1000

Also Available For Only **$9.95** EACH

The Super Stik model J1000 is compatible with Atari® Video Game, Atari® 400/800 Computers, Sears Video Arcade and Commodore VIC-20/C-64 Computers. The same rugged construction and super performance is offered. Don't miss the Super Stik at this price.

Atari, Commodore, Sears and Texas Instruments are registered trademarks. Super Stik is patent pending. Copyright, 1983. Made in U.S.A.

DON'T MISS THIS SPECIAL MAIL ORDER OPPORTUNITY

This may be your only opportunity to purchase the right joystick for your TI 99/4A at this low price. All orders will be allowed a 30-day money back guarantee if not **completely** delighted by its performance.

SEND YOUR PAYMENT WITH THE FOLLOWING COUPON TO:
RC DISTRIBUTING COMPANY
4016 Sanguinet
Fort Worth, Texas 76107

To Figure Your Payment For Each Model:

Super Stik J1020 (pair)	**$19.95** pair
Super Stik J1000	**$9.95** each
Postage & Handling for Each Item Ordered	**$2.00**
For Texas Deliveries Add 5% Sales Tax,	**$1.00** for each J1020
	$0.50 for each J1000

Please rush my joystick(s) to the address below. My payment of $ _____ is enclosed for the appropriate Super Stik Model(s). If not completely satisfied with your purchase, return your joystick(s) within 30 days of receipt (in the box it came in) for a full and prompt refund. Offer good only while supply lasts.

☐ J1020 Super Stik Quantity _____

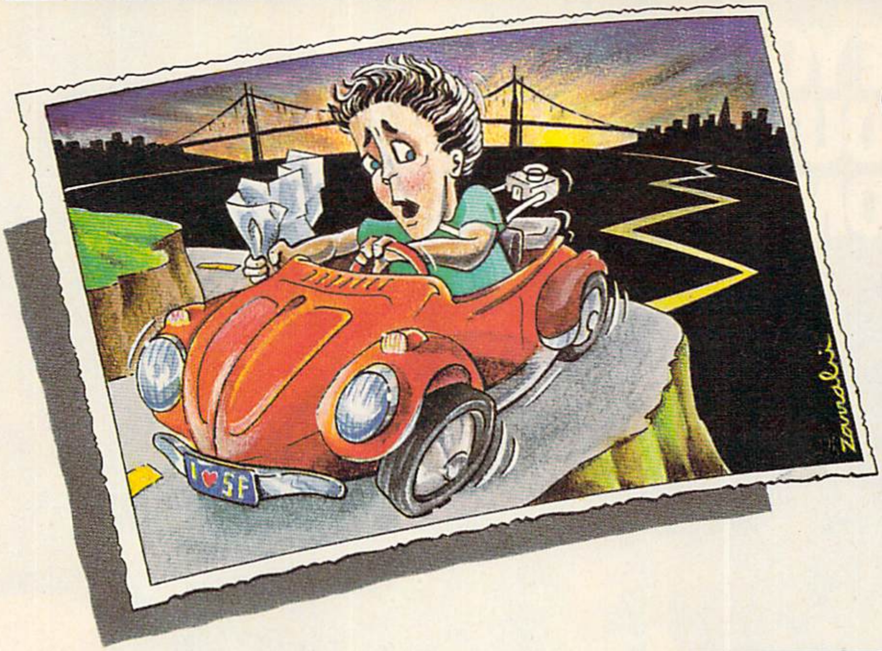
☐ J1000 Super Stik Quantity _____

NAME _____

STREET ADDRESS _____

CITY _____

STATE _____ ZIP _____



San Fransisco Tourist

You cruise into town to the strains of "I Left My Heart In San Francisco," but then the fun ends. You're from Iowa, and suddenly you're faced with driving down the "crookedest street in the world,"—Lombard street between Hyde and Leavenworth—and it's a steep hill to boot.

Use the S key to steer left and the D key to steer right. You'll probably want to take it slow at first, unless you're confident that anything these city slickers can do, you probably can do better.

If you do happen to run into the curb, you'll dent your car a little. If you're so unfortunate as to dent it 8 or so times on the way down, then you'll find yourself stranded.

San Francisco Tourist is a short, easy-to-key-in program for the VIC-20. It originally appeared in *99'er Home Computer Magazine*. [The TI version can presently be found in *The Best of 99'er*.] At the beginning it allows you to select the rate at which you travel down the street; and if you successfully negotiate the curves, it reports how many dents you received. You can then select a different speed and try the journey again.

For faster execution speed and to save memory, the shape of the car is represented by a block character. It is POKEd into screen memory. The street is printed line-by-line on the bottom of the screen causing the whole street to scroll up and simulate the car's downward movement. The keyboard is read with the GET statement, and Boolean algebra is used to change the position of the car.

—Robert Keller

***SAN FRANCISCO TOURIST* (VIC-20)**
Explanation of the Program

Line Nos.	
100-160	Program header.
170-320	Initialization.
330-420	Program loop.
430-470	End of play.
480-530	Check for crash.
540-550	Title song data.

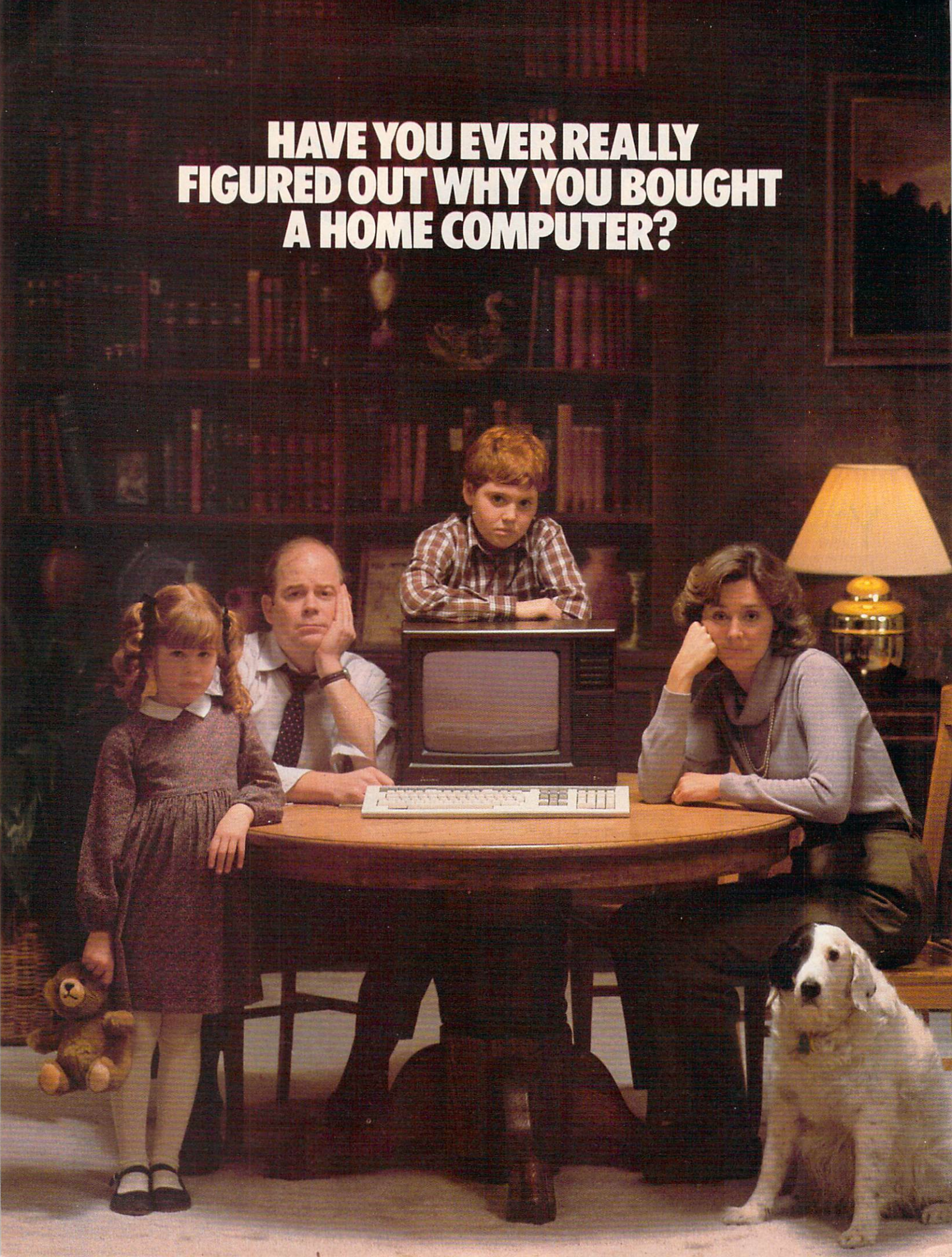
VIC-20

```
100 REM ************
110 REM *SF TOURIST*
120 REM ************
130 REM BY THE HCM STAFF
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM V20 BASIC
170 PRINT"{SHIFT CLR}{9 CRSR DOWN}SAN FRANCISCO TOURIST":A$="{CTRL RVS ON}{CTRL GRN}"+CHR$(32)+"{CTRL WHT}":B$="{CTRL RVS ON}{CTRL BLK}"+CHR$(32)+"{CTRL WHT}"
```

VIC-20

```
180 T=11:S3=36876:V=36878:POKEV,T
190 FORZ=228TO231:POKES3,Z:FORK=0TO5:NEXT:POKEV,T:T=T+1:NEXT
200 READD,N:IFD=-1THENPOKEV,0:GOTO220
210 POKES3,N:FORZ=0TOD*10:NEXT:GOTO200
220 PRINT"{SHIFT CLR}{3 CRSR DOWN}LOMBARD  STREET IS THE"
230 PRINT" CROOKEDEST STREET IN"
240 PRINT"     THE WORLD. YOUR"
250 PRINT"CHALLENGE IS TO DRIVE    DOWN IT SAFELY."
260 PRINT"{2 CRSR DOWN} USE 'S' TO MOVE LEFT    'D' TO MOVE RIGHT"
270 PRINT"{2 CRSR DOWN}PRESS ANY KEY TO PLAY"
280 GETD$:IFD$=""THEN280
290 PRINT"{SHIFT CLR}{8 CRSR DOWN}WHAT SPEED DO YOU WANT TO GO..FAST OR SLOW?"
300 GETF$:IFF$<>"F"ANDF$<>"S"THEN300
310 S=-200*(F$="S")
320 N=11:P1=7910:P=7910:PRINT"{SHIFT CLR}"
330 PRINT"{CRSR RIGHT}{9 CRSR DOWN}{CTRL RVS ON}{CTRL BLK}LOMBARD{CTRL RVS OFF}{CTRL RVS ON} STREET":FORL=1TO92:FORZ=0TOS:NEXTZ
340 IFL<4THEN370
350 X=INT(RND(0)*3)-1:IFN-4+X<3ORN+4+X>19THEN350
360 N=N+X:IFL>80THENPRINT:FORZ=1TO25:NEXT:GOTO400
370 PRINT"{CRSR RIGHT}";:FORR=1TON-5:PRINTA$;:NEXT:PRINTB$+"       "+B$;
380 FORR=N+4TO19:PRINTA$;:NEXT:PRINTA$
390 IFL<12THENNEXT
400 GETK$:P1=P1+(K$="S")-(K$="D")
410 IF(PEEK(P1+30720)AND15)<>1THENGOTO480
420 F=0:POKEP1+30720,0:POKEP1,215:POKEP+30698,1:P=P1:NEXTL
430 PRINT"{CTRL BLU}{CTRL RVS OFF}{4 SHIFT CRSR UP}{5 CRSR RIGHT}YOU MADE IT!":PRINT"NUMBER OF CRASHES:";C
440 FORZ=1TO1000:NEXT:PRINT"PRESS ANY KEY TO PLAY"
450 GETD$:IFD$<>""THEN450
460 GETD$:IFD$=""THEN460
470 F=0:C=0:PRINT"{SHIFT CLR}";:GOTO290
480 T=PEEK(36879):FORB=0TO7:POKE36879,PEEK(36879)AND248ORB:NEXT:POKE36879,T:IFFTHEN400
490 FORZ=14TO0STEP-2:POKE36874,221+Z:POKE36875,228-Z:POKE36876,216+Z:POKE36877,201+Z
500 POKE36878,Z:NEXT
510 C=C+1:F=-1:IFC>8THENPRINT"{CTRL RED}TOO MANY CRASHES!":FORN=1TO1000:NEXT:GOTO440
520 GETK$:Z1=P+(K$="S")-(K$="D"):IF(PEEK(Z1+30720)AND15)<>1THENGOTO520
530 P1=Z1:GOTO420
540 DATA 20,231,20,232,20,237,3,235,3,237,96
550 DATA 235,20,237,20,239,20,240,20,237,80,228,-1,-1
```

HCM


HAVE YOU EVER REALLY
FIGURED OUT WHY YOU BOUGHT
A HOME COMPUTER?

WE HAVE.

Most people have never really figured out why they bought a home computer. That's because most people have never figured out what their home computer can really do. There's more to it than balancing a checkbook or zapping aliens.

But how do you learn enough about your computer to figure out what it can really do? You could buy one of the thousands of computer books on the market. Or you could buy one of the thousands of pieces of educational software. But the problem is that the books and software don't go together. So when you use one, you can't use the other. And you really should have both.

SOFTEXT has the solution. We have a selection of intelligent, stimulating, integrated, software/book packages. The software and books can be used alone or in combination. And each software/book package also includes an easy-to-understand User's Guide.

So don't sit around trying to figure out why you bought a home computer. Instead, figure out how you can get your hands on a SOFTEXT package.

AT THE HEART OF THE MOUNTAIN
This package teaches children and adults to program in BASIC by involving them in a fascinating adventure story. Join the characters in a journey of discovery as they too learn about computers for the first time.

EASY PROGRAMMING WITH THE TI-99/4A
Get the most out of your TI-99/4A. Learn efficient, professional programming techniques even if you're a beginner.

DISCOVER YOUR VIC-20
Using specific examples, this is an adventure-based package that goes beyond the traditional BASIC introduction.

THE COMMODORE 64 MUSIC MASTER
This is the only true "teaching" package of its kind. Learn to compose and create music while exploring sophisticated programming skills.

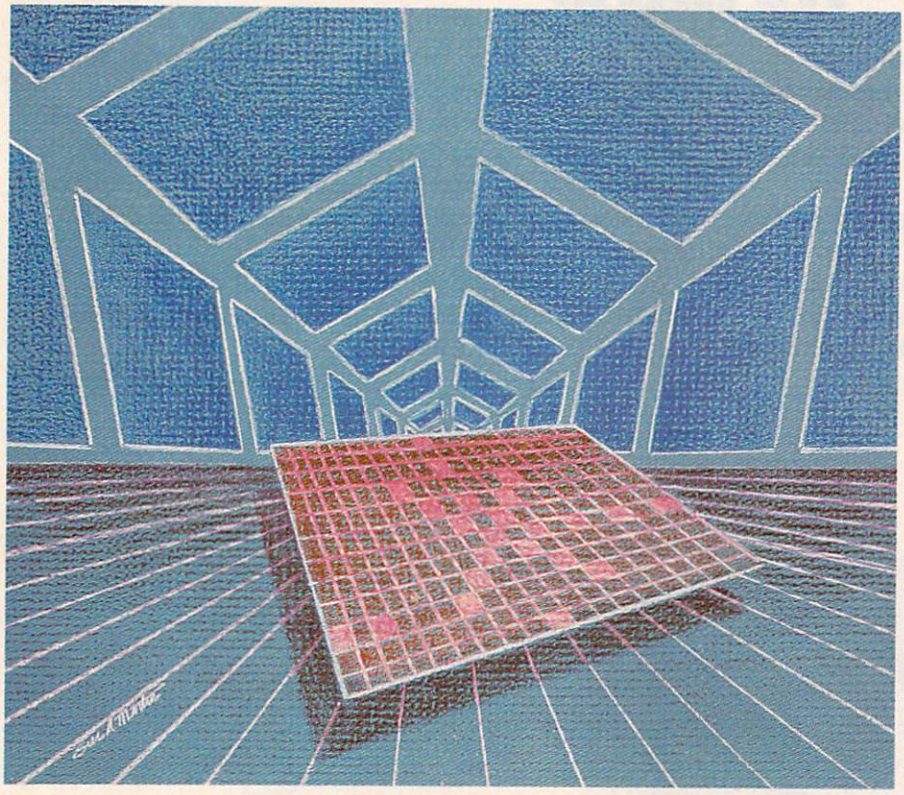
THE COMMODORE PUZZLE MASTER
This package contains fifty brainteasers and their programmed solutions. Learn how to design your own puzzles with color, sound and graphics.

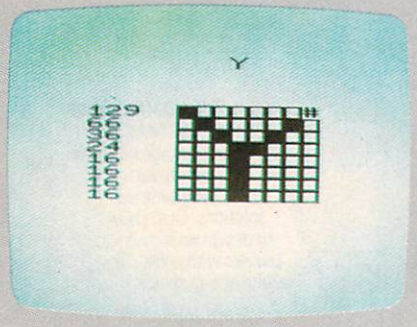
AN INTRODUCTION TO THE COMMODORE 64
This is a thoughtful, entertaining, self-paced guide to programming. An important investment for any Commodore 64 owner. (Shown with book and tape)

SOFTEXT

WE'RE THE REASON YOU BOUGHT A HOME COMPUTER.

SOFTEXT INC., 380 Green Street, P.O. Box 2007, Cambridge, MA 02139, (617) 876-2334





BUILDING YOUR CHARACTER

A Graphics Editor for the VIC-20

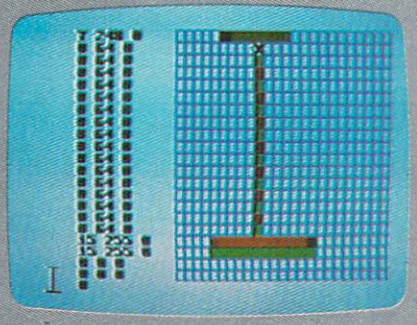
by John Thrasher
HCM Staff

Good news, VIC-20 owners! Here's a unique and powerful program that fits snugly into your machine's tiny memory and lets you flex those big programmable character muscles like never before. With this program, you can develop and experiment with programmable characters. By way of a character assembly grid displayed on the video screen, you can move a flashing cursor to turn the pixels of a character on and off.

Once you create a character you like, you have only to press the N key, and the decimal code for the character will appear on the left side of the screen—quite a relief from having to convert an 8 by 8 matrix into decimal code.

We've knocked out one tedious chore. Let's go to work on another. Press the J key, and the program will create a DATA statement containing the decimal code. You can create DATA statements for a maximum of eleven characters. (Sorry we can't give you more, but the VIC-20 memory is just too small.) Early in the program you'll be given an opportunity to select a line number for the first DATA statement to be assembled. At the same time, you'll be prompted for an increment value to create line numbers for succeeding DATA statements. (I usually use an increment value of 10 so that I can have nice even line numbers.) Each DATA statement will contain the code for one character. It takes eight codes for one character definition; thus each statement will contain eight numbers.

Before we present a method for using the DATA statements, let's summarize the special commands used in this program:



QUICK PIXEL TRIX

A Graphics Editor for the C-64

by John Thrasher
HCM Staff

Often in the course of programming on the Commodore 64 we find it necessary to program graphics characters and sprites tailored to suit the needs of a particular program. This in turn usually means that we spend a great deal of time experimenting with pencil and paper to come up with graphics that suit our purposes. Once we have shapes that appear adequate on paper, we then must convert these shapes into decimal code to be entered into our program. After the image definitions are loaded, we must then have our program display them on

KEY	COMMAND
N	Displays decimal code of the character on the screen.
U	Clears both the assembly grid and the defined character. This command can be used when you want to start assembling a new character. DATA statements remain intact.
J	Creates DATA statements containing the decimal code of the character on the assembly grid.
Q	Exits program and allows you to save DATA statements as a program file.

"Many VIC-20 users don't realize that it's possible to generate displayable characters other than those available from the keyboard."

Taking Command

Now that you know what all the commands do, let's look at how to use them. First you load the *Character Editor* program. Type RUN and hit the [RETURN] key. After a few moments, you'll get a message on the screen asking if you want to create a DATA statement program file. If you do, press Y, then [RETURN], and the program will ask for a beginning line number. This can be any number between 1 and 65000. The actual number depends on where you want the DATA statement to be in the program for which you are developing these character definitions. It is a common practice to place DATA statements at the end of programs; a line number of 10000 would probably be a good place to start. After this you'll be prompted for an increment value. If you input 10000 at the DATA statement prompt and 10 at the increment prompt, then the first DATA statement you create will have a line number of 10000, the second a value of 10010, the third a value of 10020, and so on.

Once you have entered the increment value, you are ready to start developing your own programmable characters. When the assembly grid is displayed on the screen, the *Character Editor* program will execute all of the commands we've described above. The following controls allow you to move and update the cursor position on the assembly grid:

FUNCTION	KEYBOARD	JOYSTICK
Cursor left	S Key	Move left
Cursor right	D Key	Move right
Cursor up	E Key	Move up
Cursor down	X Key	Move down
Character on or off	Spacebar	Fire-button

The spacebar or the fire-button turns the character behind the cursor on or off depending on the previous state of the character. If the character in that position was on, it will then be turned off, and vice versa.

When you have finished assembling a character, you have the option of pressing either the J key to create a DATA statement containing the decimal code or the N key to display the decimal code for copying with pencil and paper. After this you can either press the U key to clear the assembly grid and start a new character, update the character currently displayed to form a new character, or Quit.

If you are finished and have elected to create DATA statements, then you must exit the program by pressing the Q key. This will allow you to save the DATA statements you have created. If you have created DATA statements, then you obviously want to use these new character definitions in your own program. So you need to know how to incorporate these character definitions into your own program.

When I develop a program, I usually first determine if I will need special characters not found in the regular character set of the VIC-20. If I do, then I go ahead and create these new characters and their DATA statements using the Character Edi-

Continued on p. 22

the video screen to find out if our paper and pencil graphics match our video art. More often than not, we find that at least some slight modification is in order, so we make corrections and test again and again until we achieve perfection.

If you have engaged in a process like the one we've described, then you'll surely appreciate this *Graphic Editor* program. It allows you to throw away all of your blank image graphs and create graphics directly on the video screen. As an added bonus, this program provides a facility by which you may create DATA statements containing the decimal codes for the graphics you create on the screen. These DATA statements may then be saved as a stand-alone program file or merged directly into your own programs. It is also possible to print out a hard copy of the graphic image along with its decimal code.

The Equipment You'll Need

You'll need the basic essentials: a C-64 console, a video screen, some sort of storage device (either cassette tape or disk drive), and a VIC 1525 graphics printer. The printer is required only if you want a hard copy of your graphics. If you have a printer but it's not a VIC 1525, you may have to modify the print subroutines contained in lines 990-1180 and 2250-2320. It would also be nice if you had a joystick to control the cursor on the video image grid, but this is not absolutely necessary, as the cursor may be controlled by the S, D, E, and X keys. If you use a joystick, put it in control port #2.

Cursor Control

The following table describes cursor control on the image matrix:

FUNCTION	KEYBOARD	JOYSTICK
Cursor left	S Key	Move left
Cursor right	D Key	Move right
Cursor up	E Key	Move up
Cursor down	X Key	Move down
Character on or off	Spacebar	Fire-button

The spacebar and fire-button turn the character behind the cursor on or off depending on the previous state of the character. If the character in that position was on, it will then be turned off, and vice versa.

Graphic Editor Commands

The following table summarizes the commands of the *Graphic Editor* program:

KEY	COMMAND
N	Displays decimal code of image on screen.
U	Clears image grid and image definition.
C	Changes color of sprite and succeeding "on pixels" of the image assembly matrix (used only for sprites).
P	Prints image decimal code and a graph of image on the printer.
J	Creates DATA statements containing the decimal code of the image currently displayed on the screen.
Q	Exits program and allows you to save DATA statements (either alone or merged) to a program file.
2	Allows switching back and forth between sprite editing and character editing.

Some of these commands may require further clarification, so a detailed analysis of each command follows:

Continued on p. 23

Building . . .

tor. When I've finished developing all the special characters I need, I press the Q key to enable me to SAVE the DATA statements as a program file. I then SAVE the DATA statements as I would SAVE any BASIC program on the VIC-20, i.e., SAVE *"filename",devicenumber*. The filename I select is the name of the program that will use the special characters. At this point it contains only the DATA statements.

Once I have SAVEd the DATA statements as a program file, I must reset the VIC-20 before I can go any further. This is because the Character Editor program changed some very critical Operating System pointers when I pressed the Q key; they must be reset to allow normal operation. To reset the VIC, just turn it off and back on again.

Next I LOAD the program file I just SAVEd. The program in memory now contains only the DATA statements, and I just build on to it as I would any BASIC program I'm typing in.

Using Programmable Characters in Your Program

Many VIC-20 users don't realize that it's possible to generate displayable characters other than those available from the keyboard. The users manual that accompanies most VIC-20's doesn't mention it, but it's possible, and actually quite simple, once you understand the method.

The VIC-20 normally gets all the character description information from ROM (Read Only Memory). This ROM can't be changed, so the only way we can program our own characters is to tell the VIC to get its character information from somewhere else—from an area of RAM (Random Access Memory). Then we have to figure out where in RAM we want to store the character definitions. On the VIC-20 with no memory expansion, there is only one reasonable place to put them, and that is in a 512-byte area of RAM beginning at address 7168. We can tell the VIC-20 where to find its character definitions by POKEing 36869 with 255.

When we use this 512-byte area of RAM, we actually do two more things that we must be aware of. We reduce the BASIC programming area by 512 bytes, meaning we have less room in which to build our program; and since we now have only 512 bytes of storage for character definitions, we have a maximum of only 64 characters available for display. This is because each character definition requires eight bytes, and 512/8 equals 64.

Reducing our BASIC programming area by 512 bytes may seem unreasonable to VIC-20 owners, who climb mountains to save every possible byte, but if you want programmable characters, it is absolutely necessary. Because we are taking this area away from the BASIC Operating System, we must notify it of our plans so that it will not try to write over our programmable characters. We do this with the following command:

```
POKE 52,28:POKE 56,28
```

Finally, we must POKE valid character definitions into this 512-byte area of RAM. Remember that we can use only 64 characters. (For most programs this should be plenty.) I usually copy the first 64 characters from the ROM into my area of RAM. These are the characters displayed when you POKE a value between 0 and 63 to screen memory (this includes all the alphabetic and numeric characters), and they are adequate for most programs. To copy these definitions, use the following BASIC lines:

```
FOR I=7168 TO 7679 POKE I,PEEK
(I+25600):NEXT
```

The last thing we need to do is put the character definitions contained in the DATA statements created by the *Character Editor* program into this 512-byte area of RAM. When we POKE these definitions into RAM, we will be writing over the definition of one of the first 64 POKE characters, so we must be sure to write over only those characters that will not be used in our program. This usually means that we will not want to change any of the alphabetic or numeric characters. Consequently, we have a selection of some 28 characters left that we can change. As an example of how this might be done, consider the next few lines of BASIC code:

```
10 POKE 52,28:POKE 56,28
20 FOR I=7168 TO 7679
30 POKE I,PEEK(I+25600):NEXT
40 POKE 36869,255
50 FOR I=7384 TO 7383+(2*8)
60 READ A:POKE I,A:NEXT
10000 DATA 255,129,129,129,129,129,
      129,255
10010 DATA 255,255,255,255,255,255,
      255,255
```

Lines 10 through 40 set the new boundary of the BASIC programming area, copy the first 64 characters from ROM into RAM and tell the system where to find its character definitions. Lines 50 and 60 READ and POKE into memory the DATA in lines 10000 and 10010; these DATA statements are two-character definitions created by the *Character Editor* program. These definitions take the place of the POKE characters 27 and 28. The decision to write over these particular characters was based on the fact that I would not need them in my program. To get the proper address at which to begin POKEing these new definitions, I multiplied 27 by 8, which equals 216. Then I added this to the beginning address of 7168 to get a LOAD address of 7384.

So there you have it. With a little bit of practice and experience you should be able to use the *Character Editor* program to create any character you desire quickly and easily. I am sure this is one program you should take the time to type in and SAVE. Good luck.

Continued on p. 48

Quick Pixel . . . from p. 21

N KEY When pressed, the decimal code for the image currently displayed on the video matrix will be displayed on the left side of the screen. The image assembly will remain intact.

U KEY Clears the image definition from memory and also clears the video assembly matrix. DATA statements that have been created will remain intact.

C KEY Changes the color of the sprite displayed in the lower left-hand corner of the screen. The color of succeeding characters, used to represent sprite bits turned on, also changes to the new color. The number of the new color will be requested by a prompt at the top of the screen when the C key is pressed. This number is the same as the numbers normally associated with colors on the Commodore 64, That is, black equals 0, white equals 1, and so on.

P KEY If you wish to use this command, you must have a VIC 1525 graphics printer hooked up to your system. If you have a printer other than this, you will probably have to modify the subroutine that handles this command. These subroutines are from lines 990-1180 and 2250-2320. [If any readers do modify these subroutines for other printers, please share the details in our "Letters to the Editor" section.—Ed.] When this key is pressed, the decimal codes for the image currently displayed are sent out to the printer. Along with this, a graphic representation of the image is also printed. This hard copy will allow you to keep a permanent visual record of all graphics that you have created.

J KEY The use of this key is probably the most difficult to understand. When you press the J key, a subroutine is called that will create DATA statements containing the decimal code of the image currently displayed on the video assembly matrix. These DATA statements will be assembled into an area of the BASIC RAM that is not used by the *Graphic Editor* program but is still available to the BASIC operating system. This will allow you to either save the DATA statements as a stand-alone program file or append them directly to your program file in memory. This command works in conjunction with user prompts that appear when the program is first RUN.

When you first RUN the program, it prompts to find out if you are using joysticks. After this, the program asks if you want to create DATA statements from the graphics you will create. If you do, the program prompts you for the line number you want to assign to the first DATA statement. If you are merging DATA statements to a program, this line number should be greater than the last line of that program because the DATA statements will be appended to the end.

There will be four DATA lines (containing 16 numbers each) for each *sprite* on

Continued on p. 24

Quick Pixel . . . from p. 23

which you select the J command, and one DATA line, containing eight numbers, for each *character* on which you select the J command. It is possible to press the J key several times for the same image and have the same sequence of code repeated. You would not, of course, want to do this. This example merely describes an action of the J command.

After you choose the starting line number, you are prompted for an increment value. This number will be added to succeeding line numbers of DATA statements. I usually enter 10 on this prompt.

Once this prompt is entered, you are then asked if you are merging DATA statements with an existing program. If you are, then the program that the DATA statements will be appended to must already be in memory. You must load this program in memory before the *Graphic Editor* program is loaded. Here's an outline of the procedure:

1) Reset the computer by turning it off and then on again.

2) In the command mode, POKE 44,64. Address 44 is the pointer to the beginning of BASIC text. What we are doing here is allowing your program to be loaded in an area of BASIC RAM that is not used when running the *Graphic Editor* program.

3) LOAD the program you wish to append DATA statements to. Your program will be loaded starting at address 16385.

4) POKE 44,8. This sets the beginning of BASIC text pointer back to its normal location.

5) LOAD the *Graphic Editor* program.

6) RUN the program.

"The *Graphic Editor* program allows you to throw away all of your blank image graphs and create graphics directly on the video screen."

If you do not want to merge a program but instead wish to create a program file consisting only of DATA statements, then enter N to this prompt.

Q KEY This command is necessary when exiting the *Graphic Editor* program if you want to save your program containing the DATA statements. When this key is pressed, the program exits via a short routine in lines 1800 through 1830. Here the BASIC operating system is assigned a new program to work with. It is the program you created, and you may now LIST or SAVE it. After you have LISTed or SAVEd this program, you cannot rerun the *Graphic Editor* program or any other program until the system is reset and a new program is loaded. To reset the C-64, turn it off and then back on again.

2 KEY When you press the number 2 key, the program will return to the menu and let you select either the Character Editor mode or the Sprite Editor mode.

A Quick Summary

Now let's summarize the procedure to follow when using the *Graphic Editor* program. First, reset your system. Then if you want to merge DATA statements into a program, POKE 44,64 in the command mode, and LOAD your program. When the program is loaded, POKE 44,8 and LOAD the *Graphic Editor* program. RUN. From this point on, the program is user-friendly and prompts for all necessary input data. Once the Graphic Video Assembly matrix is displayed, the program will execute all of the key-initiated commands. Good luck, and I hope this program helps you as much as it has helped me.

GRAPHIC EDITOR (C-64)
Explanation of the Program

Line Nos.	
100-160	Header.
170-230	Transfer ROM characters to RAM.
240-260	Read and store special character patterns.
270	Joystick prompt.
290	Initialization of variables.
300-310	Initialize sprite #0.
320-430	Main program loop.
360-430	Check command keys.
440-570	Obtain cursor direction from either the keyboard or the joystick.
580-870	Subroutine to update cursor position and to change the sprite pattern of both the assembly matrix and the sprite displayed in the lower left-hand corner.
880-980	Subroutine to display decimal code of sprite definition on the left side of the video screen.
990-1180	Subroutine to output to printer.
1190-1220	Portion of exit program routine.
1230-1240	Clear sprite definition memory.
1250-1290	Subroutine to change color of sprite.
1300-1340	Subroutine to draw the sprite assembly matrix.
1350-1390	Subroutine to flash cursor.
1400-1520	Prompt to input information about DATA statements.
1530-1790	Subroutine to convert sprite definition to DATA statements.
1800-1830	Exit subroutine.
1840-1880	Subroutine to find address of the end of merge program.
1890-1940	Display menu subroutine.
1950-2060	Main program loop for character editor.
2070-2220	Subroutine to monitor cursor position and change character behind cursor and character above assembly matrix.
2230-2240	Subroutine to display code of character.
2250-2320	Display character on printer subroutine.
2330	Clear character subroutine.
2340-2370	Display assembly matrix.
2380-2410	Subroutine to flash cursor.
2420-2610	Subroutine to create DATA statements for characters.
2620-2630	Clear keyboard buffer subroutine.

COMMODORE 64

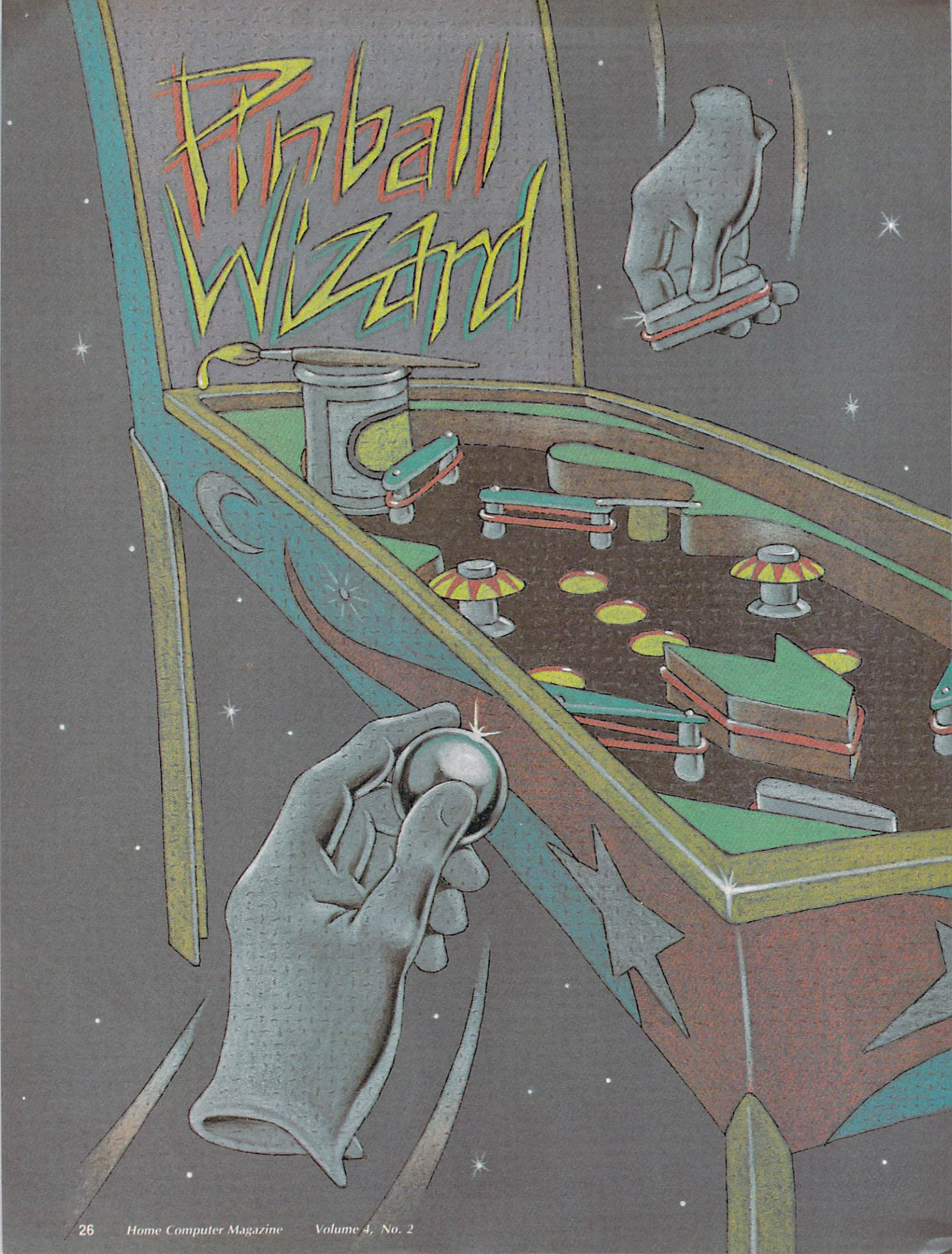
```
100 REM  ***********************
110 REM  *  GRAPHIC  EDITOR  *
120 REM  ***********************
130 REM  BY  JOHN P.  THRASHER
140 REM  HOME  COMPUTER  MAGAZINE
150 REM  VERSION  4.2.1
160 REM  C64  BASIC
170 POKE  52,56:POKE  56,56:CLR:PRINT"■SH
    IFT CLR■"
180 POKE  56334,PEEK(56334)AND254
190 POKE  1,PEEK(1)AND251
200 FOR I=0  TO 1015:POKE  I+14336,PEEK(I
    +53248):NEXT
210 POKE  1,PEEK(1)OR4
220 POKE  56334,PEEK(56334)OR1
230 POKE  53272,(PEEK(53272)AND240)OR14:
    POKE  53281,1:POKE  53280,1
240 FOR I=14848  TO  14848+(2*8)-1
250 READ  A:POKE  I,A:NEXT
260 DATA  255,129,129,129,129,129,129,25
    5,255,255,255,255,255,255,255,255
270 INPUT"■CTRL BLK■3CRSRDOWN■2CRSRR
    IGHT■USING JOYSTICKS Y OR N";J$:PRI
    NT"■SHIFT CLR■"
280 GOSUB  1400
290 DIM CR(1,2):S=54272:CL=0:GOTO  1890
```

COMMODORE 64

```
300 POKE  2040,240:POKE  53287,CL
310 POKE  53248,24:POKE  53249,225:POKE  5
    3269,1
320 GOSUB  1300:GOSUB  1230
330 GOSUB  1350:GOSUB  440
340 GOSUB  580
350 X=PEEK(197):IF  X=64  THEN  330
360 IF  X=59  THEN  1890
370 IF  X=39  THEN  880
380 IF  X=30  THEN  320
390 IF  X=20  THEN  1250
400 IF  X=41  THEN  990
410 IF  X=34  THEN  1530
420 IF  X=62  THEN  1190
430 GOTO  330
440 X1=0:Y1=0:FB=0
450 IF  J$="N"  THEN  510
460 XT%=PEEK(56320)AND31
470 X1=SGN(XT%  AND  4)  -  SGN(XT%  AND  8)
480 Y1=SGN(XT%  AND  1)  -  SGN(XT%  AND  2)
490 FB=1-SGN(XT%  AND  16)
500 RETURN
510 KY=PEEK(197):IF  KY=64  THEN  RETURN
520 IF  KY=13  THEN  X1=-1:RETURN
530 IF  KY=18  THEN  X1=1:RETURN
```

Continued on p. 47


TI-99/4A
THE EVERYTHING BOOK
For The TEXAS INSTRUMENTS 99/4A HOME COMPUTER
• HOW-TO Information on expanding your system
• GAMES - The newest, the best, the most
• GIANT Software Directory over 500 descriptions
• Peripherals and Accessories Hundreds of items: printers, joysticks, disk drives, plotters, modems, monitors & much more!
• DISCOUNTS! And Easy Order Info
GET IT FREE!
Just Call Us Toll-Free
1-800-348-2778
Or send your name and address to:
TENEX, Box 6578, South Bend, IN 46660
(219) 277-7726
© TENEX Computer Marketing Systems, Box 6578, South Bend, Indiana 46660


Pinball Wizard

REVIEW
—OF—

PINBALL CONSTRUCTION SET

by the *HCM Staff* with David Fulton and Joel Horwitz

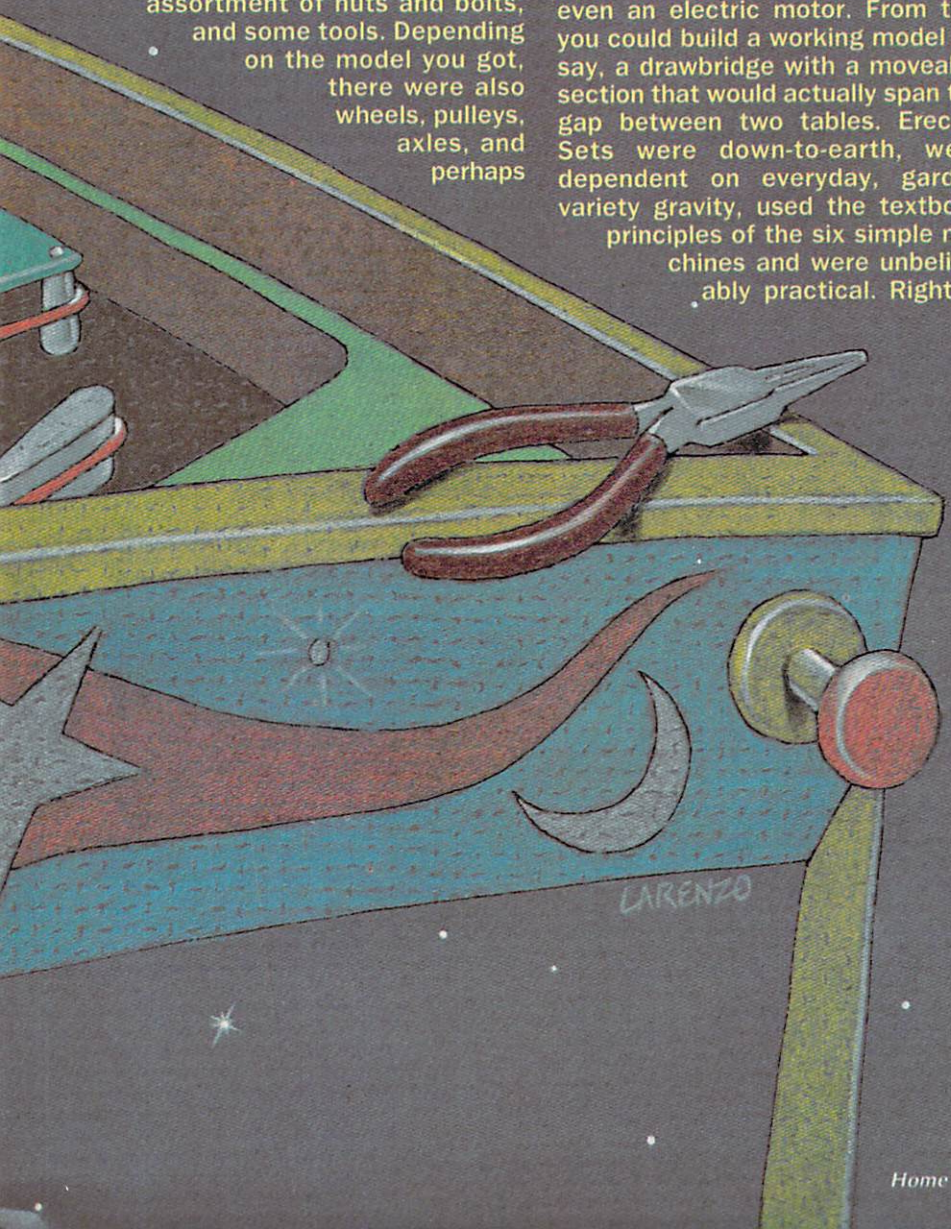
Ever since I was a young boy, I've played the silver ball. From New York down to Frisco, I must have played them all. But I ain't seen anything like this in any amusement hall. This Pinball Construction Set plays a mean pinball.

Name:	Pinball Construction Set
Program Type:	Game Development
Machine:	Apple II series and Commodore 64
Distributor:	Electronic Arts 2755 Campus Drive San Mateo, CA 94403
Price:	$40, Diskette

System Requirements:
Disk Drive, color monitor, joystick

	poor	fair	good	excellent
Performance				
Engrossment				
Documentation				

Long ago, in the Eisenhower era, those of us whose parents or relatives thought we were mechanically inclined—or ought to be—got Erector Sets as presents. You got a box filled with stamped and perforated metal girders in long, short and medium lengths, an assortment of nuts and bolts, and some tools. Depending on the model you got, there were also wheels, pulleys, axles, and perhaps even an electric motor. From this you could build a working model of, say, a drawbridge with a moveable section that would actually span the gap between two tables. Erector Sets were down-to-earth, were dependent on everyday, garden variety gravity, used the textbook principles of the six simple machines and were unbelievably practical. Right in tune with the era. No ventures into invisibility, alchemy, anti-gravity. *Pinball Construction Set* (*PCS*), from Electronic Arts, also gives you a load of parts and the tools for assembling them. But you can also change the forces working on the parts—or the properties of the parts themselves. You can change their colors—or make them invisible. And unlike new projects with Erector Sets, which meant tearing down old projects and reusing the parts in the new one, you can keep anything you build with *PCS* as long as you like, or as long as your disk budget holds out.



When you first boot up PCS, it puts graphics in the shape of the frame for an old-fashioned pinball machine on the left half of the video display. On the right half it puts a series of *icons* which represent all the parts you might find on the playing field of a pinball game. On the far right edge of the display, it has icons for a number of tools. With these "tools," you can move the parts from the "parts box" into the frame and construct your own pinball game.

Before you begin to make your own game with PCS, however, you'll want to load and play some of the DEMO game programs on the disk. This will give you a feel for the action of the various parts. Once you've done that, however, you'll want to return to the blank frame and begin dragging parts out onto the playing field. You'll see a hand in the upper left-hand corner of the screen. You can use the joystick to move the hand to any one of the parts. If you then hold down the fire button and move the joystick, the part will move with the joystick until you release the fire button. You can also pick up any of the tools along the far right edge of the screen and use them to alter the parts you've put on the screen.

Icons make *PCS* very easy to use. As you build your game, everything happens on the screen *as if* you were actually manipulating parts with

PINBALL CONSTRUCTION SEQUENCE

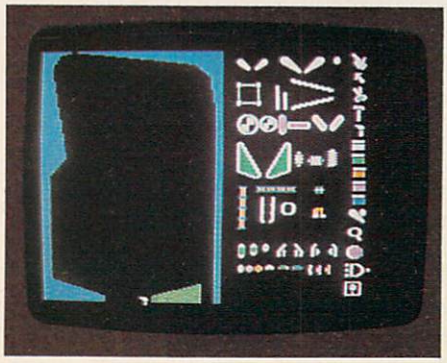
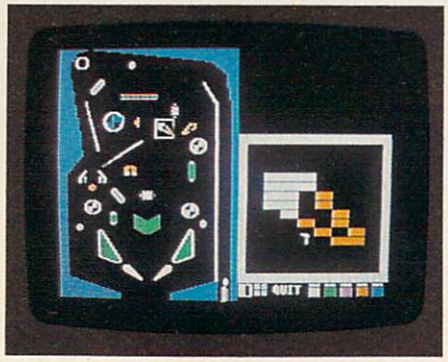


PHOTO 1 *By moving the hand over the paintbrush and pressing your joystick's fire button, you pick up the paintbrush. If you then move it to a color (below the paintbrush icon) and press the fire button again, you can then paint the elements on the screen in that color.*

PHOTO 4 *The magnifying glass (fourth icon from bottom, rightmost row, PHOTO 1) lets you move the square over any area and look at it in detail. With the magnifying glass on the screen, your tool is the paintbrush, which you can use to paint any part block by block.*



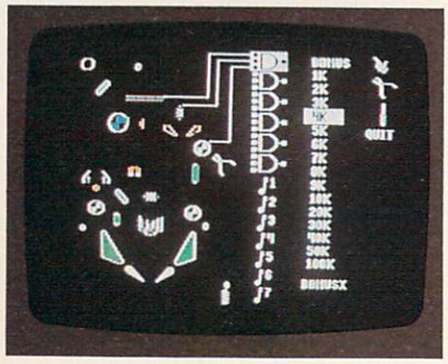


PHOTO 7 *The AND gates also connect parts together, and (when all the connected parts have been hit) give a bonus, which you can also set. The screwdriver connects parts; the pliers cut them free in case you change your mind.*

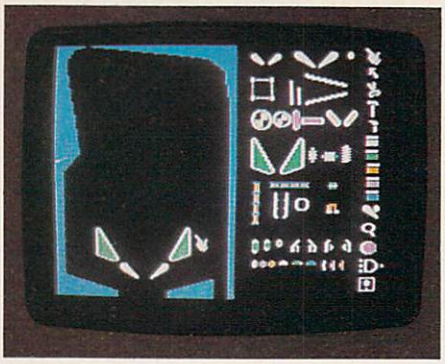
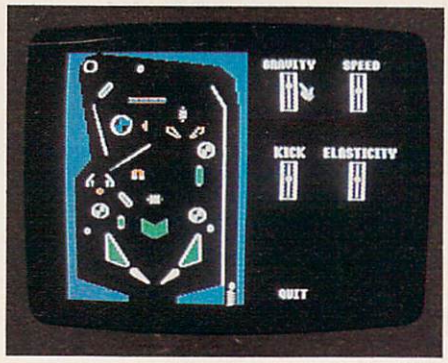


PHOTO 2 *When you move the hand to a part in the parts box on the right side of the screen and hold down the joystick's fire button, the hand picks that part up. If you hold the fire button down, the hand will move that part anywhere on the screen as you move the joystick.*

PHOTO 5 *Choosing the world icon (the circle third from the bottom on the right) lets you alter the response of the parts. You can reset gravity, the speed of the ball, the force with which the bumpers kick or the elasticity of the walls. You move the hand to an indicator, hold down the fire button and move the joystick.*



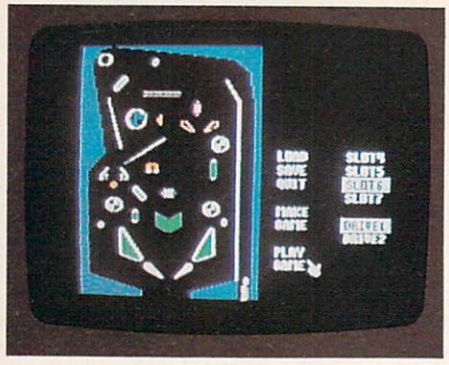


PHOTO 8 *Once you've put everything on the board, you'll want to play your game, of course. If you choose the ball and flipper icon (fourth from the bottom on the right, PHOTO 1), you can play one ball for testing purposes. If you choose the disk icon (bottom on the right, PHOTO 1), you can play a full game, save your game, or load a new one.*

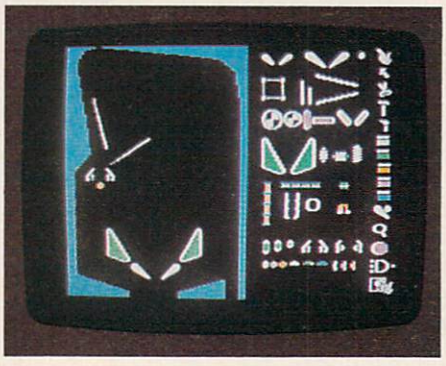
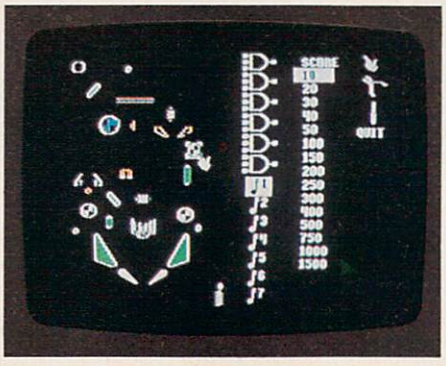


PHOTO 3 *The lines on the upper left corner of the playing field are actually boxes. After dragging them onto the field with the hand, you can move their corners around with the arrow. (In this game, the corners lie over one another, making the boxes look like lines.)*

PHOTO 6 *The odd-looking half-circle with three dots on one side (second icon from the bottom on the right, PHOTO 1), called an AND gate, lets you look at—and change, if you want—the score for any of the parts on your board. You can also change the sound generated when any part is hit.*



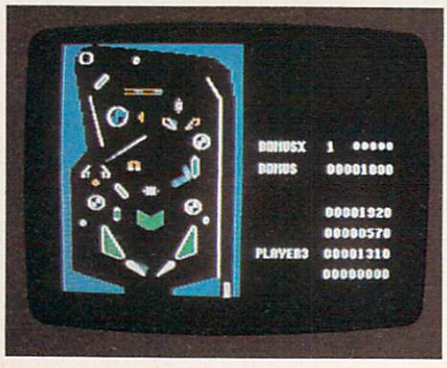


PHOTO 9 *After you've selected PLAY GAME, you use the space bar to choose the number of players. At that point,* Pinball Construction Set *works just like the pinball games you find in arcades: turn and turn about for all the players until they've used up all their balls.*

tools. There's no complicated code to decipher, no *x-y* coordinates for screen positions to figure out, no tedious shape definitions to generate, no interminable menus to run through. Pick up a part. Move it across the screen. Pick up a tool. Work on the part. Simple and straightforward. After you've put things together to your satisfaction, you can then make your computer simulate the action of the pinball game you've devised. The graphics are good, but the resolution of the computer you're using determines their sharpness. They seem to be somewhat better on the Apple than on the Commodore C-64. You'll notice that the parts change color if you move them just one pixel, due to the color resolution of the computer.

The response of the simulated pinball machine is very like the real thing. After a ball caroms off a bumper, it slows as it hits the apogee of its flight, hesitates slightly, and picks up speed as it falls again towards the bottom of the playing field. In the Commodore version of *PCS*, however, the balls sometimes had a tendency to hang up on the parts. By choosing the world icon, you can alter all of those responses. You can make the playing field seem like a pool of molasses, with cotton balls bouncing off tissue paper walls onto slo-o-o-w motion flippers.

The documentation for the game is surprisingly good. It is, first of all, informative, but beyond that it is clear and lively, without lapses into the technospeak that fills so many manuals. And in a world of manuals with paragraph 4.1.2.a followed by column after column of dull grey text, the layout and color artwork in this booklet are a delight. (One quibble: It doesn't have an index, and finding out how to set a game up for more than one player took some determined searching.) The irony is that you hardly need any documentation to develop and play games with this software. The intent here is to encourage participatory learning, with the DEMO games serving as scratch sheets on which to practice moving parts around, altering the world, changing the connections—all without having to invent a complete game from scratch.

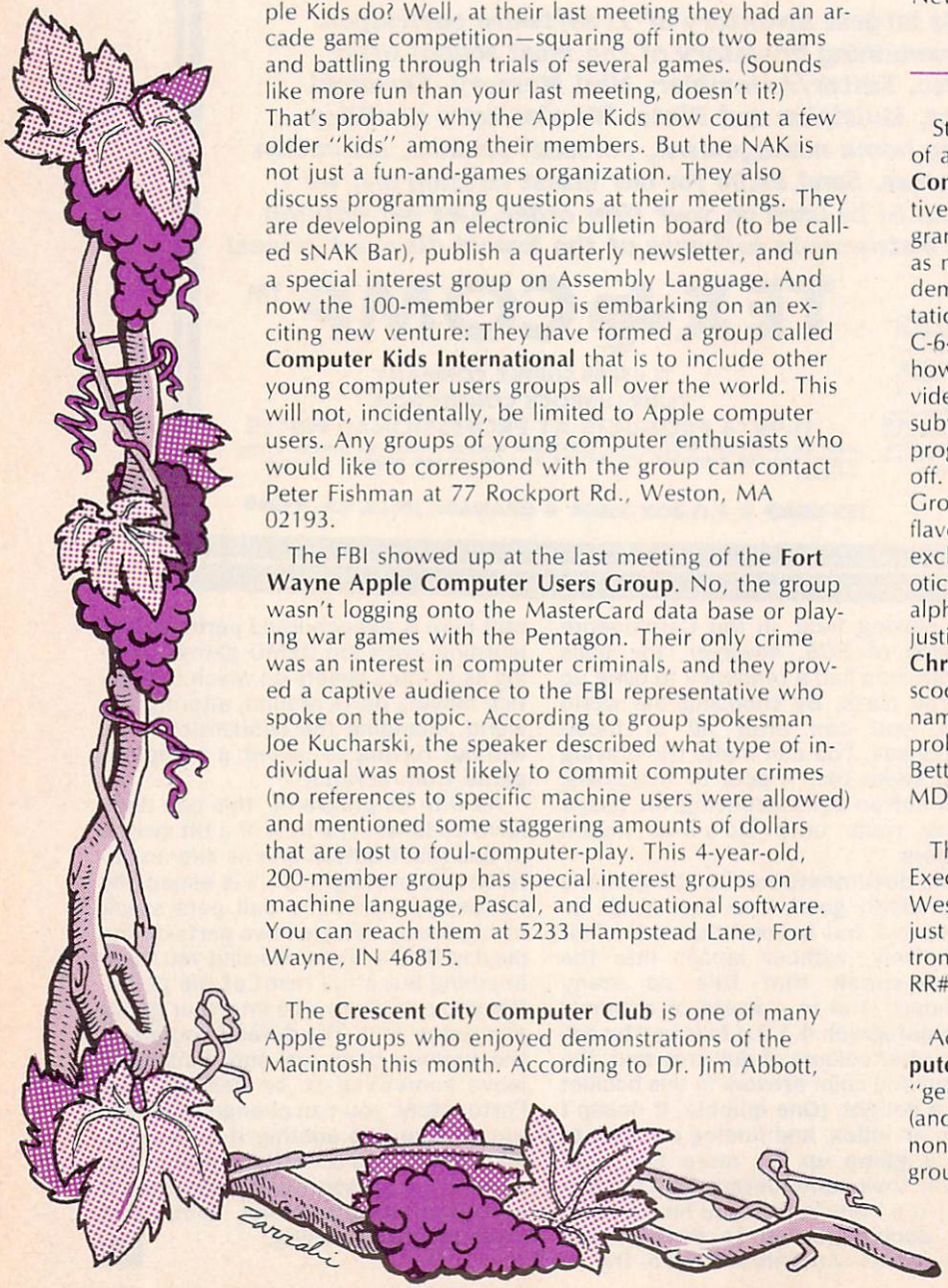
As with all programs, this one does have its flaws. The lack of a tilt switch or ball reset button seems like an unfortunate oversight. This is especially noticeable when the ball gets stuck mysteriously between two parts on the playing field without allowing you to do anything but a full reset of the game. But you can always go into your game and redesign it. The default sounds of the various bumpers and slingshots leave something to be desired, too. Fortunately, you can change from one preset sound to another if it irks you. All in all, *PCS* is something more than a video game, something less than a software development tool, and thoroughly engrossing.

HCM

Group Grapevine

News, information and upcoming events of home computer users groups around the world.

Looking to join a users group, exchange newsletters or software, increase your users group's membership or pep up your next meeting's agenda? For the latest users group news, put your ear to the Group Grapevine. And if you have a message to put out to other groups, if you are starting a new group, or have an interesting item to share, send a note or picture—or better yet, a group newsletter—to the Users Group Editor, Home Computer Magazine, 1500 Valley River Drive, Suite 250, Eugene, OR 97401, (503) 485-8796.



Raines Cohen started the **Newton Apple Kids** because the adult Apple users groups in his area were too business-oriented. What kinds of things do the Apple Kids do? Well, at their last meeting they had an arcade game competition—squaring off into two teams and battling through trials of several games. (Sounds like more fun than your last meeting, doesn't it?) That's probably why the Apple Kids now count a few older "kids" among their members. But the NAK is not just a fun-and-games organization. They also discuss programming questions at their meetings. They are developing an electronic bulletin board (to be called sNAK Bar), publish a quarterly newsletter, and run a special interest group on Assembly Language. And now the 100-member group is embarking on an exciting new venture: They have formed a group called **Computer Kids International** that is to include other young computer users groups all over the world. This will not, incidentally, be limited to Apple computer users. Any groups of young computer enthusiasts who would like to correspond with the group can contact Peter Fishman at 77 Rockport Rd., Weston, MA 02193.

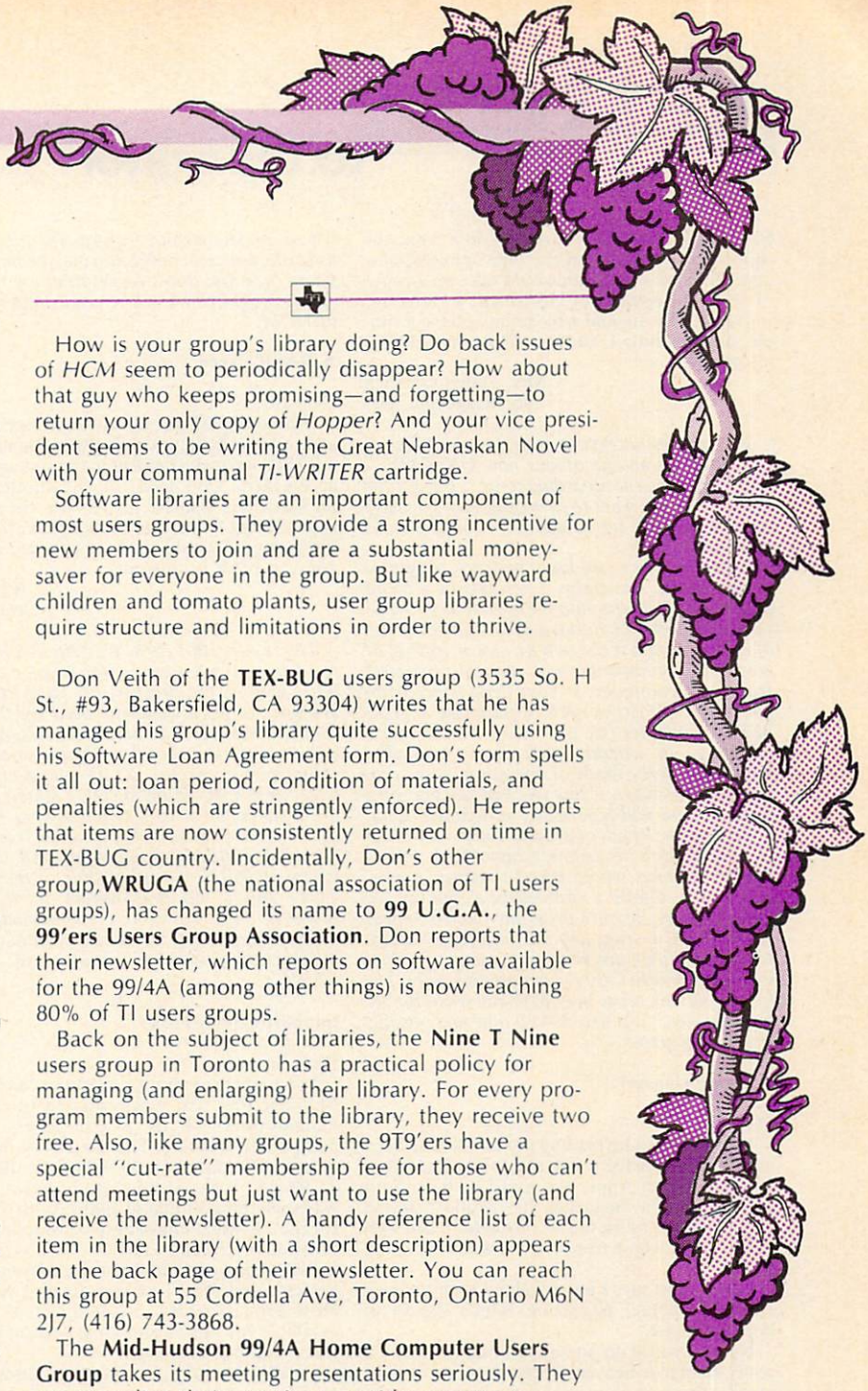
The FBI showed up at the last meeting of the **Fort Wayne Apple Computer Users Group**. No, the group wasn't logging onto the MasterCard data base or playing war games with the Pentagon. Their only crime was an interest in computer criminals, and they proved a captive audience to the FBI representative who spoke on the topic. According to group spokesman Joe Kucharski, the speaker described what type of individual was most likely to commit computer crimes (no references to specific machine users were allowed) and mentioned some staggering amounts of dollars that are lost to foul-computer-play. This 4-year-old, 200-member group has special interest groups on machine language, Pascal, and educational software. You can reach them at 5233 Hampstead Lane, Fort Wayne, IN 46815.

The **Crescent City Computer Club** is one of many Apple groups who enjoyed demonstrations of the Macintosh this month. According to Dr. Jim Abbott, the group was quite favorably impressed with the Macintosh's ability to jump from operations in one mode to another—allowing you to slip from your word processor program into a statistics program into a graphics mode without skipping a beat. The Crescent City Computer Club is a conglomerate organization of several machine-specific users groups. The group has been around since the "old days," 1976. As Abbott says, "I can remember how excited we were when we saw our first memory dump." Among the many projects the group has undertaken are the development of an intelligent keyboard and assisting a young computer whiz who is constructing his own robot. Contact the group in care of Dr. Abbott at the Department of Mathematics, University of New Orleans, New Orleans, LA 70148, (504) 283-3819.

Standing room only is the description we received of a typical meeting of the **Compucats' Commodore Computer Club** of Aberdeen, MD. This large and active group draws in the crowds with its dynamic programs and demonstrations. At each meeting there are as many as 10 machines up and running for demonstrating and trying out software. Recent presentations include: a demonstration of *Solo Flight* (for the C-64) by MicroProse, a lecture and demonstration on how to integrate your home computer with your videocassette recorder to superimpose captions and subtitles on the screen, and a presentation on using program characters to simulate the space shuttle takeoff. The group is located near the Aberdeen Proving Grounds military facility so there is an international flavor to its membership. This leads to an interesting exchange of ideas, and exposes members to such exotic creations as a program to print the Greek alphabet. In addition, these computing felines are justifiably proud of their newsletter **Compucat Chronicles** which has, according to them, managed to scoop some of the "big magazines" (please, no names) on such stories as the Commodore disk drive problem. To reach these crazy compucats, contact Betty Jane Schueler at 680 W. Bel Air Ave., Aberdeen, MD 21001, (301) 272-0472.

The **Kankakee Hackers** saw a demonstration of the Executive 64 at their last meeting. According to Rich Westerman, this suited the business-oriented hackers just fine. Future projects for the group include an electronic bulletin board. You can reach this group at RR#1, Box 279, St. Anne, IL 60964, (815) 933-4407.

According to Jim Mather of the **Westmoreland Computer Users Group**, a users group must have a "generic" title in order to qualify for non-profit status (and all the special privileges that includes)—hence the non-specific title of this distinctly Commodore-ic group. At their last meeting, members enjoyed a



demonstration and talk on disk maintenance by a representative from *Verbatim*. Disk drive maintenance is the topic of their upcoming meeting. This 230+ member group has held workshops on such topics as machine language, spreadsheets, and even building your own computer. Contact them at 3021 Ben Venue Dr., Greensburg, PA 15601, (412) 836-2224.

The Koala Pad was the focus of the last meeting of the **Mid-Missouri Commodore Club**. A local artist was on hand to demonstrate how the popular pad could assist in illustrating several nursery rhymes for children. At their next meeting, the group will see a demonstration of a machine language music program that displays musical notes on the screen as it plays. You can reach this group in care of Jim Bishop at 1804 Vandiver Dr., Columbia, MO 65201, (314) 474-4511.

The **Commodore-64 Users Group** in Clearfield, Utah, got down to basics at their last meeting with a lecture on binary numbers. They also enjoyed a presentation by Bruce Carver of Access software (the developer of such programs as *Beach Head, Neutral Zone*, and *Spritemaster 64*). Their next meeting will be centered on the theme of data bases. Demonstrations of Mirage Concepts' data base manager, *Flex File, Omni File*, and *Info Base 64* are bound to get this theme party off to a rousing start. Those interested can contact the group through Rodney Keller at 652 West 700 North, Clearfield, UT 84015, (801) 776-3950.

PC JR

With regard to the PCjr, Gary Wilcox, of the **IBM PC Users Group** in Des Moines, Iowa, is typical of the PC users group people we have heard from. He is taking a wait-and-see attitude as to whether or not his PC group will incorporate PCjr users. There are many pros and cons: First of all, most PC owners are perfectly happy with their machines and not especially enthusiastic about allotting meeting time to the PCjr—except to discuss such topics as whether or not PCjr software will be compatible with the PC. Second, as a member of the **Modesto-Turlock PC Users** pointed out, most PC users are home users who are fairly savvy about using their machines. She doubts they will have much in common with the many neophyte computer users the PCjr is bound to draw in.

On the other hand, it *will* draw them in, and an infusion of many, many members can mean exciting things for a user group. Do PC groups want to miss out on an influx of new, eager-to-learn IBM users? After all, they are members of the same family. It remains to be seen whether the PCjr will find a place on the PC groups' agenda or spawn its own network of users groups . We haven't heard of any PCjr-specific groups yet, but we will report on them here as soon as we do. Any PC groups with new insight into the matter are encouraged to write or call. We'll be happy to report your viewpoint in the next installment of Group Grapevine.

How is your group's library doing? Do back issues of *HCM* seem to periodically disappear? How about that guy who keeps promising—and forgetting—to return your only copy of *Hopper*? And your vice president seems to be writing the Great Nebraskan Novel with your communal *TI-WRITER* cartridge.

Software libraries are an important component of most users groups. They provide a strong incentive for new members to join and are a substantial money-saver for everyone in the group. But like wayward children and tomato plants, user group libraries require structure and limitations in order to thrive.

Don Veith of the **TEX-BUG** users group (3535 So. H St., #93, Bakersfield, CA 93304) writes that he has managed his group's library quite successfully using his Software Loan Agreement form. Don's form spells it all out: loan period, condition of materials, and penalties (which are stringently enforced). He reports that items are now consistently returned on time in TEX-BUG country. Incidentally, Don's other group,**WRUGA** (the national association of TI users groups), has changed its name to **99 U.G.A.**, the **99'ers Users Group Association**. Don reports that their newsletter, which reports on software available for the 99/4A (among other things) is now reaching 80% of TI users groups.

Back on the subject of libraries, the **Nine T Nine** users group in Toronto has a practical policy for managing (and enlarging) their library. For every program members submit to the library, they receive two free. Also, like many groups, the 9T9'ers have a special "cut-rate" membership fee for those who can't attend meetings but just want to use the library (and receive the newsletter). A handy reference list of each item in the library (with a short description) appears on the back page of their newsletter. You can reach this group at 55 Cordella Ave, Toronto, Ontario M6N 2J7, (416) 743-3868.

The **Mid-Hudson 99/4A Home Computer Users Group** takes its meeting presentations seriously. They are recording their meetings on videocassette so members can improve their presentations. This means members who can't attend meetings can enjoy the proceedings at home (and make fun of their friends' oratorial styles). It also provides an indisputable record of who volunteered for what, and a reference library of presentations by outside speakers. The group's address is RD 1 Box 359-5, Kingston, NY 12401.

Finally, a major event is in the works for the large and powerful **Cin-Day Users Group** in Cincinnati. They are preparing to hold the first annual *Midwest Computer Fest* in the fall. Group president Ed York is inviting all dealers, distributors, and manufacturers of products for the 99/4A to contact him about showing their wares at the show. And if they will be unable to attend, he will try to arrange for a "substitute rep" to show their product for them. Ed is expecting a big (2000 or more) turnout, so contact him for details at P.O. Box 519, West Chester, OH 45069, (513) 777-0110.

HCM

Letters to the Editor

... from p. 7

60-minute tape? Do you also know what the equivalent would be on the tape counter, how many lines of a program would take up a count of 50 on the counter or vice-versa? I know this sounds confusing and may actually have a simple answer but I cannot find any of this information.

Mrs. Dawn Gorsuch
No. Ridgeville, OH

The simplest answer to your question, Dawn, is to take advantage of our new ON TAPE™ offer and leave the recording to us... But for the cases where you want to save programs you have written, we will try to answer your questions more specifically.

First, the space you leave between programs is not critical. Generally, 5 seconds or a count of 3 or 4 on the tape counter is sufficient. Next, your tape deck will automatically stop at the end of each program if the remote motor control of your deck is compatible with the remote control from your computer. If your deck is not compatible, you'll either need to purchase a TEXSETTE™ adapter [see page 125] to insure compatibility, or operate your deck manually. Regardless of the mode of operation, your computer knows when it has reached the end of a program and will ignore any subsequent data.

Because the length of program lines vary greatly, it's difficult to estimate how much time a number of lines would take to record. A program can be almost 3 minutes (or a count of 70) long. Usually, feature programs which appear in HCM will average about 2 minutes in length. It is a good idea not to use a cassette tape that is larger than size C-30 (15 minutes per side). The longer playing tapes may be made with thinner material which can stretch with time and possibly cause data errors.

A Home Convert

Dear Sir:

You've got to be reading my mind! I've been using the 99/4A for two years, but in December I purchased an Apple IIe, and I've been both searching for a magazine that would help me with the Apple as the old *99'er Magazine* did with the TI, and wondering why I wasn't receiving *99'er*.

Needless to say, I am delighted with the new *Home Computer Magazine*, which speaks to both my needs.

Now, if you'd do one more thing for me, I'd really be in byte heaven. Please do some of your excellent articles showing how to translate from TI BASIC and Extended BASIC into Applesoft and Integer BASIC. I have a load of TI programs (many of them from your magazine) that I'd like to translate, but I need help.

Again, thanks for your newest metamorphosis.

George Perrine
Louisville, KY 40243

George, you are not the only reader with more than one home computer. We have heard from many users who own either Commodore or Apple and TI. To produce HCM with its in-depth coverage of all these machines costs a bit more, but you're worth it!

To help you in translating TI BASIC to AppleSoft BASIC, we recommend the language conversion article AppleSoft to TI BASIC *which appears in chapter three of our new book* Best of 99'er Vol. 1 *(see page 73). This article describes the major differences between the two languages, and how to resolve those that can be resolved. It will put you well on your way.*

Translating programs from one machine to another is always a challenge. In some cases, where special machine features are utilized (such as sprites), direct translation may be impossible. Good luck and don't forget to enjoy the many Apple programs we will be publishing right here in HCM.

A BASIC Need

Dear Sir:

Why did IBM put all of those nice graphics and sound capabilities in the PCjr and not supply a BASIC language resident in the system that can access them. Do I need to spend another $75.00 on Cartridge BASIC just to use the system's capabilities? Is there any way to get around this limitation? I would really appreciate any help you can offer in this area.

Albert Rice
Seattle, WA

The reason IBM chose Cassette BASIC as the resident system language was to keep compatibility with its big brother, the IBM PC. There are ways to get around the need for Cartridge BASIC, however, and access the system's capabilities. In this issue, there is an article which shows how to access the TI sound chip from Cassette BASIC using the OUT statement. The OUT statement is to I/O ports what POKE is to a memory address. Many other features are accessible with this statement, and in future issues we will be covering some of these areas. There still is no replacement for Cartridge BASIC though. The powerful commands offered in Cartridge BASIC would take a lot of code and programming time to simulate in Cassette BASIC.

Support Comes Forth

Dear Sir:

I recently read about something called an AP Modular Pak that was supposed to convert a Commodore 64 to a completely Apple-compatible machine. This may be of interest to your C-64 readers, but it is of little use for my TI-99/4A. I was wondering, however, if there was something similar available or in the works for the TI. Also, I have been considering adding Forth to my TI and I noted several versions for the 99-4A in your latest issue (one from Wycove Systems and the TI version from Tex-Comp). How about a review of the different versions available for the TI and an article on this very fast computer language.

Finally, I would just like to say how much I enjoy your new magazine format and that I'm glad to see you back. I enjoyed reading about computer systems other than my own, and, as you promised, there is no loss of TI material from *HCM*. I'm glad to see third-party distributors fill in the gap left by TI's departure. I hope that the market place will now be more open for new third party software and especially hardware for the TI, and I hope that your new advertisers that specialize in Apple, Commodore and IBM will bring their expertise to bear on products for the TI-99/4A.

Stephen Foley
Lexington, KY 40502

Thank you for your kind words, Stephen. We too share your hope for increased third-party support. Watch for future language reviews.

Support Needed Down Under Too

Dear Sir:

I am writing on behalf of the TISHUG (TI Sydney Home Computer Users Group). We have just heard from TI Australia that they shall keep up service for at least 5 years. We know that TI depends on its customers, so we have put together a petition and got it signed by all our members and given it to TI to make sure that they keep their promise of keeping up service. Getting to the point of this letter, we would like to see other users groups send petitions to TI in the USA.

Now writing on behalf of myself, I would like to compliment you on the outstanding standard of this magazine. I have read a back issue I had lying around and saw the listing of the "Poor Man's Program Loader." I found it very useful.

As you might probably notice, this letter has been printed with the TI-WRITER and my newly bought FX-80 printer. I was wondering if you could tell me how to access some of the printer's special functions such as proportional spacing with the TI-WRITER.

Again I would like to commend you for your wonderful magazine.

Steven Shraibman
Sydney Australia

Steven, we have found TI to be good with support for all of their products, and have no cause to think otherwise with the Home Computer. On the other hand, it can't hurt to let them know that the support is needed and appreciated.

The TI-Writer does not support proportional spacing directly, but the printer's special control codes could be added to the file to be printed, indirectly producing proportional spacing. Study the "transliterate" command in the Text Formatter section of the manual. Greetings to all the members of TISHUG down under!

Guns of Navarone Blast Us

Dear Sir:

In your recent review of "Henpecked," Navarone Industries was listed as the distributor, but the address is not our address. Please let your readers know that our correct address is:

Navarone Industries
510 Lawrence Expressway 800
Sunnyvale, CA. 94086

In the credits you state a System Requirement to be assembly language, but later in the review mention that "the program is a good example of the stunning graphics that can be achieved with a self-contained ROM cartridge which requires no additional memory packages, disk drives, or expensive peripherals."

Henpecked is now also available through our new "RENT n' PLAY" Club.

Chuck Humphrey, President
Navarone Industries

We stand corrected, Chuck. Thank you for bringing the two errors to our attention. Indeed, Henpecked is provided in a cartridge that will work on the TI-99/4A without any other attachments (except optional joysticks).

A LOGO Source

Dear Sir:

Concerning your article in the *LOGO Times* section on *Lyrical LOGO* (Volume 4 Number 1). Would you please tell me how I can obtain the TI version.

Mary Presley
Nowata, OK

Mary, the back issues of 99'er Magazine *that continued the original 99/4A version are collector's items now, but you can find the TI LOGO Poet in* The Best of 99'er *(page 113) along with other great LOGO programs as well. Look in some of the larger chain book stores for a copy or see the ad in this issue of HCM to order directly from us.* HCM

SPRITEMASTER 64

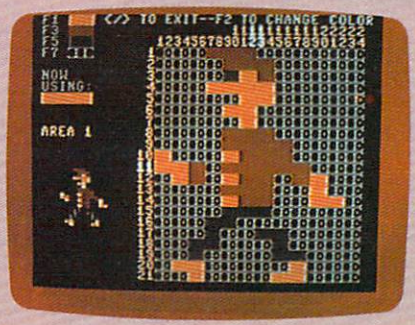
By John P. Thrasher
HCM Staff

Spritemaster 64 is a sprite generator and editor for the Commodore 64, available on tape or diskette. This program makes the tedious task of developing sprite definitions by the old pencil-and-paper method easy, interesting, and convenient. *Spritemaster 64* can generate several sprites and copy their definitions to a file to be read later by your own program. You can also produce a series of sprites to represent an object in motion and then display this sequence with a unique ANIMATE command. This command lets you put each sprite on the screen in rapid succession to create animation.

You can modify your sprite's color, size and shape to see how it will appear under various conditions. Menu-driven options give you fingertip control of every conceivable sprite register configuration ranging from single or multicolored sprites, to screen and background color, to vertical and horizontal sprite expansion.

Name:	Spritemaster 64
Program Type:	Graphics Generator
Machine:	Commodore 64
Distributor:	Access Software 925 East 900 South Salt Lake City, UT 84105
Price:	$34.95, cassette $34.95, diskette

	poor	fair	good	excellent
Performance				
Ease of Use				
Documentation				



Bring Your Imagination to Life

Spritemaster 64 could very well inspire you to new artistic heights. It lets you design several still pictures in a sequence, then display them in rapid succession to bring your creation to life. This animation technique can be practiced for its learning value alone or applied to your own programs for a new dimension in graphics.

The documentation recommends this program for C-64 users from ages 8 to adult, and indeed there is something for everyone here. I would say, though, that while children can surely learn from the exciting animation techniques, it would be too much to expect the average 8-year-old to be able to understand and implement the intricacies of this sprite utility. Suffice it to say it would be absolutely necessary for a parent or instructor to play a very interactive role when a child sits down with *Spritemaster*.

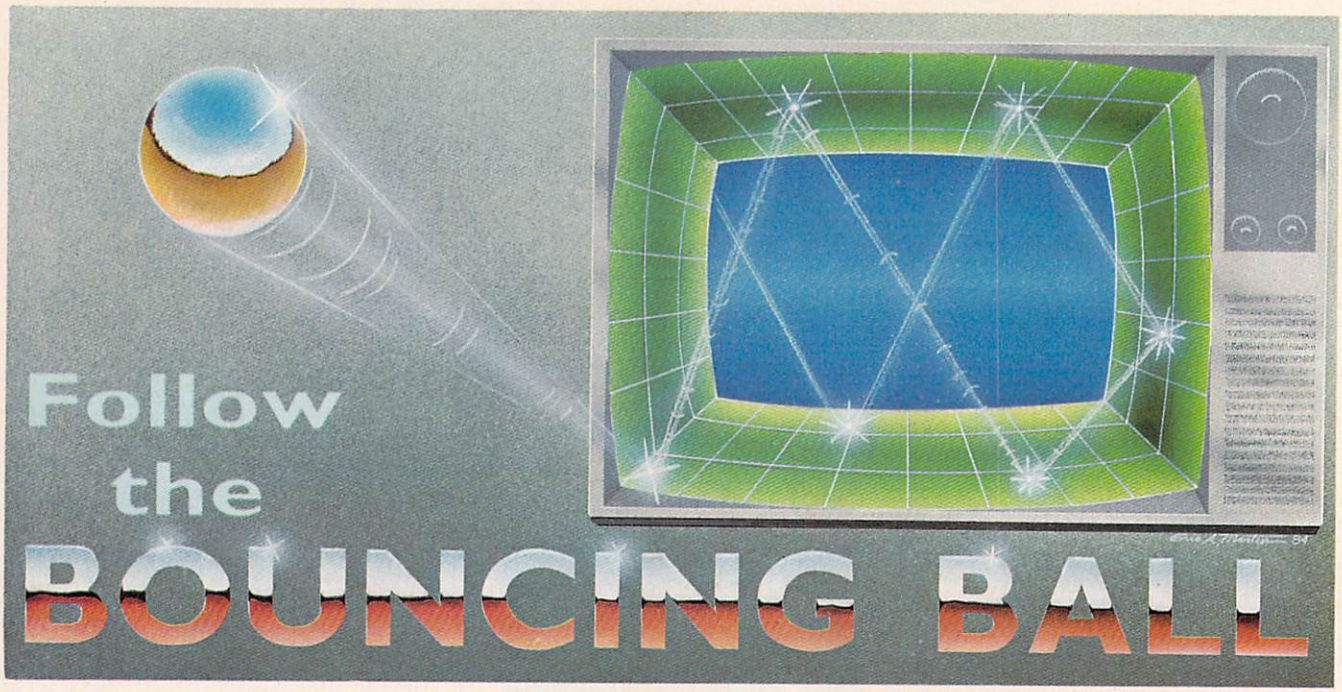
This is one program that works as advertised. Through menu-driven commands, in-depth documentation, and test files, the first-time user is led through a sprite development exercise that is both intriguing and satisfying. With a little practice and experimentation, you will be able to use *Spritemaster 64*'s many features to create exciting, animated sprite graphics for your own sprite-hungry programs.

Sprites by the Book

The users manual accompanying this program is thorough and precise. It does, however, recommend augmenting the documentation with the *Commodore 64 Programmer's Reference Manual*, as it rightly should.

The information in *Spritemaster 64*'s users manual is presented in several chapters, which flow in a logical progression. It begins with a brief introduction and then tells you, step by step, how to load the program and get started. Once this is accomplished, the manual demonstrates to the novice how to use and manipulate the ready-made sprite file that accompanies the program. The next chapter covers all the steps necessary to create animated sprites. Each command in the program's repertoire is clearly defined. Finally, there is a reference section that presents an in-depth look at the Video Display Chip as it pertains to sprite graphics, and some helpful examples for using sprite graphics in your programs.

Both the program and documentation are well thought-out and professionally presented. Sprite animation is a topic that is not usually emphasized in other sprite editor programs, and the ability to create and store sprite definitions in a program-readable file is widely neglected as well. All in all, *Spritemaster 64* could easily become a very handy addition to your library of programming aids.



by Randy Wilson

There are many of us who choose a computer for its graphics alone. We carefully take in the store demonstrations until we are sufficiently impressed to trade money for machine; then we confront the task of creating graphics equal to those which first impressed us. In this article we'll explore the fundamentals of making a shape (a ball) with Applesoft BASIC, and giving this shape "intelligent" motion. In other words, the shape will simulate the movement of a real ball.

[This article expands on some of the ideas introduced in *Colorbounce*, a program that appears on pages 89-97 of the *Applesoft Tutorial* manual.—*Ed.*]

To start, boot APPLE IIe DOS, and then type in the following lines:

```
10 GR
20 M=.2: X=1: Y=3: W=X: Z=Y
30 VX=1.001: VY=1: D=3: E=0
90 REM
110 VY=VY+M
120 Y=Y+VY: IF Y>39 OR Y<0 THEN Y=Y-VY:
    VY=-VY
140 COLOR=E:PLOT W,Z: COLOR=D: PLOT X,Y
150 W=X: Z=Y: GOTO 90
```

Run the program, and you'll see a square "ball" bouncing on the left side of the screen. Let's examine each line to see how Applesoft carries out this action.

How It Works

In line 10, the command GR puts the computer in low resolution graphics mode. This mode gives you a black screen measuring 40 columns by 40 rows. The upper left-hand corner corresponds to column 0, row 0. The bottom row number is 39 and the far right column number is 39. Between row 39 and the bottom of the screen, is room for 4 lines of text.

In lines 20-30 we initialize some of the program variables. The variable M represents the effects of gravity. Variables X and Y mark the ball's position (X is the column number and Y the row number). Because the value of X and Y change throughout the program, we have to set up two variables to temporarily store the last value of X and Y. For this purpose, W is set equal to X, and Z equal to Y. We use VX as the horizontal velocity and VY as the vertical velocity. The color variables D and E are initialized to 3 and 0 respectively—3 being the color code for violet, and 0 the code for black in Applesoft BASIC.

Line 90 is only a marker that lets the program transfer from there to line 110. Later, we'll add instructions to line 90.

In line 110 we add the pull of gravity to the ball's vertical velocity.

Line 120 carries out three tasks: First, it computes the new horizontal position of the ball. Next, it checks to see if this new position is at the uppermost row (Y<0) or the bottom row (Y>39). Finally, if the new value is less than 0 or greater than 39, the sign of the vertical velocity is changed.

The commands in line 140 make the ball bounce by alternating its colors. The low-resolution graphics mode offers 16 colors. Because the background color is black, we make the ball transparent with the COLOR=E command. Then we plot the ball at its original starting position. Next we color the ball violet and plot it at a new position determined by lines 110 and 120.

Our variables are updated in line 150. Variables W and Z now hold the new position of the ball. Therefore, the next time line 140 is executed, the ball will be erased by the COLOR=E statement. Then the ball will be rePLOTed at the new position determined by lines 110 and 120. Finally, the program branches to line 90. To break the program, press [CONTROL][C]. To list it, type TEXT. Then type LIST.

Bouncing to the Side

Now let's add line 100 to give the ball some horizontal velocity. To do this, we'll have to update X, the horizontal position, each time through the loop. Type line 100:

```
100 X=X+VX: IF X>39 OR X<0 THEN X=X-VX:
    VX=-VX
```

Note that the ball now moves from side to side. Examining line 100, we see that the algorithm for horizontal motion is the same as the vertical motion scheme in line 120. Line 100 checks to find out if the ball is at the far left (X<0) or right (X>39) side of the screen. If either condition is true, the sign of the horizontal velocity is changed, reversing the direction of the ball.

Our last change for this low-resolution graphics program is to make the ball leave a trail. We change variable E to keep the ball from being erased each time line 140 executes. Instead, we want the images to remain, making the trail. In line 30, change the value of E from 0 to 2. Then line 90 becomes:

```
90 C=C+1: IF C>760 THEN E=D: D=INT(16*RND(1)):
   C=0: IF D=E THEN D=D+1
```

Because variable C is not initialized, Applesoft will set C equal to 0. Line 90 first increments C; then Applesoft checks to see if C is greater than 760, the number of PLOTs necessary to fill the screen. As soon as C is greater than 760, the color of the trail turns the same color as the ball. Also, the ball is randomly given a new color. If the ball's new and old color are the same, 1 is added to the variable D so that the ball will always change color. If the old and random value for D is 15, adding 1 yields 16, which is equal to 0 because Applesoft BASIC's COLOR variables are modulo 15 (modulo X being the remainder of any number divided by X + 1). Applied here, the number immediately following modulo is the largest number that can be expressed in that modulus. As soon as that number goes beyond 15, the number, designated N, becomes N−16. That's as far as we'll go with low-resolution graphics, so you might want to save this program on disk.

Getting High Resolution

Now, let's look at our program in high-resolution mode. The initial changes are simple. In line 10, change GR to HGR. The COLORs and PLOTs in line 140 should be changed to HCOLORS and HPLOTS. Finally, GOTO 90 should read GOTO 100. The modified lines are as follows:

```
10 HGR
140 HCOLOR=E:HPLOT W,Z: HCOLOR=D: HPLOT X,Y
150 W=X: Z=Y: GOTO 100
```

Delete line 90 by typing 90 and pressing [RETURN]. Also, change the value of E in line 30 back to 0: E=0. Resetting this variable will once again make the trail invisible. Let's see how these new lines will affect our program.

In line 10 the HGR command tells Applesoft to enter the high-resolution mode. In this mode the screen resolution is normally 280 columns by 160 rows, so the limit checks in lines 100 and 120 must be changed to accommodate these new parameters.

Line 140 in high-resolution acts as it did in low-resolution. Note that high-resolution graphics allow only 6 colors.

Because we removed line 90, line 150 must now force the program to branch to line 100. (Line 90 is no longer necessary to keep track of loop iterations.)

On the Rebound

Now let's modify our limit checks in lines 100 and 120. In addition to these changes, we'll add a "bounce factor" into our equations, taking into account that a real ball loses some energy on each bounce until it comes to a rest. We'll call this variable R and insert the following lines:

```
90 R=.9
100 X=X+VX: IF X>279 OR X<1 THEN X=X-VX:
    VX=-VX*R
120 Y=Y+VY: IF Y>159 OR Y<1 THEN Y=Y-VY:
    VY=-VY*R
```

Now the ball should rebound about ten percent less on each bounce. To add some realism, we'll plot walls for the ball to bounce off. Change line 10 to:

```
10 HGR: HCOLOR=3: HPLOT 0,0 TO 0,159 TO 279,159 TO
   279,0 TO 0,0
```

Bouncing Ball

Examine the new features of this line. Before any graphic command can be accurately executed in Applesoft BASIC, it's always necessary to define the color for graphics. In this case, the color for the walls will be white. The command HPLOT stands for High-resolution PLOT. This command PLOTs a line from one pair of points (column 0,row 0) TO the next pair of points (column 0,row 159). If you follow each set of points with another TO statement and set of points, Applesoft will draw a line from the first set of points to the next. If you include enough points and TO statements, you can enclose an area on the screen. Now, type RUN.

Taking Control

If you would like to take a more active part in the program, you can add keyboard controls. They give the ball a boost in a particular direction. First, change line 40 to include the keyboard control variable, KC, then type the following lines.

```
40 R=.9: KC=1
130 KB=PEEK(-16384): IF KB>128 THEN POKE -16368,0:
    GOSUB 200
200 IF KB=160 THEN VY=VY/R: VX=VX/R: RETURN
210 IF KB=141 THEN VY=VY*R: VX=VX*R: RETURN
220 IF KB=193 THEN VY=VY-KC: RETURN
230 IF KB=218 THEN VY=VY+KC: RETURN
240 IF KB=149 THEN VX=VX+KC: RETURN
250 IF KB=136 THEN VX=VX-KC: RETURN
260 IF KB=209 THEN POP: GOTO 10
290 RETURN
```

We can see how these new lines affect the program: Starting at line 130, Applesoft BASIC looks at or PEEKs memory location −16384. This location contains the code of the key pushed most recently. This code is the ASCII value of the character added to 128. If this memory location contains a value greater than 128, Applesoft will POKE—that is, write—a 0 at location −16368. This POKE will clear the high order bit at −16384 so that the computer can read the next keyboard character. After preparing to receive the next character, the program jumps to the subroutine starting in line 200.

If the key pressed is the space bar (160 − 128 = 32), then the vertical and horizontal velocity are divided by .9. This will increase both values, and the ball will move faster. If the [RETURN] key(141 − 128 = 13) is pressed, then the horizontal and vertical velocity are multiplied by .9. This decreases both values, moving the ball more slowly.

Pressing the A key (193 − 128 = 65) decreases the vertical velocity, causing the ball to move up. The Z key (218 − 128 = 90) increases the vertical velocity, and the ball drops.

If the right arrow key (149 − 128 = 21) is pressed, then the horizontal velocity is increased. This causes the ball to move to the right. The left arrow key(136 − 128 = 8) decreases the horizontal velocity and moves the ball to the left.

If the key pressed is Q(209 − 128 = 81), the system executes a POP command. To understand POP, we have to see how Applesoft executes a GOSUB. To set up a GOSUB, it must first write itself a note to mark the location of the GOSUB. That way, it can RETURN to the right place in the program. In computer jargon this note is called the control stack, and the correct point of return is the return address. The POP command throws away the most recent return address because, due to program flow, it is no longer needed. Therefore, when the Q key is pressed, Applesoft doesn't need to know where to return because we're going to send it somewhere else. The last statement on line 260 directs Applesoft back to line 10 where program execution starts from the beginning.

The purpose of line 290 is to direct program flow to the Applesoft statement immediately after the GOSUB if the key pressed is not one of those listed above, or if no key is pressed. Now try running the program using these options.

Plotting for Size

The last programming technique we'll investigate is the use of shape tables to determine the ball's size. Applesoft controls its graphics by utilizing a consecutive group of bytes that make up the shape table. Each byte is divided into three sections. The information contained in these sections tells the computer either to move one pixel or to plot a point and then move a pixel. This information is known as a plotting vector. Further, Applesoft labels each of these fields according to Figure 1.

C		B			A		
7	6	5	4	3	2	1	0

PLOTTING VECTOR BYTE

Figure 1

As we begin to see how different configurations of these bits control the graphics, note that the C field has only two bit positions. As you might guess, having only two bit positions limits the usefulness of the C field. Table 1 shows how Applesoft uses these fields.

Table 1

Code	Meaning
000	DON'T PLOT MOVE UP
001	DON'T PLOT MOVE RIGHT
010	DON'T PLOT MOVE DOWN
011	DON'T PLOT MOVE LEFT
100	PLOT AND MOVE UP
101	PLOT AND MOVE RIGHT
110	PLOT AND MOVE DOWN
111	PLOT AND MOVE LEFT

To obtain the codes for the C field, use only the second and third digits of the code. Because the C field is only two digits long, you can't use it to plot a point. Note also that a C value of zero will not move. Now enlarge the ball to 2 pixels by 2 pixels. Follow the steps and explanations in Table 2. A pictorial representation follows in Figure 2.

Table 2

Location In Figure 2	Explanation	Codes	Field
	BYTE 1		
1	PLOT AND MOVE RIGHT	101	A FIELD
2	PLOT AND MOVE RIGHT	101	B FIELD
3	MOVE DOWN	10	C FIELD
	BYTE 2		
4	DON'T PLOT MOVE LEFT	011	A FIELD
5	PLOT AND MOVE LEFT	111	B FIELD
	CAN'T PLOT IN C FIELD	00	C FIELD
	BYTE 3		
6	PLOT AND MOVE LEFT	111	A FIELD
	FILL REMAINING FIELDS	000	B FIELD
	WITH 0'S	00	C FIELD

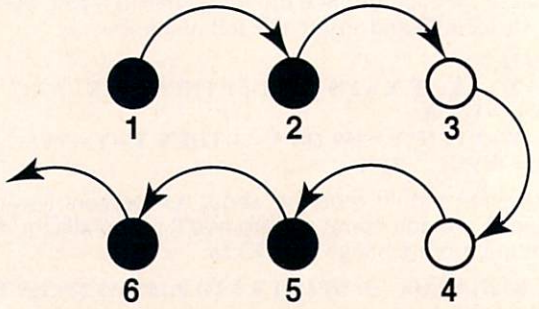


Figure 2

Applesoft Format

Now that we have the codes to produce a graphic character, we can organize this information into a format that Applesoft can understand.

Table 3

Byte Number	Field C	Field B	Field A	Hexadecimal Number	Decimal Number
1	10	101	101	AD	173
2	00	111	011	3B	59
3	00	000	111	7	7
4	00	000	000	0	0

You'll notice that we included a fourth byte in Table 3. When Applesoft finds a byte made up entirely of 0's, it knows it's at the end of the shape table. Note also the hexadecimal and decimal values for the shape table bytes. Applesoft needs the decimal values POKEd into its shape table.

Before we put these values in memory, we must supply two more pieces of information: first, the number of shapes in the table, this byte to be followed by one byte of 0's. Then we supply the number of bytes from the beginning of the table to the first shape definition. This information is in the next two bytes. The first byte is the low order byte. The second is the high order byte. Applesoft needs to find these values starting at, in our case, memory location 768. In Applesoft BASIC the simplest way to put these values in memory is to use a POKE list. Table 4 shows the complete POKE list.

Table 4

Address	Value	Comment
768	01	Number of shapes used
769	00	set to 0'S
770	04	Low byte of offset from the beginning of the table
771	00	High byte of offset from the beginning of the table
772	173	First shape byte
773	59	Second shape byte
774	7	Third shape byte
775	0	Mark end of table

Now that we understand the requirements of a shape table, let's write the code to put it in memory.

```
5 GOSUB 300
310 FOR I=0 TO 7: READ A: POKE I+768,A: NEXT:
    RETURN
320 DATA 1,0,4,0,173,59,7,0
```

Next, we need to give Applesoft the starting address of the shape table. We can POKE this into memory locations 232 and 233. Applesoft expects to find the starting address at these two consecutive locations. Our starting address, 768 decimal, is equivalent to 0300 hexidecimal. We have to put the decimal equivalent of the low byte, 00, at location 232, and the decimal equivalent of the high byte, 03, at location 233. Type the following line:

```
300 SCALE=1: ROT=0: POKE 232,0: POKE 233,3
```

Two new commands show up in this line. SCALE determines the magnification of the shape. In this case, the value 1 means it does not magnify at all. ROT stands for rotation. Here a value of 0 draws the shape as it is oriented in the shape table.

Because we are going to be drawing shapes, we need to change the HPLOT commands in line 140. Edit line 140 to read:

```
140 XDRAW 1 AT W,Z: XDRAW 1 AT X,Y
```

Let's see how this line will affect our program. The XDRAW command draws a shape—in our case shape number 1—AT the column and row coordinates specified. The XDRAW command alternates the shape's color between the starting color and its complement. This is how we erase the shape as it bounces.

To erase the first shape that appears, we'll have to XDRAW it before entering the main loop. So, we'll add line 50.

```
50 XDRAW 1 AT X,Y
```

When you RUN the program, you should see the 2 x 2-pixel ball bouncing across the screen. If you'd like to try some variations, change line 140.

```
140 XDRAW 1 AT W,Z: HPLOT W,Z: XDRAW 1 AT X,Y
```

The addition of the HPLOT command makes the ball leave a trail of dots. To leave a line behind the ball, add a TO command and a second pair of coordinates to line 140.

```
140 XDRAW 1 AT W,Z: HPLOT W,Z TO X,Y: XDRAW 1 AT
    X,Y
```

For a trail of equal thickness, simplify line 140 to read:

```
140 DRAW 1 AT X,Y
```

Change the trail to violet by adding some HCOLOR statements:

```
140 HCOLOR=2: DRAW 1 AT W,Z: HCOLOR=3:
    DRAW 1 AT X,Y
```

Finally, here's how to make a round ball:

```
100 X=X+VX: IF X>272 OR X<1 THEN X=X-VX:
    VX=-VX*R
120 Y=Y+VY: IF Y>159 OR Y<1 THEN Y=Y-VY:
    VY=-VY*R
140 XDRAW 1 AT W,Z: XDRAW 1 AT X,Y
310 FOR I=0 TO 36: READ A: POKE I+768,A: NEXT:
    RETURN
320 DATA 1, 0, 4, 0, 9, 9, 9, 17, 27, 59, 31, 19, 9, 45, 45, 17,
    59, 63, 63, 23
330 DATA 41, 45, 45, 21, 59, 63, 63, 23, 9, 45, 45, 17, 27, 59,
    31, 19, 0
```

Now you should be ready to create your own graphics and do a lot more than just follow the bouncing ball.

HCM

EL DORADO TRADING GROUP

Your Western U.S. Exclusive Texas Instruments High Technology Distributor.

With over 1 million 99/4A owners wanting software and accessories, now's the time to jump on our wagon!

YOUR PROFIT GOALS WILL "PAN OUT" WITH US!

Our support services will help your company reach its profit goals by keeping you abreast of the newest and fastest turning high technology items in the line.

"PICK" THE HOTTEST SELLING SOFTWARE

Our trained sales staff will assist you in ordering the "fastest turning" software by suggesting new packages and proven best sellers such as Parsec (PHM 3112), Early Learning Fun (PHM 3002), Dragon Mix (PHM 3117), TI Invaders (PHM 3053), Touch Typing Tutor (PHM 3064) and Munch Man (PHM 3051).

EXPRESS DELIVERY

Our extensive inventory allows your order to be processed within a 24 hour time frame.

DON'T MISS OUT ON THE RUSH!

Call us today, **TOLL FREE 800-227-8292,** in California **800-248-2224** except 415 area code call **494-6600** for a confidential "High Technology" dealers price catalog.

All dealers with an assigned customer number and a retail storefront will want to act NOW to ensure delivery during the primary 4th quarter selling season.

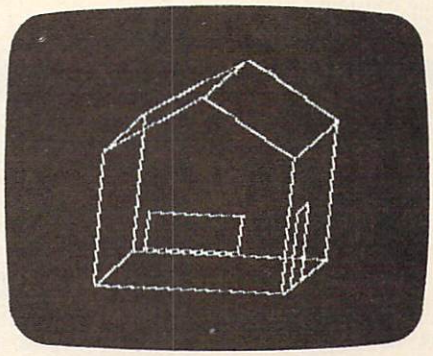
EL DORADO TRADING GROUP
760 San Antonio Rd.
Palo Alto, CA 94303

3D-IIe

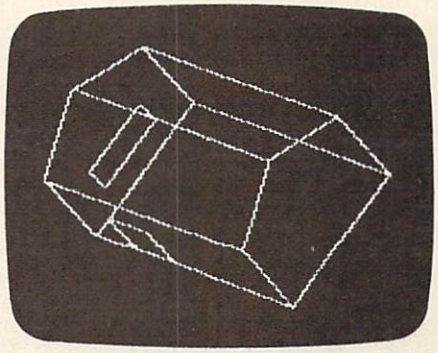
Apple Graphics in Three Dimensions

Part II

by Michael D. Brownsworth



Line drawings can be represented in three dimensions with the *Applesoft 3-D Graphics* program.



Three dimensional figures can be rotated for a view from any angle.

We hope you enjoyed *Applesoft 3-D Graphics* in the previous *Home Computer Magazine*. That program allowed you to view an object from any angle. In addition to demonstrating some of the principles of three-dimensional graphics, it turned out to be just plain fun. If you wished for an easy way to create objects of your own after you'd displayed the graphics supplied by that program, you'll be pleased with this month's companion software, an editor program that enables you to do just that.

Those of you who created graphics objects with the DATA statement method described in last month's article no doubt soon became frustrated by that tedious, error-prone process. Unfortunately, the editors supplied with many of the commercial three-dimensional graphics programs are not much easier to use. They still require you to enter *x*, *y*, and *z* coordinates for each point to be drawn, and this makes creating objects of any complexity difficult and time consuming. An editor program should be simple to use, and it should handle the more mundane details of creating three-dimensional objects—such as keeping track of point coordinates—for you.

Such obliging software does exist for professional CAD (Computer-Aided Design) systems costing considerably more than an Apple. With many of these sophisticated editors, you design the figure on-screen by moving a cursor to plot points and lines, and you can see the design as it is created.

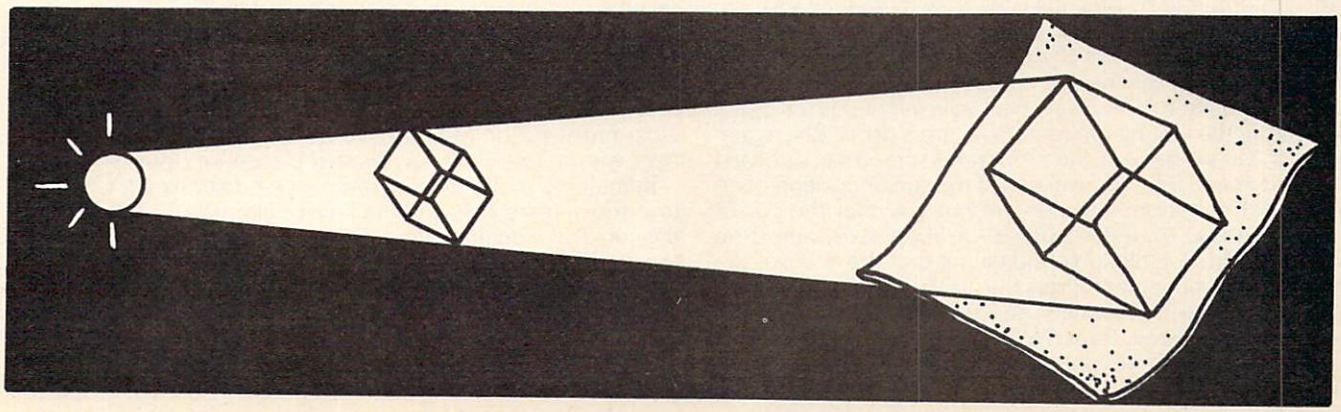
The *Editor 3-D* in Listing 5 was inspired by these CAD editors. Like them, it features on-screen object creation using a cursor that is moveable along all three axes. During the entire process you see the object from a fixed three-quarter view. The *x* and *y* axes are in standard horizontal and vertical directions, but the *z* axis is represented on the two-dimensional screen by the cursor moving obliquely, thereby simulating depth.

Entering the Listings

The Editor program is rather long and will require time and patience to type in, but you'll be amply rewarded for your efforts. The length of the program is actually a measure of its power and convenience . The Editor offers many features that make creating objects a natural, effortless process.

The listing for the Editor is heavily commented with REMs to help you understand the program flow. You may omit the REMs when you type it in, but if you elect to do this, save this *Home Computer Magazine* with the commented listing in case you wish to make your own modifications to the program at a later time.

You would be well-advised to start with a blank initialized diskette and add the elements of the system one at a time. Assuming that you already have a copy of *Applesoft 3-D* from the last issue, type in and SAVE *Exec Create* (Listing 2), the



HELLO PROGRAM — Listing 1
Explanation of the Program

Line Nos.	
100-160	Program header.
170	Define constant.
180-190	EXEC LOMEN.EXEC.

APPLE II Series

```
100 REM  **********************
110 REM  *  HELLO PROGRAM  *
120 REM  **********************
130 REM  BY M.D. BROWNSWORTH
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  APPLE II SERIES APPLESOFT
170 D$ = CHR$ (4)
180 PRINT D$;"EXEC LOMEM.EXEC"
190 END
```

HCM

Hello program (Listing 1), the *Menu* program (Listing 3), and *Editor 3-D* (Listing 5). Then add this month's new disk and error-handling routines (Listing 4) to *Applesoft* 3-D (from *HCM* Volume 4 Number 1). To perform this last typing chore, simply LOAD the original program and type in the new lines; then SAVE the entire program to disk with the name APPLESOFT 3-D. You also need to change two lines from the previous issue's program. In Line 1080, change 2000 to 2060 so the quit option will go to the correct place. In addition, because we are adding the Load from Disk option, Line 770 needs to be modified to read:

770 ON OBJ GOSUB 1190,1320,1480,1900: REM RE-INITIALIZE ARRAY

When you've finished typing in these listings, you will have created a "turn-key" menu-driven system to load and run the appropriate module; this makes it easy to use the Editor in conjunction with the 3-D display program. Here's an overview of system operation: On boot-up, the *Hello* program executes (EXECs) LOMEN.EXEC, which sets LOMEM: to protect the High Resolution (Hi-Res) screen, then runs the system *Menu* program that allows you to select either the editor program *(Editor 3-D)* or the display program (*Applesoft 3-D*). You may return to the MENU at any time from either module. You create the EXEC file LOMEM.EXEC by RUNing the short *Exec Create* program in Listing 2. After you've finished typing in and saving the programs to disk, the catalog of your 3-D System Disk should look like this:

A 002 HELLO
A 003 EXEC CREATE
T 002 LOMEM.EXEC
A 003 MENU
A 022 APPLESOFT 3-D
A 038 EDITOR 3-D

NOTE: Once you have created LOMEN.EXEC by RUNing *Exec Create*, you can delete *Exec Create* from the disk.

Getting to Know the Editor

Now let's try out the Editor and familiarize ourselves with its features. From the Editor's menu, select 1)CREATE NEW OBJECT. You will be presented with the main command screen, which is mostly blank at the top except for the cross-hair cursor at the lower left. From this screen you can always return to the Editor menu by pressing [ESC]. Try it, then choose option 1 again to return to the command screen. Now press [RETURN] to see a list of the commands available to you. (For a description of these commands, see "Editing Commands at a Glance.") You can avail yourself of this list at any time during the editing process. Now press [RETURN] again, and you will be back at the command screen (as opposed to the command list). You will notice the cursor-position coordinates at the bottom-left; they inform you that the cursor is at position 80 on the *x* axis, 40 on the *y* axis, and 0 on the *z* axis. The X is blinking, indicating that the *x* axis is active for cursor movement. Press the arrow keys—you will see the cursor move horizontally, ten units at a time, as the *x* coordinate is updated. Press S to toggle the cursor movement to only one unit per keypress if you want more precise cursor movements. Move the cursor to an even unit of tens and

EXEC CREATE — Listing 2
Explanation of the Program

Line Nos.	
100-160	Program header.
170-180	Define string constants.
190-200	OPEN text file and start write.
210	Initialize pointers to protect hi-res screen.
220-240	Run MENU, CLOSE file and END.

APPLE II Series

```
100 REM  **********************
110 REM  *  EXEC CREATE  *
120 REM  **********************
130 REM  BY M.D. BROWNSWORTH
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  APPLE II SERIES APPLESOFT
170 D$ = CHR$ (13) + CHR$ (4)
180 TF$ = "LOMEM.EXEC"
190 PRINT D$;"OPEN ";TF$
200 PRINT D$;"WRITE ";TF$
210 PRINT "POKE 103,1: POKE 104,64: POKE 16384,0: POKE 16385,0: POKE 16386,0
220 PRINT "RUN MENU"
230 PRINT D$;"CLOSE ";TF$
240 END
```

HCM

hit S to toggle back to the faster speed. Change to the *y* axis by pressing the Y key, and use the arrows to move the cursor. It now moves vertically. Press the *z* key, then the right arrow. The cursor moves in a diagonal direction to the upper-right; this is our simulated third dimension of depth mentioned earlier. Play around with the cursor on the various axes for a while, to get the feel of things. You may notice that the cursor is prevented from moving off the screen or to a coordinate less than 0 on the *z* axis; in certain circumstances, this means that the cursor will halt in the middle of the screen. All plotting with the Editor will be done with positive coordinates along the three axes.

Now let's use the Editor to create a pyramid similar to the one programmed by *Applesoft 3-D* last month. First, return the cursor to its original coordinates of 80,40,0 (*x*,*y*, *z*). Press

"When you've finished typing in these listings, you will have created a 'turn-key' menu-driven system to load and run the appropriate module; this makes it easy to use the Editor in conjunction with the 3-D display program."

P to get a startpoint. You will hear a beep and notice that [POINT] appears just above the cursor coordinates. This is the plot status indicator. With the *x* axis active, using the right arrow, move the cursor to 160. Press L. Not only is a line drawn from the startpoint to 160, but the status indicator has changed to [LINE]. Now press Z and move the cursor *back* to 80 on the *z* axis and press L to draw another line from the end of the preceding line.

If you make a mistake at any point, press E to erase the last line plotted. (It's possible to erase an entire figure in reverse of the order in which it was drawn. You may also use the faster method of pressing C to clear the entire screen and start over.) Now change to the *x* axis, move to 80 again (however, as you can see, this point on the *x* axis is 80 units *behind* the original point), and draw another line. Move the cursor *forward* on the *z* axis to the original point, and connect with a line. You now have the base for your pyramid.

Remaining on the *z* axis, return the cursor to 40. Change to *y*, move up to 110, and then move along *x* to 120 to place the cursor at a point in three-dimensional space precisely over the middle of the pyramid base. The cursor coordinates should indicate: *x*: 120, *y*: 110, *z*: 40. Draw a line from point 1, the front left corner of the base, to point 5, the present cursor position. Until now, we have been using L to draw lines, but a convenient feature of the Editor is its ability to draw lines to a common vertex marked by the cursor posi-

MENU PROGRAM Listing 3
Explanation of the Program

Line Nos.

100-160	Program header.
170	Define constant.
180-220	Display menu.
230-270	Get input and branch to appropriate program.

APPLE II Series

```
100 REM  **********
110 REM  *  MENU  PROGRAM  *
120 REM  **********
130 REM  BY M.D. BROWNSWORTH
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  APPLE II SERIES APPLESOFT
170 TEXT : HOME : D$ = CHR$ (4)
180 HTAB 10: PRINT "3-D GRAPHICS SYSTEM
    ": HTAB 10: FOR X = 1 TO 19: PRINT
    "-": NEXT
190 VTAB 6: HTAB 1: PRINT "OPTIONS:"
200 VTAB 9: HTAB 5: PRINT "1) CREATE OR
    EDIT OBJECT"
210 VTAB 11: HTAB 5: PRINT "2) DISPLAY
    COMPILED OBJECT"
220 VTAB 13: HTAB 5: PRINT "3) EXIT TO
    BASIC"
230 VTAB 16: PRINT "CHOOSE ONE: ";: POK
    E 36,0: GET K$: PRINT K$
240 IF K$ = "1" THEN PRINT D$;"RUN EDI
    TOR 3-D"
250 IF K$ = "2" THEN PRINT D$;"RUN APP
    LESOFT 3-D"
260 IF K$ = "3" THEN END
270 GOTO 230
```

HCM

tion, such as the top point of our pyramid. These lines to a common vertex are called *rays* and are summoned by the R key. To demonstrate, move the cursor along the *x* axis to 160, down the *y* axis to 40, then forward on the *z* axis to 0. Press R and notice that a line is drawn from the cursor to the starting point. Move the cursor back on the *z* axis to 80 and press R to draw another ray. It is important to understand the difference between L and R commands: An L sets the startpoint of the next line to be drawn to the cursor position (or rather, to the end of the line just drawn); an R draws lines to a common vertex at the cursor position. You will soon learn to use a combination of lines and rays to speed up the creation of objects with common vertices. To complete the pyramid, move the cursor along the *x* axis to 80, draw the final ray, and you're done.

You have to admit that this was considerably easier than manually figuring out all those *x,y,z* coordinates. If you want to plot at another point on the pyramid, or elsewhere on the screen, you must set a new point again with P. Otherwise, the next line drawn would be connected to the end of the last line of the pyramid, at either the summit (if a ray were used) or at the base if a line were used to complete the figure.

Now you're ready to SAVE the pyramid to disk. You can either SAVE to the 3-D System Disk or use a separate data disk. Press [ESC] for the menu. Choose 3)SAVE TO DISK. Next enter the filename under which you wish to save the object. Then you are asked to PRESS 'E' TO SAVE EDIT FILE ONLY, ANY OTHER KEY FOR ALL FILES. For now, we want to SAVE all the files, so press any key. When is E used? If you were in the process of editing a figure but had to stop for some reason, you would then probably SAVE only the Edit file and continue editing later. The other files are used by the APPLESOFT 3-D display program. SAVE all files after you have finished drawing the object, or any time you want to display your work.

Keep in mind that as long as the Edit file for the object exists on disk, you can use the Editor to modify or add on to the object any time, even if you had regarded it as finished. To assure yourself of this, clear the pyramid from the screen; then from the menu choose 4) LOAD OBJECT FROM DISK to reload the pyramid. Note that the cursor is positioned at the end of the last line drawn. Now go back to the menu and quit the Editor by selecting 5) RETURN TO SYSTEM MENU. Select the display program from the system menu, load in the pyramid, and rotate it. All is well; your Edit file remembers the pyramid and where you left off, and you can always modify your object.

APPLESOFT 3-D GRAPHICS (new routines) Listing 4
Explanation of the Program

Line Nos.

1640	Begin Load from disk routine.
1650	Initialize ONERR branch locations.
1660-1690	Get Input and branch.
1700-1790	Input PNTCOUNT file.
1800-1870	Input LINECOUNT file.
1880-1890	Branch to Init array and branch to main routine.
1900-1940	Init Array routine.
1950-2050	Disk Error-Handling.
2060-2090	Return to MENU program.

APPLE II Series

```
1640 REM  LOAD FROM DISK*****
1650 ONERR GOTO 1950: REM  DISK ERROR-H
     ANDLING ROUTINE
1660 PRINT : PRINT : PRINT "PRESS 'C' TO
     CATALOG, 'A' TO ABORT"
1670 PRINT : INPUT "FILENAME: ";NAME$
1680 IF NAME$ = "C" THEN PRINT D$;"CATA
     LOG": GOTO 1650
1690 IF NAME$ = "A" THEN RETURN
1700 FLNAME$ = NAME$ + ".XYZ"
1710 PRINT D$;"LOAD ";FLNAME$: REM  FOR
     CES ERROR TO CONFIRM FILE PRESENT O
     N DISK BEFORE OPENING
1720 PRINT D$;"OPEN ";FLNAME$
1730 PRINT D$;"READ ";FLNAME$
1740 INPUT PNTCOUNT
1750 FOR J = 1 TO PNTCOUNT
1760 INPUT X%(J): INPUT Y%(J): INPUT Z%(
     J)
1770 Y%(J) = ABS (Y%(J) - 160)
1780 NEXT J
1790 PRINT D$;"CLOSE ";FLNAME$
1800 FLNAME$ = NAME$ + ".PC"
1810 PRINT D$;"OPEN ";FLNAME$
1820 PRINT D$;"READ ";FLNAME$
1830 INPUT LINECOUNT
1840 FOR J = 1 TO LINECOUNT * 2
1850 INPUT PC(J)
1860 NEXT J
1870 PRINT D$;"CLOSE ";FLNAME$
1880 GOSUB 1900: REM  INITIALIZE ARRAY
1890 GOTO 1090: REM  MUST BE GOTO; RETUR
     N IS EATEN BY ONERR
1900 REM  INITIALIZE ARRAY*****
1910 FOR J = 1 TO PNTCOUNT
1920 X(J) = X%(J): Y(J) = Y%(J): Z(J) = Z%
     (J)
1930 NEXT J
1940 RETURN
1950 REM  DISK ERROR-HANDLING*****
1960 ERR = PEEK (222): POKE 216,0: REM
     DETERMINE ERROR #, RESET ERROR FLA
     G
1970 PRINT
1980 IF ERR = 13 THEN GOTO 1720: REM
     FILE EXISTS
1990 IF ERR = 6 THEN PRINT NAME$;" NOT
     FOUND ON DISK"
2000 IF ERR = 8 THEN PRINT "NO DISK IN
     DRIVE OR DRIVE DOOR OPEN"
2010 IF ERR = 9 THEN PRINT "DISK FULL"
2020 PRINT : PRINT "<RETURN> TO TRY AGAI
     N>, <ESC> TO ABORT";: GET K$: PRIN
     T
2030 IF K$ = CHR$ (27) THEN GOTO 960
2040 IF K$ = CHR$ (13) THEN GOTO 1640
2050 GOTO 2020: REM  MUST BE GOTO; BUG I
     N APPLESOFT'S RESUME COMMAND
2060 REM  RETURN TO MENU*****
2070 TEXT : HOME
2080 PRINT D$;"RUN MENU"
2090 END
```

HCM

The following is some additional information about the Editor that you should keep in mind:

1. The Editor is designed to plot lines or rays only; it is not possible to plot isolated points.

2. If a point has been set, you must follow through with a line or a ray.

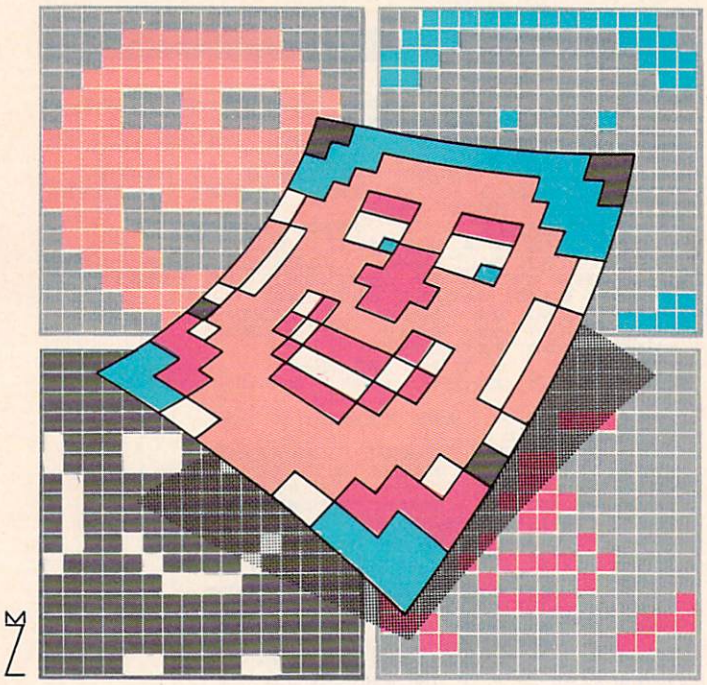
3. Line erase is inactive if only a point has been set; if you must erase, first draw a line, then erase.

4. After loading an object to edit, you must first set a point before drawing.

This powerful, easy-to-use tool provides you with the ability to create some interesting, complex figures. Try creating a house with furniture, for example, or perhaps an airplane. Unleash you imagination, and have fun.

Continued on p. 157

DOUBLE YOUR COLOR
DOUBLE YOUR FUN



A Sprite Tutorial

by W. K. Balthrop
HCM Staff

If you've ever tried to design your own graphics, you may have been frustrated by the limited number of colors you can display in a small area of the screen. This tutorial will show you how to get more color into your screen displays by overlapping sprites and screen graphics. And our *Sprite Color Graphics* program lets you create a colorful face by layering sprites and using sprite priorities to create animation.

Sprites are graphics shapes that move with the highest resolution of movement possible so that they appear to glide across the screen. The video processor will allow only a set number of sprites to be placed on top of each other without disappearing. (The TI-99/4A allows four, and the Commodore 64 allows eight; the TI video processor, however, will allow up to 32 sprites to appear on screen if the four-in-a-row constraint isn't exceeded.) If more than the allowable number of sprites occupy the same horizontal line on the screen, the sprite with the lowest priority will disappear, until only the allowable number of sprites with the highest priorities remain. The lower the sprite number, the higher that sprite's priority. Sprite #1 has the highest priority on the 99/4A, and sprite #0 the highest priority on the Commodore 64. These disappearing sprites don't create much of a problem and can even be used to your advantage, as you will see in the program below.

At the heart of the TI-99/4A's graphics prowess is the Texas Instruments TMS9918A video processor. One of the most powerful features of the 9918A video chip is its sprite capabilities. Unfortunately, the TI-99/4A computer allows you to use only two colors within a single character position (unless you use assembly language). That restriction, however, does not apply to sprites, which brings us to the point of this tutorial: It is possible to get a wide range of colors in a small area by overlapping sprites and screen graphics. The TI version of *Sprite Color Graphics* will draw a face on the screen using five different colors.

A maximum of six colors can be placed in a single character simply by laying sprites on top of each other. We used only one background color in this program, so our face has only five colors. Had we put a second background color behind the sprites, we would have had a sixth color.

Line 180 of the TI program clears the screen with the CALL CLEAR command and sets it to dark yellow with the CALL SCREEN command. There are no sprites on the screen yet, though it should be noted that CALL CLEAR does not erase existing sprites: Only the *characters* on the screen are erased. CALL SCREEN sets the screen color. The default background color for all of the characters is color #1, which is clear, so that when anything is printed or displayed, only the foreground color will show up.

You can change the foreground and background colors for each of the characters. When the screen is blank, it is actually full of the blank character (#32). If you gave the character set containing character #32 (set #1) a background of dark yellow, then the screen would become dark yellow, except for the borders. The borders would retain their original color. This color is changed by the CALL SCREEN command.

Lines 190 and 250 define the sprites' shapes and assign those shapes to characters. In Extended BASIC it is possible to assign a shape to more than one character with only one statement. Each of the CALL CHAR statements here assigns a shape to four characters. Line 190 assigns a shape to characters 96, 97, 98, and 99. The characters within quotes contain the graphics shape information in a *hexadecimal* representation. This is a numbering system that the computer uses, which counts to 16 in the same way as we count to 10. Think of the computer as having 16 fingers to count on instead of ten. The letters A through F represent those new numbers between 10 and 15. F represents 15, not 16, because the counting starts at 0, not 1. Counting from 0 to 15 takes 16 numbers.

Shape Definition

It takes 16 of the characters in the quotes to make the shape for one graphics character. There are 64 characters in each of the statements that makes four characters. The only exception to this is in lines 240 and 250, which have only 62 characters in each statement. This is because the last two characters would have been 00. Whenever there are trailing zeros in the CALL CHAR statement, you can leave them

off. However, you need at least one zero to assign a shape to the character. For example, if line 190 had only 48 characters in the definition, then the fourth graphics character would not be defined. If a 49th character were added, then the fourth character would be defined with its upper left corner defined to the shape prescribed by the 49th character in the statement.

Line 260 sets character set #12 to a black foreground and background. Character set #12 includes characters 120 to 127. All of these characters will now be displayed as a solid black block. Character 120 is then placed in the center of the screen in line 270. The CALL HCHAR statement has a fourth, optional parameter, which lets you repeat the character any number of times you wish. Here we repeat the character four times and use the CALL HCHAR statement four times to produce a 4 x 4 black block. The sprites we create later will be placed on top of this block.

Line 280 sets the magnification of the sprites to 4. At this magnification each sprite will be made up of four characters and will also be double its normal size. This makes the sprite four characters high and four characters wide. Try experimenting with different values (from 1 to 4) in this statement to see the effects of the CALL MAGNIFY command.

Line 290 places the four highest-priority sprites at the edge of the screen. With these four sprites at the edge, any sprites with lower priorities placed on the same horizontal line will be invisible. Remember, the 99/4A computer displays only the four sprites with the highest priority. The next four sprites we place on the screen (in lines 300-330) will not be visible yet because the four sprites at the edge have lower sprite numbers, which gives them higher priorities. Line 340 will wait for you to press [ENTER] and then set one of the four sprites at the edge in motion upwards. As this sprite moves up, in the center of the screen one of the sprites that makes up the face gains priority and starts becoming visible. If you press [ENTER] again, a second sprite at the edge will start moving up, giving priority to a second sprite in the face. This continues until all four sprites at the edge of the screen have moved away. To make sure the last sprite moves all the way to the top of the face, the time delay loop in line 350 will wait and then delete all four of the sprites that were used at the edge.

The last part of this short program adds a bit of animation to the face by changing the pattern of the sprites. You may have noticed that in lines 190 to 250 we defined enough shapes for seven sprites, but we used only four in the face. The other three shapes will be assigned to the sprites that make the face appear to change. Lines 370, 380, 400, and 410 will cause the eyes to roll back and forth. Lines 390 and 420 will make the mouth open and close. Notice that when the mouth opens, it's black inside. Remember the black characters we put down under the sprites? That black was there all along.

To lend a little more excitement to the face, line 430 contains a variable time delay loop. After each shape has been updated in lines 370 to 420, the variable for the time delay loop is set. A GOSUB takes the program to the subroutine temporarily, to help slow things down. You may want to experiment with the delay times by changing the value assigned to the variable D. The larger the number, the slower the response. You might also want to alter the sequence of changes, or even make them random. There are all sorts of exciting things you could try. Have fun, and happy computing!

Continued on p. 45

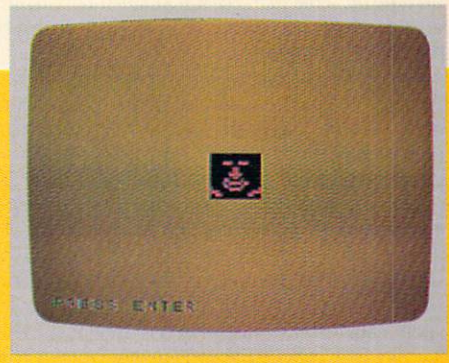


Photo 1: The black square is placed on the screen to help make the face construction more visible. Here you see the first sprite, colored red, placed over the box. At this point all four sprites are on the screen. They are kept invisible by the four higher priority sprites at the edge of the screen. Press [ENTER] to see the second sprite appear.

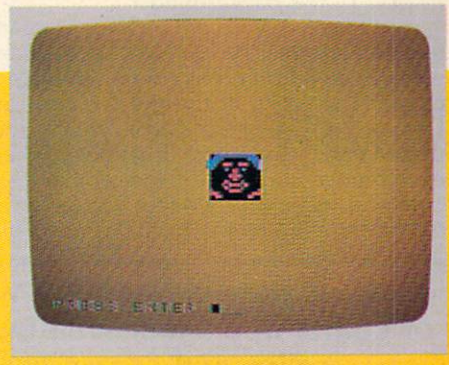


Photo 2: The second sprite is blue and is used to create the hair, the center of the eye, and the collar. Later in the program this sprite's shape will be changed so that the eyes appear to move from left to right and back again. Press [ENTER] to have the next sprite displayed.

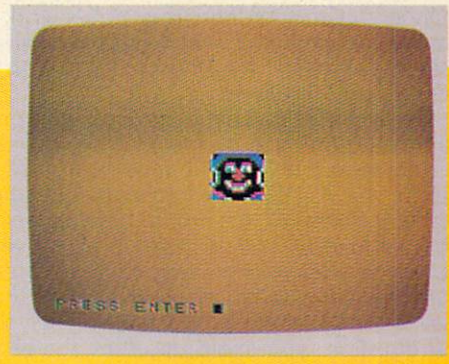


Photo 3: The third sprite is white and is used to finish the eyes, fill in the mouth, and make the ears. Press [ENTER] to see the last sprite appear on the face.

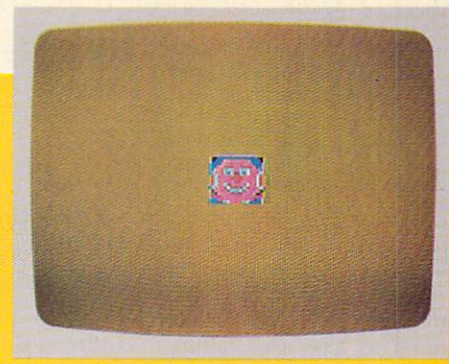


Photo 4: The last sprite to make up the face is light red. This sprite is used to fill in the rest of the face. You will not need to press [ENTER] anymore. From this point the program goes into an animation sequence which moves the eyes and mouth.

The Commodore 64 home computer offers BASIC programmers a number of methods for creating spectacular graphics. You can use the resident character set, or you can define graphics characters of your own. In the normal graphics mode, you can assign any one of the sixteen colors to a character, but you are limited to one color for the background. In the multicolor mode, you can choose any one of four colors to assign to a pair of pixels (a pixel is the

Continued on p. 44

smallest dot on the screen). This mode has limitations, however. The horizontal screen resolution is cut in half, and you are limited to four colors, including the background, for the whole screen.

Sprite Color Graphics allows you to have the best of both worlds. With the C-64 version, you can use up to ten colors in a limited area of the screen without sacrificing screen resolution. Every pixel can be defined with one of the ten colors and with a resolution of 320 x 200.

This method involves the use of sprites. The C-64 offers eight sprites, which can be programmed with eight different colors and can all be placed on top of each other. Our strategy for creating a shape with ten different colors utilizes both of those capabilites: You turn on selected bits in each sprite and place them on top of each other, then you display two colors behind them, using both a regular character and the background. Animation is easy because of the flexibility of the sprites. You can move the sprites with high resolution, alter their shapes and colors, and change their priority with respect to the background.

I used the *Sprite Color Graphics* program with seven sprites to create a face. The hair, skin, nose, ears, mouth, and eyes were assigned different colors. The eye is made up of two colors. The white of the eye is white, and the iris is cyan. I made the background solid black, though I could have placed standard graphics characters in the background to give even more color. The C-64 version of the program uses eight colors to make up the face: The seven sprites all have different colors, plus the background is black.

"It is possible to get a wide range of colors in a small area by overlapping sprites and screen graphics"

For the purpose of this tutorial, the sprites were magnified so that the different colors in the face would be easier to see. Once you've seen how the sprites work in their magnified mode, delete line 230 and run the program again. You will see the sprites at their normal size, with all of the different colors still displayed.

To add a little variety and make the program more fun, I animated the face by moving the eyes back and forth and opening and closing the mouth. Instead of creating a new sprite shape, I decided to change only the part of the shape table in memory that would be affected. To move the eyes back and forth, I could have simply moved the whole sprite. Unfortunately, this method causes problems if you want to use the same sprite for other parts of the shape, but don't want to move them all. By changing only the area of the sprite in question, you avoid upsetting the rest of the picture.

To find the memory locations that need to be altered, you first need to know which DATA statements contain the sprite graphics patterns. The following table shows which line numbers define each sprite, and what part of the face each sprite is used for.

440-470	**Sprite #0 hair**
480-510	**Sprite #1 mouth**
520-550	**Sprite #2 center of the eye**
560-590	**Sprite #3 eyeball**
600-630	**Sprite #4 nose**
640-670	**Sprite #5 main face**
680-710	**Sprite #6 ears**

In line 190 we see that the shapes start at memory location 15360. If you want to change the shape of the eye in line 530, simply count the number of items in the DATA statements up to the byte you want to change and add that number to 15360. To change the eye in this program, we need to alter the values in addresses 15509, 15510, and 15511. By changing the values in these three addresses, we can make the eyes move. The mouth is a little more difficult, because it requires alteration of more memory. Here we must alter one byte of data from line 500 and seven bytes of data from line 510. The addresses are calculated as explained above. Lines 350-430 make all of this happen. This section of the program is broken up into several subroutines that each perform one of the movements.

Lines 260 through 330 branch to the subroutines in the proper sequence. The mouth is opened when the eyes reach the left side and closed when they reach the right side. The time delay loops in lines 260 to 330 slow the program down so that you can see the movement. Try changing some of the values in the loops to see what effect they have on the action. You may even want to try branching to the subroutines in a different sequence. This can be done by changing the line numbers after the GOSUB in lines 260 through 330. Valid line numbers to use are 350, 360, 370, 400, and 410.

***SPRITE COLOR GRAPHICS* (C-64)**
Explanation of the Program

Line Nos.	
100-170	Program header.
180	Clear screen; set screen color to black.
190	Load sprite shapes into memory.
200	Load sprite colors into memory.
210	Load sprite shape pointers into memory.
220	Move all sprites off screen.
230	Expand sprites.
240	Enable sprites.
250	Place sprites on screen with a short time delay after each one.
260-340	Branch to subroutines to move eyes and mouth.
350-430	Subroutines to move eyes and mouth.
440-710	Data for the sprite shapes.
720	Data for the sprite colors.

COMMODORE 64

```
100 REM  ********************
110 REM  * SPRITE COLOR GRAPHICS *
120 REM  ********************
130 REM  BY WILLIAM K. BALTHROPE
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  C-64
170 REM
180 PRINT "■[SHIFT CLR]": POKE 53281,0
190 POKE 52,56:POKE 56,56:FOR X=15360 T
    O 15807:READ Y:POKE X,Y:NEXT X
200 FOR X=53287 TO 53293:READ CO:POKE X
    ,CO:NEXT X
210 Y=239:FOR X=2040 TO 2046:Y=Y+1:POKE
    X,Y:NEXT X
220 FOR X=53248 TO 53261:POKE X,0:NEXT
    X
230 POKE 53271,255:POKE 53277,255
240 POKE 53269,255
250 FOR X=53248 TO 53261:POKE X,150:FOR
    TD=1 TO 100:NEXT TD:NEXT X
260 FOR X=1 TO 50:NEXT X:GOSUB 350
270 FOR X=1 TO 50:NEXT X:GOSUB 360
280 FOR X=1 TO 50:NEXT X:GOSUB 370
290 FOR X=1 TO 50:NEXT X:GOSUB 360
300 FOR X=1 TO 50:NEXT X:GOSUB 350
310 FOR X=1 TO 50:NEXT X:GOSUB 400
320 FOR X=1 TO 50:NEXT X:GOSUB 410
330 FOR X=1 TO 50:NEXT X:GOSUB 400
340 GOTO 260
350 POKE 15509,1:POKE 15510,129:POKE 15
    511,128:RETURN
360 POKE 15509,3:POKE 15510,3:POKE 1551
    1,0:RETURN
370 POKE 15509,6:POKE 15510,6:POKE 1551
    1,0:POKE 15470,255:POKE 15472,2
380 POKE 15473,0:POKE 15474,64:POKE 154
    75,3:POKE 15476,129:POKE 15477,192
390 POKE 15479,255:RETURN
400 POKE 15509,0:POKE 15510,192:POKE 15
    511,192:RETURN
410 POKE 15509,0:POKE 15510,96:POKE 155
    11,96:POKE 15470,129:POKE 15472,1
420 POKE 15473,255:POKE 15474,128:POKE
    15475,0:POKE 15476,126:POKE 15477,0
430 POKE 15479,0:RETURN
440 DATA 3,255,192,15,255,240,63,255,25
    2,124,0,62,112,0,14,224
450 DATA 0,7,192,0,3,128,0,1,0,0,0,0,0,
    0,0,0
460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0
470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0
480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0
```

COMMODORE 64

```
490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,129,192
510 DATA 1,255,128,0,126,0,0,0,0,0,0,0,0,0,0,0
520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
530 DATA 0,0,0,0,0,0,1,129,128,0,0,0,0,0,0,0
540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
570 DATA 0,0,3,195,192,7,231,224,3,195,192,0,0,0,0,0
580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

COMMODORE 64

```
610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,0,0
620 DATA 0,0,60,0,0,60,0,0,126,0,0,0,0,0,0,0,0,0
630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
640 DATA 0,0,0,0,0,0,0,0,0,0,3,255,192,15,255,240,31
650 DATA 255,248,60,60,60,56,24,28,60,60,63,231,252,63,231
660 DATA 252,63,195,252,63,195,252,63,129,252,31,255,248,28,126,56
670 DATA 14,0,112,15,129,240,7,255,224,3,255,192,0,255,0,0
680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
690 DATA 0,0,0,0,0,0,64,0,2,192,0,3,192,0,3,192,0
700 DATA 3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
720 DATA 9,2,3,1,7,10,13
```

HCM

Double Your Color . . . from p. 43

SPRITE COLOR GRAPHICS (TI)

Explanation of Program

Line Nos.	
100-170	Program header.
180	Clear the screen.
190-250	Define sprite shapes.
260-270	Make the characters for the box black and place the black box in the center of the screen.
280	Set sprite magnification to 4.
290	Place the four blocking sprites on the screen.
300-330	Display the four sprites which make up the face.
340	Scroll the blocking sprites off the screen.
350-360	Delete the blocking sprites when the scrolling is finished.
370-420	Recursive routine to move the eyes and the mouth.
430	Time delay loop.

TI-99/4A

```
100 REM  *********************
110 REM  *  SPRITE COLOR GRAPHICS  *
120 REM  *********************
130 REM  BY WILLIAM K. BALTHROP
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  TI EXTENDED BASIC
170 REM
180 CALL CLEAR :: CALL SCREEN(11)
190 CALL CHAR(96,"000F1F3F7171FFFFFFFF3717
    83C1F0F0300F0F8FC8E8EFFFFFFCCF8E1E
    3CF8F0C0")
200 CALL CHAR(100,"3F70E0C08002000000000000
    0000000C0F0FC0E0703011000000000000000
    000000030F")
210 CALL CHAR(108,"3F70E0C08004000000000000
    0000000C0F0FC0E0703012000000000000000
    000000030F")
220 CALL CHAR(112,"000000000E8E8E4040404048
    8403000000000C0000000007171020202121C
    000000030")
```

TI-99/4A

```
230 CALL CHAR(116,"3F70E0C08008000000000000
    0000000C0F0FC0E0703014000000000000000
    000000030F")
240 CALL CHAR(124,"000000000E0001010301040
    0B04830E03000000000007000808080C08020D02
    0C1070C")
250 CALL CHAR(132,"000000000E0001010301040
    0B04808E33000000000007000808080C08020D02
    001C70C")
260 CALL COLOR(12,2,2)
270 CALL HCHAR(11,15,120,4)::CALL HCHA
    R(12,15,120,4)::CALL HCHAR(13,15,1
    20,4)::CALL HCHAR(14,15,120,4)
280 CALL MAGNIFY(4)
290 CALL SPRITE(#1,32,1,81,255,#2,32,1,
    81,255,#3,32,1,81,255,#4,32,1,81,25
    5)
300 CALL SPRITE(#5,124,7,81,113)
310 CALL SPRITE(#6,108,6,81,113)
320 CALL SPRITE(#7,112,16,81,113)
330 CALL SPRITE(#8,96,10,81,113)
340 FOR X=1 TO 4::DISPLAY AT(24,1)BEE
    P:"PRESS ENTER":::ACCEPT AT(24,13)
    :AS::CALL MOTION(#X,-2,0)::NEXT
    X
350 FOR X=1 TO 1000::NEXT X::CALL D
    ELSPRITE(#1,#2,#3,#4)
360 DISPLAY AT(24,1):"                ";
370 CALL PATTERN(#6,100)::D=70::GOSU
    B 430
380 CALL PATTERN(#6,108)::D=20::GOSU
    B 430
390 CALL PATTERN(#5,132)::D=10::GOSU
    B 430
400 CALL PATTERN(#6,116)::D=70::GOSU
    B 430
410 CALL PATTERN(#6,108)::D=20::GOSU
    B 430
420 CALL PATTERN(#5,124)::D=10::GOSU
    B 430::GOTO 370
430 FOR TD=1 TO D::NEXT TD::RETURN
```

HCM

.....Texas Instruments TI-99/4A — COMPUTERS, COMPONENTS AND SOFTWARE.....

TEX-COMP * P.O. BOX 33084 * GRANADA HILLS, CA 91344* (818)366-6631

TEX-COMP™

TI USERS SUPPLY COMPANY "YOUR LUBBOCK CONNECTION™"

America's Number One TI computer retailer

TI-99/4 PRODUCTS AT PROFESSIONAL PRICES

ATTENTION TI-99/4 USERS TEX COMP HAS BEEN SUPPORTING THE ENTIRE 99/4 PRODUCT LINE TO PROFESSIONAL AND INDUSTRIAL ACCOUNTS. SINCE THE INTRODUCTION OF THE TI-99/4 NOW YOU CAN PURCHASE ALL YOUR TI-99/4 REQUIREMENTS FROM ONE SOURCE AT LOW "PROFESSIONALLY DISCOUNTED" PRICES ON THE **ENTIRE** 99/4 PRODUCT LINE. WE ARE A DIRECT TI NATIONAL ACCOUNT AND DEAL DIRECTLY WITH TI ASSURING YOU THE BEST SERVICE, EARLIEST SHIPMENT AND LOWEST POSSIBLE PRICES

AUTHORIZED DEALER

EDITOR ASSEMBLER PHM3055

TEX-COMP HAS PURCHASED THE ENTIRE REMAINING INVENTORY OF TI EDITOR /ASSEMBLER (PHM3055) FROM TEXAS INSTRUMENTS. TI IS NOW SOLD OUT OF THIS PROGRAM AND WILL NOT BE PRODUCING ANY MORE. THIS IS THE COMPLETE EDITION INCLUDING THE MODULE, MANUAL, TWO DISKS AND A VINYL BINDER.

$38.95

A DISK DRIVE AND 32K RAM CARD ARE REQUIRED. THIS PROGRAM WILL ALLOW YOU TO PROGRAM YOUR TI HOME COMPUTER IN TMS 9900 ASSEMBLY LANGUAGE AND GIVES YOU ACCESS TO ALL SYSTEM FEATURES AND PROVIDES THE FASTEST SPEED POSSIBLE FROM THE 16 BIT PROCESSOR!

SEND $2.00 FOR THE TEX-COMP CATALOG OF TI AND THIRD PARTY PRODUCTS

NOW -- make a complete backup disk in 3 passes or less!! with QUICK-COPYer and TI EDITOR ASSEMBLER.....

This new disk program from Quality Software enables you to copy any single-sided disk in three passes or less. It is ten times faster than the TI Disk Manager Module and saves wear and tear on your disk drive.
Requires 32k and TI Editor Assembler Module(PHM3055)
$32.95 postpaid (US)
SPECIAL INTRODUCTORY PRICE FROM TEX-COMP

TI-FORTH VERSION 4.5

PHD 5098

TI-FORTH AVAILABLE
REQUIRES EDITOR/ASSEMBLER
By special arrangement with Texas Instruments, TEX-COMP TI USERS SUPPLY COMPANY is now publishing the unreleased TI FORTH Disk Software Program and the sprial bound documentation.

This long waited program is now in stock for immediate delivery from TEX-COMP for $38.95.

The Last Word on the TI-99/4A $10.95

by Linda Schreiber

this is an indispensable guide for every TI user. Schreiber explores the myriad programming possibilities of the TI-99/4A: arcade and family games . . . educational programs . . . financial, medical, and hobby applications . . . message center and security uses . . . and more! There's valuable information on the computer's architecture and the fundamentals of building a program, plus explanations of commands, functions, screen displays, and program and data storage.

HIGHLIGHTS: Commands, Statements and Functions—Making Decisions—Arithmetic Functions—Finding And Trapping Errors—Special Functions—Advanced Programming Skills—Sprites—Using The Disk—Putting It All Together.

224 pp., 50 illus.

POSTPAID (US)

ALL ADVERTISED ITEMS CAN BE PURCHASED AT OUR WAREHOUSE OUTLET. CALL FOR HOURS!

Complete Your 99/4A Computer System

New

Fundamentals of TI-99/4A Assembly Language

by M. S. Morley

—a comprehensive, easy-to-follow guide to programming in assembly language for faster program execution and more efficient micro use!

There are full details on 16-bit data transfer, code conversion and sophisticated applications like the manipulation of tables and lists. Detailed data on the 9900 instruction set and data sheet, TI-99/4A technical information, and the mini memory module is provided.

HIGHLIGHTS: Part I: Programming Languages—Microprocessor Operations—Addressing Modes—Part II: The 9900 And The TI-99/4A Home Computer—The 9900 Microprocessor—The TI-99/4A Home Computer—Part III: Simple Program Loops—Character-Coded Data—Code Conversion—Arithmetic Problems—Tables And Lists—Subroutines—Input/Output.

$10.95 322 pp., 70 illus.

POSTPAID (US)

SEND ORDER AND MAKE CHECKS PAYABLE TO:

TEX-COMP™ P.O.BOX 33084 — GRANADA HILLS, CA. 91344

[818] 366-6631

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100 and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads.*

Credit Cards add 3%

MasterCard VISA

TEX-COMP IS THE LARGEST EXCLUSIVELY TI HOME COMPUTER DEALER IN THE WORLD. WHAT WE ADVERTISE, WE STOCK. TEX-COMP MAINTAINS A $100,000.00 TI INVENTORY AT ALL TIMES AND IS A DIRECT TI DEALER.
WHEN IN SOUTHERN CALIFORNIA VISIT OUR MODERN WAREHOUSE OUTLET STORE WHERE YOU CAN PURCHASE THE ENTIRE TI LINE AT OUR REGULAR DISCOUNT PRICES.

Quick Pixel . . . from p. 24

COMMODORE 64

```
540 IF KY=14 THEN Y1=-1:RETURN
550 IF KY=23 THEN Y1=1:RETURN
560 IF KY=60 THEN FB=1
570 RETURN
580 IF X1=0 AND Y1=0 AND FB=0 THEN RETU
    RN
590 OP=1078+(CR(1,1)*40) + CR(1,2)
600 IF X1=0 THEN 660
610 IF X1=-1 THEN 640
620 IF CR(1,2)=24 THEN 660
630 CR(1,2)=CR(1,2)+X1:GOTO 660
640 IF CR(1,2)=1 THEN 660
650 CR(1,2)=CR(1,2)+X1
660 IF Y1=0 THEN 720
670 IF Y1=-1 THEN 700
680 IF CR(1,1)=21 THEN 720
690 CR(1,1)=CR(1,1)+Y1:GOTO 720
700 IF CR(1,1)=1 THEN 720
710 CR(1,1)=CR(1,1)+Y1
720 IF FB=0 THEN RETURN
730 NP=1078+(CR(1,1)*40)+CR(1,2)
740 GC=PEEK(NP)
750 IF GC=65 THEN GC=64:CK=14:GOTO 770
760 GC=65:CK=CL
770 POKE NP,GC:POKE S+NP,CK
780 CT=0:XY=0
790 Z=((CR(1,1)-1)*3)+INT((CR(1,2)-1)/8
    +1)
800 P1=INT((Z-1)/3):P2=(Z-1)-(P1*3)
810 FOR X=7 TO 0 STEP -1
820 Y=PEEK(1119+(P1*40)+(P2*8)+X)
830 Y=Y-64
840 XY=XY+((2↑CT)*Y)
850 CT=CT+1:NEXT
860 POKE 15359+Z,XY
870 RETURN
880 PRINT "■CTRL BLK■■HOME■■CRSRDOWN■"
890 FOR X=15360 TO 15360+62 STEP 3
900 N1=PEEK(X):N2=PEEK(X+1):N3=PEEK(X+2
    )
910 GOSUB 960:NEXT
920 X=0
930 PRINT TAB(3)"
940 PRINT TAB(3)"■SHIFT CRSRUP■";X
950 GOTO 330
960 PRINT TAB(3)"
970 PRINT TAB(3)"■SHIFT CRSRUP■";N1;"■S
    HIFT CRSRLEFT■";N2;"■SHIFT CRSRLEFT
    ■";N3
980 RETURN
990 OPEN4,4:B=15359:CT=1:SS=1078
1000 FOR A=1 TO 8
1010 PRINT#4,"
1020 FOR I=1 TO 8
1030 X=PEEK(B+I)
1040 PRINT#4,X;" ";
1050 NEXT I
1060 B=B+I-1:NEXT
1070 PRINT#4,"
1080 REM
1090 FOR I=1 TO 21
1100 FOR A=1 TO 24
1110 X=PEEK(SS+(40*CT)+A)
1120 IF X=64 THEN P$=CHR$(119):GOTO 1140
1130 P$=CHR$(113)
1140 PRINT#4,P$;
1150 NEXT A
1160 CT=CT+1:PRINT#4,"  ":NEXT I
1170 PRINT#4,;:CLOSE4
1180 GOTO 330
1190 POKE 53269,0:PRINT"■SHIFT CLR■■CTRL
     BLK■":GOSUB 2620
     IF SD=DS THEN END
     IF D$="N" THEN END
1200 GOTO 1800
1210 FOR I=0TO65:POKE I+15360,0:NEXT
1220 RETURN
1230 GET A$:IF A$<>"" THEN 1250
1240 INPUT"■CTRL BLK■■HOME■INPUT COLOR N
     UMBER";CL
1250 PRINT"■HOME■
     "
1260 POKE 53287,CL
1270 GOTO 330
1280 PRINT"■CTRL BLK■■SHIFT CLR■■CRSRDOW
     N■"
1290 FOR I=1 TO 21:GOSUB 1340:NEXT
1300 CR(1,1)=1:CR(1,2)=1
1310 RETURN
1320 PRINT TAB(15)"■CMDR BLU■■24SHIFT -■
     ":RETURN
1330 AD=1078+(CR(1,1)*40)+CR(1,2)
1340 CS=PEEK(AD):CC=PEEK(AD+S):POKE AD,8
     6:POKE S+AD,0
1350 FOR X=1 TO 200:NEXT
1360 POKE AD,CS:POKE S+AD,CC
1370 RETURN
1380 PRINT"■SHIFT CLR■■CTRL BLK■■DO YOU W
     ANT TO CREATE DATA STATEMENTS"
1390 PRINT "FOR CHARACTER OR SPRITE DEFI
     NITIONS?"
1400 INPUT "ENTER Y OR N";D$
1410 IF D$="N" THEN RETURN
1420 INPUT "ENTER STARTING LINE NUMBER";
     SL
1430 INPUT "ENTER INCREMENT VALUE";IV
1440 SL=SL-IV
```

COMMODORE 64

```
1470 PRINT "DO YOU WANT TO MERGE DATA ST
     ATEMENTS"
1480 PRINT "WITH AN EXISTING PROGRAM?"
1490 INPUT "ENTER Y OR N";G$
1500 IF G$="Y" THEN GOSUB 1840:GOTO 1520
1510 DS=16385:SD=DS
1520 RETURN
1530 SA(1)=15360:EA(1)=15423
1540 AC=2
1550 DP=131:CM=44:N1=AC-1:AC=1
1560 DP=SD+2:IF N1=0 THEN 1790
1570 SL=SL+IV:PRINT"■HOME■";SL
1580 LH=INT(SL/256):LL=SL-(256*LH)
1590 POKE DP,LL:POKE DP+1,LH:POKE DP+2,D
     P:POKE DP+3,32:DP=DP+4
1600 FOR X=1 TO 16
1610 SN=PEEK(SA(AC)):CT=2:IF SN=0 THEN C
     A(CT)=10:GOTO1640
1620 CA(CT)=INT((SN+.0001)/10↑CT):SN=SN-
     (CA(CT)*10↑CT):CT=CT-1:IF CT>-1 THE
     N1620
1630 CT=2
1640 IF CA(CT)>9 THEN1670
1650 IF CA(CT)=0 THEN CT=CT-1:GOTO1650
1660 CA(CT)=CA(CT)+48:POKE DP,CA(CT):DP=
     DP+1:CT=CT-1:GOTO1680
1670 POKE DP,48:DP=DP+1:GOTO1690
1680 IF CT>-1 THEN1660
1690 SA(AC)=SA(AC)+1:IF SA(AC)>EA(AC) TH
     EN X=16:GOTO1710
1700 IF X<16 THEN POKE DP,CM:DP=DP+1
1710 NEXT
1720 POKE DP,0:DP=DP+1
1730 SH=INT(DP/256):LS=DP-(SH*256)
1740 POKE SD,LS:POKE SD+1,SH
1750 SD=DP
1760 IF SA(AC)<EA(AC) THEN1560
1770 IF SA(AC)=EA(AC) THEN1560
1780 AC=AC+1:N1=N1-1:GOTO1560
1790 DP=DP-2:POKE DP,0:POKE DP+1,0:SD=DP
     :GOTO 330
1800 DP=DP+2:PH=INT(DP/256):PL=DP-(PH*25
     6)
1810 POKE 16000,PL:POKE 16001,PH
1820 POKE 45,PEEK(16000):POKE 46,PEEK(16
     001):POKE 44,64
1830 END
1840 PRINT:PRINT:PRINT "WORKING....."
1850 AS=16385
1860 X=PEEK(AS):IF X<>0 THEN AS=AS+1:CT=
     0:GOTO 1860
1870 CT=CT+1:IF CT<>3 THEN AS=AS+1:GOTO
     1860
1880 DS=AS-1:SD=DS:RETURN
1890 GOSUB 2620:PRINT"■SHIFT CLR■■2CRSRD
     OWN■ENTER 1 FOR SPRITE EDIT"
     PRINT" ENTER 2 FOR CHARACTER EDIT"
1900 INPUT" ";X$
1910 IF X$="1" THEN 300
1920 IF X$="2" THEN POKE 53269,0:GOTO 195
     0
1930 GOTO 1890
1940 GOSUB2340:GOSUB2330
1950 GOSUB2380
1960 GOSUB 440
1970 GOSUB2070
1980 X=PEEK(197):IFX=64THEN1960
1990 IFX=39THEN2230
2000 IFX=30THEN1950
2010 IFX=34THEN2420
2020 IFX=62THEN 1190
2030 IF X=41 THEN 2250
2040 IF X=59 THEN 1890
2050 GOTO1960
2060 IFX1=0ANDY1=0ANDFB=0THENRETURN
2070 IFCR(1,1)=8ANDY1=1THENY1=0
2080 IFCR(1,1)=1ANDY1=-1THENY1=0
2090 IFCR(1,2)=1ANDX1=-1THENX1=0
2100 IFCR(1,2)=8ANDX1=1THENX1=0
2110 CR(1,1)=CR(1,1)+Y1:CR(1,2)=CR(1,2)+
     X1
2120 IFFB=0THENRETURN
2130 NP=1237+(CR(1,1)*40)+CR(1,2):GC=PEE
     K(NP)
2140 IFGC=65THENGC=64:GOTO2170
2150 GC=65
2160 POKENP,GC:POKES+NP,0
2170 CT=0:XY=0
2180 FORX=7TO0STEP -1
2190 Y=PEEK(1238+(CR(1,1)*40)+X)
2200 Y=Y-64:XY=XY+((2↑CT)*Y):CT=CT+1:NEX
     T
2210 POKE14863+CR(1,1),XY:RETURN
2220 PRINT "■CTRL BLK■■HOME■■5CRSRDOWN■"
     :FORX=14864TO14871
2230 N1=PEEK(X):PRINT"■SHIFT
     CRSRUP■";:PRINT"■7CRSRRIGHT■";N1:NEX
     T:GOTO1960
2240 OPEN 4,4:B=14863:CT=1:SS=1237
2250 PRINT#4,"  ":FOR I=1TO8:X=PEEK(B+I):
     PRINT#4,X;" ";:NEXT
2260 PRINT#4,"  ":FOR I=1TO8:FOR A=1TO8
2270 X=PEEK(SS+(40*CT)+A):IF X=64 THEN P
     $=CHR$(119):GOTO 2300
2280 P$=CHR$(113)
2290 PRINT#4,P$;:NEXT A
2300
```

Continued on p. 49

***Building* . . . from p. 23**

VIC-20 CHARACTER EDITOR (VIC-20)

Explanation of the Program

Line Nos.	
100	Lowers BASIC programming area.
110	Copies ROM characters to RAM.
120-130	Reads in special characters.
150-260	Main program loop.
270-390	Subroutine to input cursor direction information.
400-450	Moves cursor.
460-550	Updates character behind cursor and character above assembly grid.
560-570	Subroutine to print decimal codes on screen.
580-600	Part of End of Program routine.
610-650	Subroutine to put assembly grid on screen.
660-690	Subroutine to flash cursor.
700-750	Initialization routine.
760-950	Converts character definitions to DATA statements.
960-990	End of program routine.

DON'T type in lines 30 to 90. The title of the program is required for copyright purposes, but its inclusion will make the program too big for the VIC-20's memory.

VIC-20

```
 30 REM  **********************
 40 REM  *  CHARACTER EDITOR  *
 50 REM  **********************
 60 REM  BY JOHN THRASHER
 70 REM  HOME COMPUTER MAGAZINE
 80 REM  VERSION 4.2.1
 90 REM  V20 BASIC
100 POKE52,27:POKE56,27:CLR:PRINT"■SHIFT CLR■"
110 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT:POKE36869,255:POKE36879,25
120 FORI=7376TO7376+(2*8)-1:READA:POKEI,A:NEXT
130 DATA 255,129,129,129,129,129,129,129,255,255,255,255,255,255,255,255
140 INPUT"■CTRL BLK■■3CRSRDOWN■2CRSRRIGHT■USING JOYSTICKS Y OR N■2CRSRDOWN■";J$
150 PRINT"■SHIFT CLR■":GOSUB 700
160 DIM CR(1,2):S=30720:CL=0
170 GOSUB 620:GOSUB 610
180 GOSUB 660
190 GOSUB 270
200 GOSUB 400
210 X=PEEK(197):IFX=64THEN180
220 IFX=28THEN 560
230 IFX=51THEN170
240 IFX=20THEN 760
250 IFX=48THEN 580
260 GOTO 180
270 X1=0:Y1=0:FB=0:POKE37154,127:XT%=PEEK(37152)AND128:POKE37154,255
280 IFJ$="N"THEN 330
290 XT%=XT%OR(PEEK(37137)AND127)
300 X1=SGN(XT%AND16)-SGN(XT%AND128)
310 Y1=SGN(XT%AND4)-SGN(XT%AND8)
320 FB=1-SGN(XT%AND32):RETURN
330 KY=PEEK(197):IF KY=64THENRETURN
340 IFKY=41THENX1=-1:RETURN
350 IFKY=18THENX1=1:RETURN
360 IFKY=49THENY1=-1:RETURN
370 IFKY=26THENY1=1:RETURN
380 IFKY=32THENFB=1
390 RETURN
400 IFX1=0ANDY1=0ANDFB=0THENRETURN
410 IFCR(1,1)=8ANDY1=1THENY1=0
420 IFCR(1,1)=1ANDY1=-1THENY1=0
430 IFCR(1,2)=1ANDX1=-1THENX1=0
440 IFCR(1,2)=8ANDX1=1THENX1=0
450 CR(1,1)=CR(1,1)+Y1:CR(1,2)=CR(1,2)+X1
460 IFFB=0THENRETURN
470 NP=7820+(CR(1,1)*22)+CR(1,2):GC=PEEK(NP)
480 IFGC=27THENGC=26:GOTO 500
490 GC=27
500 POKENP,GC:POKES+NP,0
510 CT=0:XY=0
520 FORX=7TO0STEP-1
530 Y=PEEK(7821+(CR(1,1)*22)+X)
540 Y=Y-26:XY=XY+((2↑CT)*Y):CT=CT+1:NEXT
550 POKE7391+CR(1,1),XY:RETURN
560 PRINT"■CTRL BLK■■HOME■■6CRSRDOWN■"
    :FORX=7392TO7399
570 N1=PEEK(X):PRINT"■SHIFT CRSR UP■":PRINT"■3CRSRRIGHT■";N1:NEXT:GOTO180
580 IFSD=DSTHEN 990
590 IFD$="N"THEN 990
600 GOTO 960
610 FORI=0TO7:POKEI+7392,0:NEXT:POKE7758,0:RETURN
620 PRINT"■CTRL BLK■■SHIFT CLR■■6CRSRDOWN■"
630 FORI=1TO8:GOSUB 650:NEXT
```

Building

VIC-20

```
640 CR(1,1)=1:CR(1,2)=1:RETURN
650 PRINTAB(9)"ZZZZZZZZ":RETURN
660 AD=7820+(CR(1,1)*22)+CR(1,2)
670 CS=PEEK(AD):CC=PEEK(AD+S):POKE AD,3
    5:POKE S+AD,0
680 FORX=1TO200:NEXT
690 POKEAD,CS:POKES+AD,CC:RETURN
700 PRINT" SHIFT CLR CTRL BLK DO YOU W
    ANT TO CREATE DATA STATEMENTS?"
710 INPUT"EN"THENRETURN
720 IFD$="N"THENRETURN
730 INPUT"ENTER STARTING LINE NUMBER
    2CRSRDOWN";SL
740 INPUT"ENTER INCREMENT VALUE 2CRSRDO
    WN 7CRSRRIGHT";IV
750 SL=SL-IV:DS=6913:SD=DS:RETURN
760 SA=7392:EA=7399:DT=131:CM=44
770 DP=SD+2:SL=SL+IV:PRINT"HOME";SL
780 LH=INT(SL/256):LL=SL-(256*LH)
790 POKE DP,LL:POKE DP+1,LH:POKE DP+2,D
    T:POKE DP+3,32:DP=DP+4
800 FORX=1TO8
810 SN=PEEK(SA):CT=2:IFSN=0THEN CA(CT)=
    10:GOTO 840
820 CA(CT)=INT((SN+.0001)/10↑CT):SN=SN-
    (CA(CT)*10↑CT):CT=CT-1:IFCT>-1THEN
    820
830 CT=2
840 IFCA(CT)>9THEN 870
850 IFCA(CT)=0THENCT=CT-1:GOTO 850
860 CA(CT)=CA(CT)+48:POKE DP,CA(CT):DP=
    DP+1:CT=CT-1:GOTO 880
870 POKEDP,48:DP=DP+1:GOTO 890
880 IFCT>-1 THEN 860
890 SA=SA+1:IF SA>EATHENX=8:GOTO 910
900 IFX<8THENPOKEDP,CM:DP=DP+1
910 NEXT
920 POKEDP,0:DP=DP+1
930 SH=INT(DP/256):LS=DP-(SH*256)
940 POKESD,LS:POKESD+1,SH:SD=DP
950 POKEDP,0:POKEDP+1,0:GOTO 180
960 GETA$:IFA$<>""THEN960
970 DP=DP+2:PH=INT(DP/256):PL=DP-(PH*25
    6)
980 POKE5000,PL:POKE5001,PH
990 POKE45,PEEK(5000):POKE46,PEEK(5001)
    :POKE44,27
1000 POKE36869,240:PRINT" SHIFT CLR":EN
     D
```

HCM

Quick Pixel . . . from p. 47

COMMODORE 64

```
2310 CT=CT+1:PRINT#4,"";:NEXT I
2320 PRINT#4,:CLOSE4:GOTO1960
2330 FORI=0TO7:POKEI+14864,0:NEXT:POKE11
     62,66:POKES+1162,0:RETURN
2340 PRINT" CTRL BLK SHIFT CLR 5CRSRDO
     WN"
2350 FORI=1TO8:GOSUB 2370:NEXT
2360 CR(1,1)=1:CR(1,2)=1:RETURN
2370 PRINTTAB(14)" SHIFT *";:RETURN
2380 AD=1237+(CR(1,1)*40)+CR(1,2)
2390 CS=PEEK(AD):CC=PEEK(AD+S):POKE AD,8
     6:POKE S+AD,0
2400 FORX=1TO300:NEXT
2410 POKEAD,CS:POKES+AD,CC:RETURN
2420 SA=14864:EA=14871:DT=131:CM=44
2430 DP=SD+2:SL=SL+IV:PRINT"HOME";:SL
2440 LH=INT(SL/256):LL=SL-(256*LH)
2450 POKE DP,LL:POKE DP+1,LH:POKE DP+2,D
     T:POKE DP+3,32:DP=DP+4
2460 FORX=1TO8
2470 SN=PEEK(SA):CT=2:IFSN=0THEN CA(CT)=
     10:GOTO2500
2480 CA(CT)=INT((SN+.0001)/10↑CT):SN=SN-
     (CA(CT)*10↑CT):CT=CT-1:IFCT>-1THEN2
     480
2490 CT=2
2500 IFCA(CT)>9THEN2530
2510 IFCA(CT)=0THENCT=CT-1:GOTO2510
2520 CA(CT)=CA(CT)+48:POKE DP,CA(CT):DP=
     DP+1:CT=CT-1:GOTO2540
2530 POKEDP,48:DP=DP+1:GOTO2550
2540 IFCT>-1 THEN2520
2550 SA=SA+1:IF SA>EATHENX=8:GOTO2570
2560 IFX<8THENPOKEDP,CM:DP=DP+1
2570 NEXT
2580 POKEDP,0:DP=DP+1
2590 SH=INT(DP/256):LS=DP-(SH*256)
2600 POKESD,LS:POKESD+1,SH:SD=DP
2610 POKEDP,0:POKEDP+1,0:GOTO1960
2620 GETA$:IF A$<>""THEN 2620
2630 RETURN
```

HCM

ANNOUNCING . . .
VERSION 2.0

EXTENDED PASCAL FOR YOUR IBM PC, PC jr., APPLE CP/M, MSDOS, CP/M 86, CCP/M, OR CP/M 80

NOW . . .
WITH
WINDOWING
$49.95

"What I think the computer industry is headed for: well documented, standard, plenty of good features, and a reasonable price."

Jerry Pournelle,
Byte, February 1984

"The Perfect Pascal"
Alan R. Miller,
Interface Age, January 1984

If you already own Turbo Pascal version 1.0, you can upgrade to 2.0 for $29.95. Just send in your old master with your check. (Manual update included of course).

NEW FEATURES

WINDOWING!
. . . This is a real shocker. On the IBM PC or PC jr. you'll now have a procedure to program windows. . . . Any part of the screen can be selected as a window and all output will automatically go to this part of the screen only. As many windows as you please can be used from the same program.

AUTOMATIC OVERLAYS!
. . . No addresses or memory space to calculate, you simply specify OVERLAY and TURBO PASCAL will do the rest.

GRAPHICS, SOUND AND COLOR SUPPORT
. . . For your IBM PC or JR!

FULL HEAP MANAGEMENT!
. . . via dispose procedure.

OPTIONAL 8087 SUPPORT!
. . . Available for an additional charge.
If you have a 16 bit computer with the 8087 math chip—your number crunching programs will execute up to 10X faster!

ORDER YOUR COPY OF TURBO PASCAL VERSION 2.0 TODAY

For VISA and MasterCard orders call toll free: **1-800-227-2400 x968**
In CA: **1-800-772-2666 x968**

(lines open 24 hrs, 7 days a week)
Dealer & Distributor Inquiries welcome
408-438-8400

CHOOSE ONE (please add $5.00 for shipping and handling for U.S. orders)

_____ Turbo Pascal 2.0 $49.95
_____ Turbo Pascal 2.0 with 8087 support $89.95
_____ Update (1.0 to 2.0) Must be accompanied by the original master $29.95
_____ Update (1.0 to 8087) Must be accompanied by the original master $69.95

Check _____ Money Order _____
VISA _____ MasterCard _____
Card #: _____
Exp. date: _____ Shipped UPS

BORLAND INTERNATIONAL

Borland International
4113 Scotts Valley Drive
Scotts Valley, California 95066
TELEX: 172373

My system is: 8 bit _____ 16 bit _____
Operating System: CP/M 80 _____
CP/M 86 _____ MS DOS _____ PC DOS _____
Computer: _____ Disk Format: _____
Please be sure model number & format are correct.
NAME: _____
ADDRESS: _____
CITY/STATE/ZIP: _____
TELEPHONE: _____

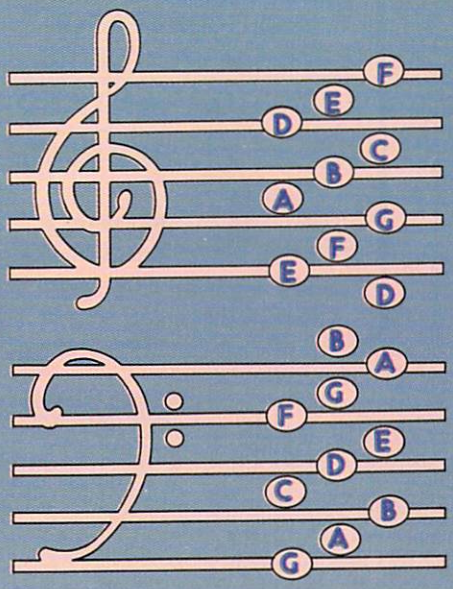
California residents add 6% sales tax. Outside U.S.A. add $15.00. (If outside of U.S.A. payment must be by bank draft payable in the U.S. and in U.S. dollars.) Sorry, no C.O.D. or Purchase Orders. A5

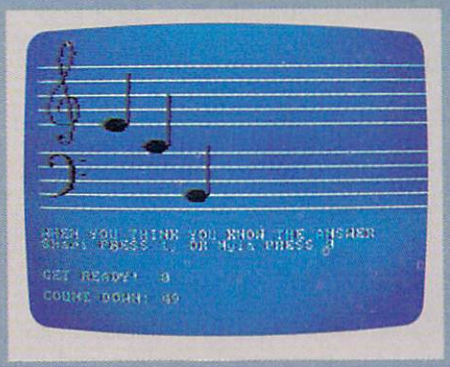
Musical Mystery Words

Learning to read music can be about as much fun as learning a second language, and almost as difficult. But if you have gotten to the point where you can begin to identify each note's letter name, *Mystery Words* can provide a painless way to imprint them on your mind.

The program was designed to supplement elementary school music instruction in the names of the notes of each staff. *Mystery Words* is a game that uses the letter names of musical notes to spell a variety of words, such as "deaf," "bed," or "cage." The program will randomly choose one of these words and represent each letter in music notation on the staff (in the bass or treble clef, or both). It will also play the sound of each note. The players must identify the word within a time limit, and they have the option to exclude some of the more difficult words from their choices at the game's start.

If you are in the process of learning to read music, or you just want a review, *Mystery Words* is a fun way to test your knowledge quickly, without ever picking up an instrument.





Using the Program

The first screen following the title screen will ask you to input two players' names, and the next screen will ask you whether you would like to exclude the difficult words from the word list. If you want the simple list, press Y, and if you want the more complicated words, press N.

The second screen will give you three options. You can use treble clef notes only, bass clef notes only, or mix them together at the same time. Each of the seven notes is repeated three times in the scale of the treble and bass clef, as the diagram illustrates.

The next screen will display the musical staff and the two clefs. The message at the bottom of the screen indicates that the computer is waiting for both players to respond by pressing 0 and 1. The program will not continue until both players have indicated that they are ready by pressing their control keys. Next the notes are displayed until one of the control keys is pressed, then the screen will clear so that the player who elected to answer the question can no longer see the notes. The player must already know the answer before pressing the control key. At the bottom of the screen the program will indicate the player who is to input the answer, which is helpful if both players hit their control keys at the same time. If the player gives a wrong answer, the other player then gets a chance to answer the problem.

After you select a clef option, you will be prompted to input a time between 3 and 60 seconds. This will be the maximum amount of time you will have to figure out the problem, and to press keys 0 or 1 to indicate that you want to try an answer.

After you enter the time limit, you are shown a single screen of instructions which briefly tell you which keys to use and when to use them.

When the players are ready, a 5-second countdown occurs before the notes are displayed and played. The countdown includes a counter on the screen after the message GET READY! When the countdown is finished, the notes will be displayed one at a time, accompanied by the tone of the note. Once the notes have finished playing, both players have only a limited amount of time to indicate that they have an answer by pressing 0 or 1. If both players fail to indicate that they have an answer before time is up, they both lose their turn and the program continues on to the next word.

If both players miss the question, the correct answer will be displayed and the program will continue to the next problem. When a player gets the correct answer, his score is updated and the total scores for both of the players are momentarily displayed on the screen.

The program will continue for as long as you like. If at any time you would like to stop the program, simply press [F1] (the [Fn] key and the 1 key at the same time). This gives the program a more orderly halt than if you simply pressed [Fn] [B] (BREAK).

—William Balthrop

Continued on p. 54

For the sake of simplicity and easy reading, we chose white musical staffs and notes, displayed on a black background for the Apple version of *Mystery Words*. This way, the program runs identically on color or monochrome televisions and monitors, with the note displays done via the high-resolution (hi-res) screen. In addition, as each note is displayed, a machine language music routine plays a tone corresponding to the note on the screen, reinforcing the identification of written notes with particular tones.

The values needed to produce the tones are READ into an integer array (NT%(NT,OC)), where NT identifies the NoTe (A=1, B=2, etc.) and OC identifies the OCtave of the note (1 for lowest, 3 for highest). Also, the NT variable is used to select the subroutine for that particular note (see line 1260). Once control is transferred to the subroutine for that note, the NT variable is used to select the position of the note on the staff (or staves) that is being used for the game.

Earlier in the program (line 480), AN$ is selected at random from the word list. Then the letters are selected one at a time by the MID$ command in line 1260. Also in line 1260, the ASCII value is decreased by 64 to index the proper subroutine. For example, let's say that the letter selected is A. The ASCII for A is 65, so the computer will select the first subroutine in the ON__GOSUB statement (GOSUB

Note: TI readers who have been with us from the beginning will remember the TI version of Mystery Words *by John and Norma Clulow (Now available in the* Best of 99'er *article* Let's Learn Notes *starting on page 199).*

Mystery Words

640) because 65-64 = 1. The octave of the note to be displayed is determined by whatever clef is being used (CF = 1 for treble and CF = 2 for bass).

Some notes can actually be placed in one of two different places on the staves. For example, there are two A's on the bass clef staff—the bottom space and the top line. If a note can appear in either of two places, the choice is made randomly by the appropriate routine between lines 1200 and 1250. Depending upon which OCtave is selected, the correct starting row (Y position) for the display is placed in the variable RS. A 1 or 2 in the LS variable indicates whether the note is to be drawn on a line or in a space; 1 is for line, and 2 is for space. Now, when the computer returns to line 1270, it uses all of these variables to display the proper note in the proper place when it finally branches to either line 1430 or 1450.

—Roger Wood

MYSTERY WORDS (Apple)
Explanation of the program

Line Nos.	
100-170	Program Header.
180-200	Title Screen.
210-240	DIM arrays and initialize constants.
250-260	Display second screen (get names).
270-290	Display third screen (exclude some words?).
300-340	Display fourth screen (which clef of two clefs?).
350-390	Display instruction screen.
400-460	Display staff and wait for ready signal.
470-510	Display problem and wait for response.
520-630	Get answer, evaluate, and update scores.
640-1190	Subroutines to prepare to display notes.
1200-1250	Select random numbers.
1260-1270	Routine to display notes.
1280-1290	Routine to draw staff.
1300-1360	Routine to draw treble clef.
1370-1420	Routine to draw bass clef.
1430-1440	Routine to draw note in a space.
1450-1480	Routine to draw note on a line.
1490-1500	Routine to play a note.
1510-1560	Poke machine language music routine.
1570-1580	Data for word lists.
1590	Data for pitches of notes.

APPLE II Series

```
100 REM  **********************
110 REM  *  MYSTERY WORDS  *
120 REM  **********************
130 REM  BY
140 REM  THE HCM STAFF
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.2.1
170 REM  APPLE II SERIES APPLESOFT
```

APPLE II Series

```
180 REM  TITLE SCREEN
190 TEXT : HOME : VTAB 13: HTAB 12: PRI
    NT "MYSTERY WORDS": VTAB 22: HTAB 7
    : PRINT "PRESS RETURN TO BEGIN"
200 GET K$: IF K$ < > CHR$ (13) THEN
    200
210 HOME : DIM NT%(7,3),S$(59),P$(1):NS
    = 59
220 GOSUB 1510
230 FOR I = 1 TO 59: READ S$(I): NEXT I
240 FOR I = 1 TO 7: FOR J = 1 TO 3: REA
    D NT%(I,J): NEXT J: NEXT I
250 REM  SECOND SCREEN
260 TEXT : HOME : VTAB 5: PRINT "PLAYER
    1 — ENTER YOUR NAME": INPUT P$(0
    ): VTAB 10: PRINT "PLAYER 2 — ENTER
    YOUR NAME": INPUT P$(1): HOME
270 REM  THIRD SCREEN
280 VTAB 8: PRINT "SOME OF THE WORDS IN
    THE WORD LIST ARE NOT READILY UNDE
    RSTOOD BY YOUNGER CHILDREN": PRINT
    "WOULD YOU LIKE TO EXCLUDE THESE WO
    RDS? (Y/N)"
290 GET K$: IF K$ = "Y" THEN NS = 52
300 REM  FOURTH SCREEN
310 HOME : VTAB 3: HTAB 7: PRINT "THREE
    DRILLS ARE AVAILABLE": VTAB 7: PRI
    NT "1) TREBLE CLEF"
320 VTAB 9: PRINT "2) BASS CLEF": VT
    AB 11: PRINT "3) TREBLE AND BASS CLE
    F": VTAB 14: PRINT "YOUR CHOICE
    (1, 2, OR 3)": ;
330 GET K$: IF K$ < "1" OR K$ > "3" THE
    N 330
340 CF = VAL (K$)
350 REM  INSTRUCTION SCREEN
360 HOME : HTAB 14: PRINT "INSTRUCTIONS
    ": PRINT "KEY FUNCTION": PRINT : PRINT "0
    " ; P$(0); " SCONTROL KEY"
370 PRINT : PRINT "1" ; P$(1); " SCONTRO
    L KEY": PRINT : PRINT "A—G SELECT NOTES
    THROUGH G"
380 VTAB 11: PRINT "BOTH KEYS 0 AND 1 MUST BE
    PRESSED TO START A NEW TURN.": PRINT
    "WHEN YOU HAVE DECODED THE WORD, PRESS
    YOUR CONTROL KEY, AND THEN USE KEYS A THROUGH G TO
    ENTER YOUR ANSWER."
390 VTAB 22: HTAB 5: PRINT "PRESS RETUR
    N TO CONTINUE": GET K$: IF K$ < > CHR$ (13) THEN 390
400 HOME : F = 0: HGR : HCOLOR= 3
410 IF CF < > 2 THEN RS = 20: GOSUB 12
    80: GOSUB 1300
420 IF CF = 1 THEN 440
430 RS = 80: GOSUB 1280: GOSUB 1370
440 HOME : VTAB 21: PRINT P$(0); " PRESS
    0 WHEN YOU'RE READY": PRINT P$(1);
    " PRESS 1 WHEN YOU'RE READY"
450 GET A$: IF A$ > "1" OR A$ < "0" THE
    N 450
460 GET B$: IF (B$ = A$) OR (B$ > "1")
    OR (B$ < "0") THEN 460
470 HOME : VTAB 22: PRINT "PRESS YOUR C
    ONTROL KEY WHEN YOU THINK YOU KNOW
    THE ANSWER"
480 AN = INT ( RND (1) * NS) + 1: AN$ =
    S$(AN): GOSUB 1260
```

APPLE II Series

```
490 GET A$: IF A$ < "0" OR A$ > "1" THEN 490
500 F = 1: A = VAL (A$): IF A = 1 THEN B = 0: GOTO 520
510 IF A = 0 THEN B = 1
520 TEXT : HOME
530 VTAB 10: PRINT "OK, "; P$(A)
540 PRINT "KEY IN YOUR ANSWER.": PRINT "IT HAS "; LEN (AN$);" LETTERS"
550 VTAB 15: FOR I = 1 TO LEN (AN$)
560 GET K$: IF K$ < "A" OR K$ > "G" THEN 560
570 HTAB I: PRINT K$;: IF K$ = MID$ (AN$,I,1) THEN NEXT I: GOTO 600
580 PRINT : PRINT "THAT'S NOT RIGHT "; P$(A): POKE - 16368,0: FOR DE = 1 TO 2000: NEXT : POKE - 16368,0: A = B: IF F = 1 THEN F = 0: HOME : VTAB 10: PRINT "IT'S YOUR TURN TO TRY "; P$(A): GOTO 540
590 GOTO 630
600 PRINT : PRINT "THAT'S CORRECT "; P$(A)
610 IF A = 1 THEN S1 = S1 + 1: GOTO 630
620 S0 = S0 + 1
630 PRINT "THE SCORE IS:": PRINT : PRINT P$(0);" "; S0: PRINT P$(1); " "; S1: GOTO 390
640 REM A NOTE
650 IF CF = 1 THEN OC = 3: GOTO 680
660 IF CF = 2 THEN GOSUB 1200: GOTO 680
670 GOSUB 1240
680 IF OC = 1 THEN RS = 110: LS = 2: GOTO 710
690 IF OC = 2 THEN RS = 80: LS = 1: GOTO 710
700 RS = 40: LS = 2
710 RETURN
720 REM B NOTE
730 IF CF = 1 THEN OC = 3: GOTO 760
740 IF CF = 2 THEN GOSUB 1200: GOTO 760
750 GOSUB 1240
760 IF OC = 1 THEN RS = 110: LS = 1: GOTO 790
770 IF OC = 2 THEN RS = 70: LS = 2: GOTO 790
780 RS = 40: LS = 1
790 RETURN
800 REM C NOTE
810 IF CF = 1 THEN OC = 3: GOTO 840
820 IF CF = 2 THEN OC = 1: GOTO 840
830 GOSUB 1240
840 IF OC = 1 THEN RS = 100: LS = 2: GOTO 870
850 IF OC = 2 THEN RS = 70: LS = 1: GOTO 870
860 RS = 30: LS = 2
870 RETURN
880 REM D NOTE
890 IF CF = 1 THEN GOSUB 1220: GOTO 920
900 IF CF = 2 THEN OC = 1: GOTO 920
910 GOSUB 1240
920 IF OC = 1 THEN RS = 100: LS = 1: GOTO 950
930 IF OC = 2 THEN RS = 60: LS = 2: GOTO 950
940 RS = 30: LS = 1
```

APPLE II Series

```
950 RETURN
960 REM E NOTE
970 IF CF = 1 THEN GOSUB 1220: GOTO 100
980 IF CF = 2 THEN OC = 1: GOTO 1000
990 GOSUB 1240
1000 IF OC = 1 THEN RS = 90: LS = 2: GOTO 1030
1010 IF OC = 2 THEN RS = 60: LS = 1: GOTO 1030
1020 RS = 20: LS = 2
1030 RETURN
1040 REM F NOTE
1050 IF CF = 1 THEN GOSUB 1220: GOTO 1080
1060 IF CF = 2 THEN OC = 1: GOTO 1080
1070 GOSUB 1240
1080 IF OC = 1 THEN RS = 90: LS = 1: GOTO 1110
1090 IF OC = 2 THEN RS = 50: LS = 2: GOTO 1110
1100 RS = 20: LS = 1
1110 RETURN
1120 REM G NOTE
1130 IF CF = 1 THEN OC = 3: GOTO 1160
1140 IF CF = 2 THEN GOSUB 1200: GOTO 1160
1150 GOSUB 1240
1160 IF OC = 1 THEN RS = 120: LS = 1: GOTO 1190
1170 IF OC = 2 THEN RS = 80: LS = 2: GOTO 1190
1180 RS = 50: LS = 1
1190 RETURN
1200 REM RND 1 OR 2
1210 OC = INT ( RND (1) * 2) + 1: RETURN
1220 REM RND 2 OR 3
1230 OC = INT ( RND (1) * 2) + 2: RETURN
1240 REM RND 1, 2, OR 3
1250 OC = INT ( RND (1) * 3) + 1: RETURN
1260 FOR I = 1 TO LEN (AN$): NT = ASC ( MID$ (AN$,I,1)) - 64: ON NT GOSUB 640,720,800,880,960,1040,1120
1270 CS = I * 20 + 20: ON LS GOSUB 1450,1430: GOSUB 1490: NEXT I: RETURN
1280 REM DRAW STAFF
1290 CS = 0: FOR RW = RS TO RS + 40 STEP 10: HPLOT CS,RW TO CS + 270,RW: NEXT RW: RETURN
1300 REM TREBLE CLEF
1310 CL%(0) = 1: CL%(1) = 3: CL%(2) = 5: CL%(3) = 7: CL%(4) = 9: CL%(5) = 12: CL%(6) = 15: CL%(7) = 17: CL%(8) = 19: CL%(9) = 21: CL%(10) = 23
1320 FOR I = 1 TO 2: IF I = 2 THEN FOR J = 0 TO 10: CL%(J) = CL%(J) + 1: NEXT J
1330 HPLOT CL%(4),11 TO CL%(6),65: HPLOT TO CL%(4),70: HPLOT TO CL%(4),67: HPLOT TO CL%(4),67: HPLOT CL%(4),11 TO CL%(8),18
1340 HPLOT CL%(8),19 TO CL%(7),24: HPLOT CL%(8),21 TO CL%(3),38: HPLOT CL%(7),25 TO CL%(1),44: HPLOT TO CL%(0),50: HPLOT TO CL%(1),55: HPLOT TO CL%(3),60
```

Continued on p. 54

Mystery Words . . . from p. 53

APPLE II Series

```
1350 HPLOT CL%(7),60 TO CL%(9),55: HPLOT
     TO CL%(10),50: HPLOT TO CL%(8),4
     5: HPLOT TO CL%(6),42: HPLOT TO C
     L%(3),47: HPLOT TO CL%(4),50
1360 NEXT I: RETURN
1370 REM BASS CLEF
1380 CL%(0) = 1:CL%(1) = 3:CL%(2) = 5:CL
     %(3) = 7:CL%(4) = 9:CL%(5) = 12:CL%
     (6) = 15:CL%(7) = 17:CL%(8) = 19:CL
     %(9) = 21:CL%(10) = 23
1390 FOR I = 1 TO 2: IF I = 2 THEN FOR
     J = 0 TO 10:CL%(J) = CL%(J) + 1: NE
     XT J
1400 FOR K = 88 TO 89: HPLOT CL%(3),K TO
     CL%(1),K: NEXT : HPLOT TO CL%(1),
     85 TO CL%(4),80
1410 HPLOT CL%(7),80 TO CL%(9),85 TO CL%
     (10),90 TO CL%(10),96 TO CL%(9),100
     TO CL%(8),102 TO CL%(6),106 TO CL%
     (3),111
1420 HPLOT CL%(10) + 5,85 TO CL%(10) + 5
     ,87: HPLOT CL%(10) + 5,93 TO CL%(10
     ) + 5,95: NEXT : RETURN
1430 REM DRAW NOTE IN SPACE
1440 HPLOT CS,RS + 4 TO CS,RS + 7 TO CS +
     2,RS + 9 TO CS + 6,RS + 9 TO CS +
     9,RS + 6 TO CS + 9,RS + 3 TO CS +
     6,RS + 1 TO CS + 3,RS + 1 TO CS,RS
     + 4: RETURN
1450 REM DRAW NOTE ON LINE
1460 HPLOT CS,RS - 1 TO CS + 2,RS - 4 TO
     CS + 6,RS - 4 TO CS + 9,RS - 1: HP
     LOT CS,RS + 1 TO CS + 2,RS + 4 TO C
     S + 6,RS + 4 TO CS + 9,RS + 1
1470 IF OC = 2 AND NT = 3 THEN HPLOT CS
     - 2,RS TO CS + 11,RS
1480 RETURN
1490 REM MAKE NOTE
1500 POKE 768,10: POKE 769,NT%(NT,OC): C
     ALL 770: RETURN
1510 FOR I = 768 TO 795
1520 READ IN
1530 POKE I,IN
1540 NEXT I
1550 RETURN
1560 DATA 0,210,172,1,3,174,1,3,232,20
     8,253,169,4,32,168,252,173,48,192,1
     36,208,239,206,0,3,208,231,96
1570 DATA AGE,ACE,ADD,ABE,BAD,BAA,BAG,B
     EE,BEG,BED,BEA,CAB,CAD,DAB,DAD,DEB,
     DEE,EBB,EGG,FAB,FAD,FED,GAG,GEE,GAB
     ,ADDED,DABBED,CAGED,DEFACED,BAGGED
1580 DATA AGED,BABE,BEEF,BEAD,CAFE,CAG
     E,DEAD,DEAF,DEED,EDGE,FACE,FEED,GAG
     E,FEE,CAGED,DEFACE,DECADE,CABBAGE,E
     DGED,BAGGAGE,EBBED,FACED,BADE,ACCED
     E,AGAGE,BAGDAD,DECA,FACADE,EFFACE
1590 DATA 52,166,222,74,177,228,85,182,
     231,105,192,237,125,202,241,134,206
     ,243,24,151,214
```

HCM

Mystery Words . . . from p. 51

MYSTERY WORDS (IBM PCjr) Explanation of the program

Line Nos.	
100-190	Program header.
200-240	Initialize program variables, graphics information, and the screen graphics mode.
250	Display the title screen.
260-300	Input players' names and ask if the players would like to omit the more difficult words.
310-340	Input option to use treble, bass, or both. Also input option for length of time to answer a problem.
350-390	Instruction page.
400-430	Input both players' control keys (1 and 0) before starting the next turn.
440-450	Countdown before displaying the problem.
460	Branch to the routine that displays the problem and plays the note.
470-490	Wait for one of the players to enter either 0 or 1, indicating readiness to answer.
500	Out of time routine when neither player answers the problem.
510-540	Player enters the answer. Loop has a time restriction determined by the amount of time players decided on at the beginning of the game.
550	Check for a correct answer.
560-590	Either wrong answer or out-of-time messages.
600-610	Player got the right answer. Update and display score.
620	Branch to subroutines that display and play the notes.
630-690	Determine which of the three octaves the note should play, and the vertical position of the notes on the screen.
700	Draw the note and play it.
710-720	Draw the music page graphics, with treble and bass clefs.
730	Subroutine to clear the bottom area of the screen.
740-750	DATA statements containing the program's vocabulary.
760	END of the program.

IBM PCjr

```
100 REM ********************
110 REM *  MYSTERY WORDS  *
120 REM ********************
130 REM BY
140 REM THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.2.1
170 REM IBM PCjr
180 REM CARTRIDGE BASIC
190 REM
```

Mystery Words

IBM PCjr

```
200 CLEAR ,,,32768!:SCREEN 5:COLOR 7,1:
    CLS:DIM S$(59):BEEP OFF:SOUND ON
210 FOR Z=1 TO 59:READ S$(Z):NEXT Z
220 BASS$="EL2UR3HL2U3RUR3HR2D2HD2RDRD
    RD3ED5RNU3RND7GD8LNU3D3LNU2DBM+0,-10
    RULD"
230 TREB$="HU4EUEUERERE6FRFD2FDFD4GD2GDG
    GGL13HLHHUHUHU8EU2EUEUE7UEEU3UEUEUE
    UEUUEU3EUUHUUHUH3G3DGDDGD3GD3FD3FD4
    FD3FD2FD2FD2FD2FD3FD4FD5FD7GD4GDG3L
    3H3U3ED3RELU"
240 NOTE$="NU35LNU35H3LHL9GLR11FRL14GR1
    6FL18DR19GL17FR15GL13RFR9"
250 KEY OFF:LOCATE 12,13:PRINT "MYSTERY
    WORDS":FOR D=1 TO 1000:NEXT D:CLS
260 LOCATE 5,1:PRINT "PLAYER 1: ENTER Y
    OUR NAME.":INPUT P1$:PRINT:PRINT:PR
    INT "PLAYER 2: ENTER YOUR NAME.":IN
    PUT P2$:CLS
270 LOCATE 2,1:PRINT "PRESS F1 AT ANY T
    IME TO QUIT"
280 LOCATE 8,1:PRINT "SOME OF THE WORDS
    IN THE WORD LIST ARE NOT READILY
    UNDERSTOOD BY YOUNGER CHILDREN.":PRI
    NT:PRINT:PRINT "WOULD YOU LIKE TO EX
    CLUDE THESE WORDS? ( Y / N )"
290 ON KEY(1) GOSUB 760:KEY(1) ON
300 A$=INKEY$:IF A$="" THEN 300 ELSE IF
    A$="Y" THEN NSET=52 ELSE IF A$="N"
    THEN NSET=59 ELSE GOTO 300
310 CLS:LOCATE 3,7:PRINT "THREE DRILLS
    ARE AVAILABLE":LOCATE 7,3:PRINT "1)
    TREBLE CLEF":LOCATE 9,3:PRINT "2)
    BASS CLEF":LOCATE 11,3:PRINT "3) TR
    EBLE AND BASS CLEF":LOCATE 15,3:PRI
    NT "YOUR CHOICE (1, 2, or 3)?
320 A$=INKEY$:IF A$="" THEN 320 ELSE IF
    A$<"1" OR A$>"3" THEN 320 ELSE CLE
    F=VAL(A$)
330 CLS:LOCATE 3,1:PRINT "ENTER THE AMO
    UNT OF TIME YOU WOULD LIKE TO HAVE
    TO ANSWER THE PROBLEM:"
340 LOCATE 6,1:INPUT "IN SECONDS (3 - 6
    0):",SEC$:IF LEN(SEC$)<1 THEN 340 E
    LSE IF SEC$<"1" OR SEC$>"60" THEN 3
    40 ELSE IF VAL(SEC$)<3 OR VAL(SEC$)
    >60 THEN 340 ELSE SEC=VAL(SEC$)
350 CLS:PRINT TAB(14);"FUNCTION":PRINT:P
    RINT "KEY FUNCTION":PRINT:PRINT "
    0 ";P2$;" CONTROL KEY":PRINT:PRINT "
    1 ";P1$;" SELECT KEY":PRINT:PRINT "A-G
    NOTES A THROUGH G"
360 PRINT:PRINT "BOTH KEYS 1 AND 0 MUST
    BE PRESSED TO START A NEW TURN.":PRI
    NT:PRINT "WHEN YOU HAVE DECODED THE
    WORD PRESS YOUR CONTROL KEY, AND THE
    N USE KEYS A THRO UGH G TO ENTER YOU
    R ANSWER.
370 LOCATE 22,5:PRINT "** PRESS ENTER T
    O CONTINUE **"
380 PALETTE 13,15:PALETTE 8,0:PLAY "MFV
    15L6"
390 A$=INKEY$:IF A$="" THEN 390 ELSE GO
    SUB 710
```

IBM PCjr

```
400 LOCATE 18,1:PRINT "WHEN YOU BOTH AR
    E READY":PRINT:PRINT "PRESS 1",,"
    AND PRESS 2";:PRINT:PRINT "PRESS 0.
    ":PRINT:PRINT "THE TURN WILL START
    WHEN BOTH ARE PRESSED.";
410 A$=INKEY$:IF A$="" OR A$<"0" OR A$>
    "1" THEN 410
420 RANDOMIZE TIMER
430 B$=INKEY$:IF B$="" THEN 430 ELSE IF
    (A$="1" AND B$<>"0") OR (A$="0" AN
    D B$<>"1") THEN 430
440 GOSUB 730:LOCATE 18,1:PRINT "WHEN Y
    OU THINK YOU KNOW THE ANSWER
    P1$;" PRESS 1, OR ";P2$;" PRESS 0":
    PRINT:PRINT:PRINT "GET READY!"
450 FOR CD=5 TO 0 STEP -1:LOCATE 22,12:
    PRINT CD:FOR TD=1 TO 500:NEXT TD:NE
    XT CD
460 A$=S$(INT(RND*NSET)+1):GOSUB 620
470 DEF SEG=0:POKE 1050,PEEK(1052):DEF
    SEG:POKE 1060,0:TM=TIMER:LOCATE 24,1
    :PRINT "COUNT DOWN:";
480 LOCATE 24,12:PRINT INT(SEC-(TIMER-T
    M));
490 IF TM+SEC<TIMER THEN 500 ELSE A$=INK
    EY$:IF A$="1" OR A$="0" THEN 510 E
    LSE 480
500 GOSUB 730:LOCATE 18,1:PRINT "OUT OF
    TIME!":PRINT:SOUND 220,9,15:PRINT
    "THE CORRECT ANSWER IS: ";ANS$:FOR T
    D=1 TO 3000:NEXT TD:CLS:GOSUB 710:
    GOTO 400
510 PLEVAL(A$):CLS:LOCATE 18,1:PRINT "EN
    TER YOUR ANSWER:";:IF PL=0 THEN "***
    ";P2$;" ELSE PRINT "***";P1$;"***"
520 PRINT:PRINT "THERE ARE ";LEN(ANS);"
    LETTERS (A through G).
530 ANS$="":CP=9:FOR Z=1 TO LEN(ANS)
540 A$=INKEY$:IF A$="" THEN 540 ELSE IF
    A$<"A" OR A$>"G" THEN 540 ELSE LOCA
    TE 17,CP:PRINT A$:CP=CP+4:ANS$=ANS$
    +A$:NEXT Z
550 IF ANS$=ANS$ THEN T(0)=0:T(1)=0:GOTO
    600
560 PLAY "L24O1BAGFEDCL6":GOSUB 730:LOCA
    TE 18,1
570 T(PL)=1:IF T(0)=1 AND T(1)=1 THEN T
    (0)=0:T(1)=0:CLS:LOCATE 12,7:PRINT
    "YOU BOTH BLEW IT!":PRINT:PRINT TAB
    (7);"THE WORD WAS ";ANS$:FOR TD=1 TO
    2000:NEXT TD:GOSUB 710:GOTO 400
580 IF PL=0 THEN PRINT P1$;" YOUR TURN
    TO TRY. . .":FOR TD=1 TO 2000:NEXT TD
    :A$="1":GOTO 510
590 PRINT P2$;" YOUR TURN TO TRY. . .":FO
    R TD=1 TO 2000:NEXT TD:A$="0":GOTO
    510
600 IF PL=0 THEN RED=RED+1 ELSE BLUE=BL
    UE+1
610 GOSUB 730:PLAY "L36O3CDEFGAB>CDEFGA
    BL602":LOCATE 18,1:PRINT P1$;"!":PR
    INT " YOUR SCORE IS: ";BLUE:PRINT:PR
    INT P2$;"!":PRINT " YOUR SCORE IS:
    ";RED:FOR TD=1 TO 3000:NEXT TD:CLS:G
    OSUB 710:GOTO 400
```

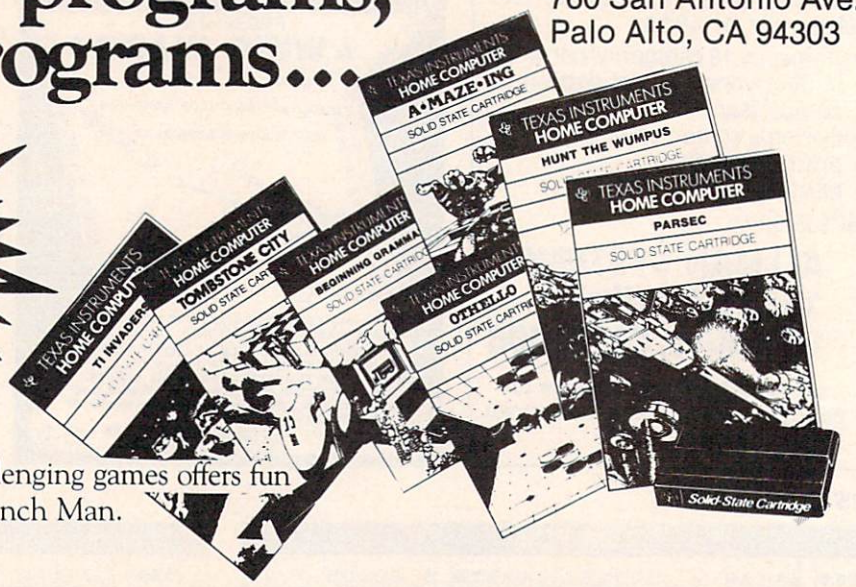
Continued on p. 149

The BACH Company

760 San Antonio Ave.
Palo Alto, CA 94303

Programs, programs, programs...

Pick any three $12.95 each

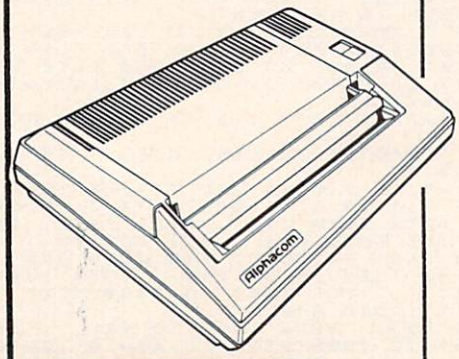


An exciting array of mind-challenging games offers fun for the whole family. Parsec. Munch Man. TI Invaders. Tombstone City. And many, many more.

Your TI 99/4A Headquarters

JUST $169.95

- Perfect for word processing, letters, charts, reports—just about anything!
- Full 80-column width
- Fast—100 characters per second
- Interface cables sold separately



Alphacom 81™
80-Column Printer

SUPER SKETCH

CREATE SUPER VIDEO GRAPHICS

As a new art form, the user can experiment with different color combinations with the simple push of a button. Children will be proud of their graphics. Adult artists can explore new techniques with little or no background in computers.

$59.95

The BOSS™ is here!

$19.95

- No-slip grip
- Fast on the draw
- Split second moves
- One-year limited warranty
- Works with Atari,® Sears and Commodore systems, plus 5 other home video games and computers (with adaptors).

TI-99/4A Adaptor _____ 12.95

Call us **TOLL FREE** at **800-227-8292** (including Alaska and Hawaii), in California call 415 **424-0800.**

MINER 2049'ER

A Review by Greg Roberts
HCM Staff

Name:	Miner 2049'er
Program Type:	Mining Adventure
Author:	Bill Hogue
Machine:	TI-99/4A (other versions available for Commodore 64 and Apple II from different distributors)
Distributor:	Tigervision Tiger Electronic Toys 909 Orchard Mundelein, IL 60060
Price:	$39.95, cartridge

System Requirements:
Joysticks

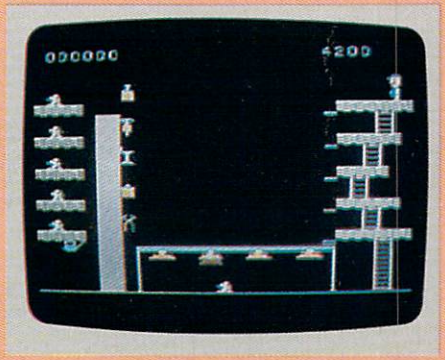
	Poor	Fair	Good	Excellent
Performance:			■	
Engrossment:				■
Documentation:			■	

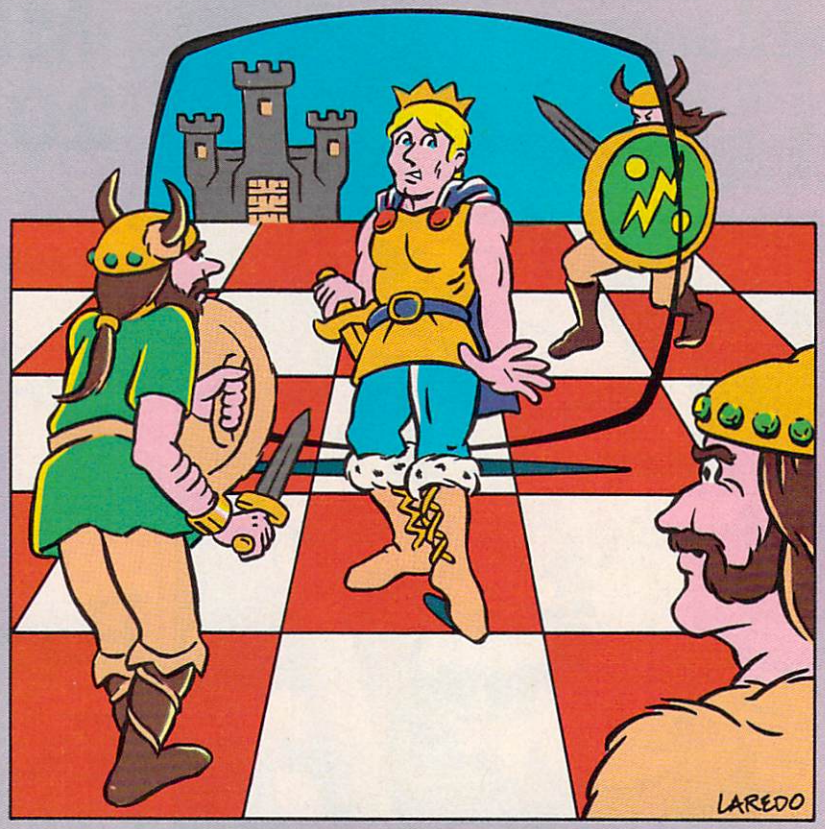
Miner 2049'er is easy at first. When the deadly mutant organisms approach, you just act as southerners do when 'gators come into the back yard: Climb the ladder that leans against the porch. Once the mutants wander off, you can sneak down from your ladder and grab the convenient pickax. This move will temporarily transform the mutants into benign beasts, so that you can destroy them if you act quickly.

This player-friendly way of drawing you into the action is desirable in any game program, as long as it's followed up with something more challenging. *Miner 2049'er* does not disappoint you. After that first easy screen, it hits you with some tough going that keeps getting more difficult right through the eighth level.

The various levels are remarkably distinct from one another. Whereas many programs put up screens that vary only slightly from the preceding level, this one

Continued on p. 77

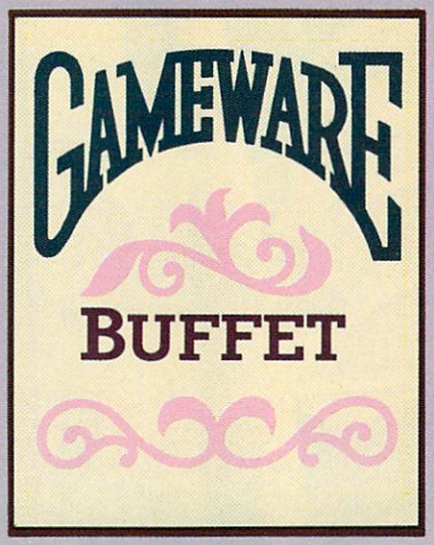




PC

A SPECIAL NOTE TO IBM PC OWNERS:
If you have an RGBI color monitor and a copy of BASICA, *Tablut* is for you too!

TABLUT

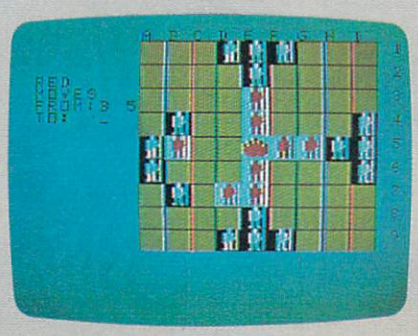


by James J. Mulligan
and the HCM Staff

Tablut is a game from Lapland which may have originated as far back as the fourteenth century. It is played by two opponents, each of whom has different strengths and different objectives. As in many good games, its rules and moves are quite simple and easy to learn, but the strategy can become more complex as one learns to play it.

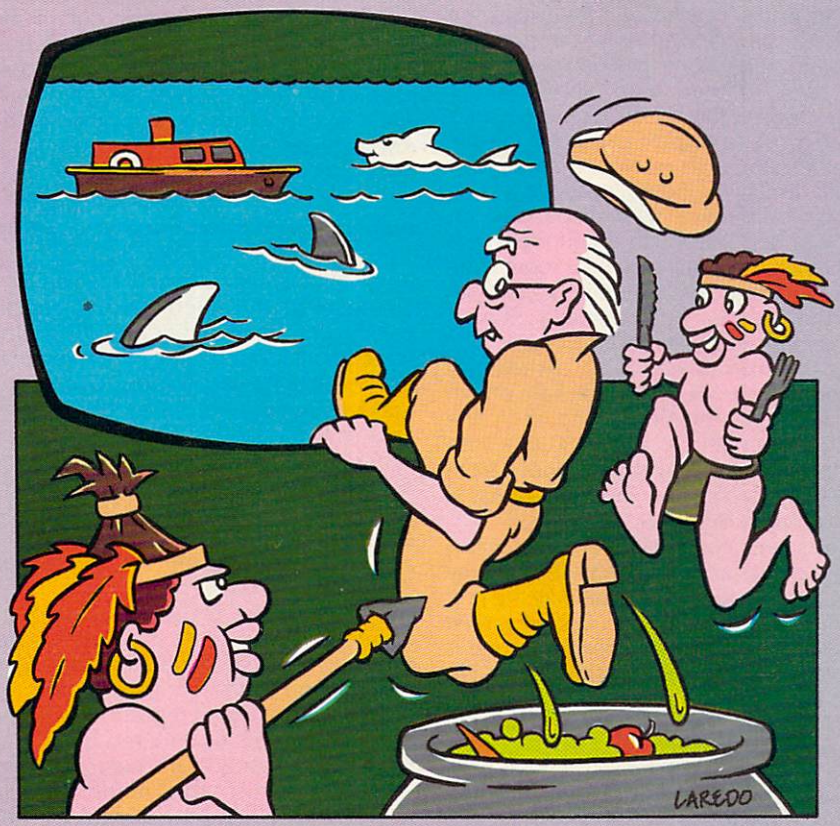
Tablut is played on a 9×9 board. The pieces all move in the same manner as rooks in chess: any distance either vertically or horizontally, but never diagonally. All pieces are confined to this vertical or horizontal movement, even the king. However, unlike the rook, a piece cannot capture by entering an enemy square.

One player has a king, who begins on the center square. Only the king is allowed to occupy or move through that square. This player also has eight pieces to guard the king. His object is to get the king to an end space of any row or column. As soon as the king reaches the edge of the board, the game is over and that player wins. The second player has sixteen pieces, which start the game near the edges of the board. His object is to capture the king. If he succeeds, the game is immediately over and the second player wins.



All pieces are subject to capture. To take any piece (except the king), a player must move one of his pieces adjacent to the enemy piece. If there is another friendly piece on the opposite side of the enemy (in the same row or column), then the enemy is removed from the board. This is called capture by "containment." The king cannot capture a piece. If he moves adjacent to an enemy, he has no effect. The king can be captured only if surrounded by four enemy pieces, or by three enemy pieces and the center square. It should be noted that a piece can move into a space between two enemy pieces or can move through such a space without being captured. The king, however, cannot make a move into a surrounded square next to the center since he would then be captured.

On each player's turn, the program first checks to see that the proper input has been given. It then checks the legality of the move and the status of the king, and then tests for captured pieces. Then it updates the board by moving pieces and removing captured ones. If there is a winner at the end of any move, the program proceeds to the conclusion, announces the winner, and asks if the player would like another game.



CANNIBALS

by Carl Carrozza

I had two reasons for naming this game *Cannibals*. The first will be obvious when you see the scenario. You play Dr. Livingston exploring a remote island to study the inhabitants' unique culture. You find out that unfortunately, their latest fad is *le haute cuisine homo sapiens*. And you are the only one on the island who is not invited to dinner.

My second reason for choosing the name *Cannibals* has to do with my design method. Inventors and mechanics are notorious for *cannibalizing*—taking parts from one machine to build another. I cannibalized shamelessly in creating this game. I borrowed the logic that enables the cannibals to "sense" your position and give chase from David Bojekian, one of my seventh grade students. The graphics were adapted from other games, and you might recognize the ship and man-eating sharks from *Counting Lesson (99'er HCM*, Vol. 1, No. 4). I also used some tips from "Fun and Games" (in the same issue of *99'er*) for programming the joystick/arrow keys option.

The object, of course, is to get Dr. Livingston to his ship before he is caught and slow-cooked by the natives to the strains of "Smoke Gets in Your Eyes." If the good doctor decides to swim for the ship, the man-eating sharks will devour him with a quick gulp to the tune of "Mack the Knife." In the lower levels of play, swamps with pits of quicksand appear on the screen. Dr. Livingston may not enter these areas; fortunately, the natives can't either and must find their way around them in order to get to the doctor. Dr. Livingston's advantage is that he can wrap around the screen to avoid the natives; the natives can't wrap around in pursuit of him. Each time the doctor reaches the ship he will be safe. But it's hard to make an adventurer like Dr. Livingston stay put. He keeps returning to explore the jungle. Each time the doctor reaches the boat, the level of difficulty increases, and there aren't as many swamps to slow the natives down. You will have to be very cunning by the time you reach the tenth level, where there are no swamps or obstacles of any kind to keep the natives from coming straight for you.

TI

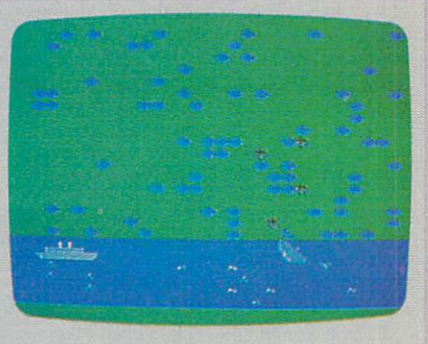
This version of *Cannibals* is written in TI BASIC but will also run in Extended BASIC. The game plays the same in either BASIC except that Extended BASIC is a little faster. You can use either joysticks or the four arrow keys (ESDX) to move Dr. Livingston. The fact that Dr. Livingston can run faster than the natives (thanks to the adrenal gland) coupled with his ability to wrap around the screen may be all that keep the doctor out of the bellies of the natives. To get him in the boat, you need only move the doctor to the bank opposite the boat. The boat is four characters long, and any part of it can be used for boarding. Because the program is written in TI BASIC, it does not use sprites to move the doctor and the natives. Instead they are moved whole characters at a time. Each character is redefined and then located on the screen with the HCHAR command.

Continued on p. 63

C64

When we program a game that requires a player to move a character on the video screen, we like to give the player the option of using either joysticks or the keyboard. Joysticks are usually preferred because more indications may be monitored simultaneously, whereas only one key at a time may be read from the keyboard. Unfortunately, not everyone has a joystick, so we will provide a dual method.

In the C-64 version of *Cannibals* we use the subroutine from lines 890 through 1010 to detect a direction input from either the joystick or the keyboard.



Let's look at each line in some detail.

Line 890 initializes the *x* and *y* coordinate variables to zero.

Line 900 works in conjunction with the input statement of line 450. If you're not using joysticks, the program skips over the next five lines (those that check the joystick) and checks only the keyboard.

Continued on p. 61

Tablut . . . from p. 58

The Apple version of *Tablut* uses the HPLOT command to draw the row and column values along the edge of the playing area, rather than showing the characters on the screen. This is because it is difficult to display characters on the screen in the high resolution mode. We could display the numbers and letters on the screen by defining a shape table for the numbers 1 through 9 and the letters A through I, but that would be more complicated. Instead of the numbers 1 through 9, the Apple version uses Roman numerals, which are easier to draw. The Roman numerals lend a little authenticity to the game as well. The labels are drawn by a long subroutine, beginning in line 1220 and ending in line 1500. Because the pieces on the board are drawn from the shape table, their positioning has a resolution of 280x160.

The screen coordinates for the positions of the pieces are in an array so that they can be indexed quickly. This way the program doesn't need to calculate the screen position of the shapes every time one is moved, which speeds up the game a bit. The data for this array is located in line 1650. Only two shapes are defined in the shape tables in lines 1700 to 1830. The king is represented by a crown-shaped figure. Both of the players share the same shape, and are distinguished by their colors. This is done initially when the screen is set up in lines 340 to 390. During play, lines 780 through 800 make sure the right shape and color are placed on the screen.

Continued on p. 118

The TI Extended BASIC version of *Tablut* is simplified by the use of the HCHAR command to redefine characters and place them on the screen. The program is made even simpler by an extremely short control loop that is used for both players. This two-line control loop, which consists of only GOSUBs, checks the game status with the variable KING. If KING is equal to 1, then the king has made it safely to the border; if KING is equal to 2, then the king has been captured.

Two subroutines (at lines 1130 and 1140) make it simple to enter the coordinates. The first routine allows entry of only the letters A through I and gives a little beep when the correct response has been entered. The second routine allows entry of only the numbers 1 through 9. At no time do you need to hit [ENTER]. The error checking starts when you enter the first coordinate. If the first set of coordinates is not an occupied square, a warning flashes on the screen, and you are re-prompted for an input.

The initial screen display is created with DATA statements containing characters that represent the board. You will notice that the characters in the DATA statements are not the ones that have been redefined for the graphics. This is because it is difficult to print characters that do not appear on the keyboard. The loop in lines 440 to 460 takes care of this by plucking each character out of the DATA, reading it, and adding an offset to its ASCII value to generate the characters needed for the display. At the same time, line 450 checks for an even row. If the row is even, then the row number is placed to the right of the board.

Continued on p. 121

Have you ever wanted to display a character on the video screen against a background color different from the screen background color? If so, you may have run into frustration when you realized that the normal graphics mode on the C-64 assigns all off pixels in a character definition to the background color of the screen. The C-64 version of *Tablut* utilizes an important feature built into the powerful VIC II video chip that allows us to obtain an "Extended Background Color" mode.

There are five registers (or memory locations) that concern us. One is a control register that turns the Extended Background mode on and off, and the other four are color registers. Any one of sixteen colors (numbered 0-15) may be POKEd into any of the color registers.

When we use this mode we are limited to the first 64 characters in the character set. This is because the upper two bits of a value POKEd or PRINTed on the screen are used to select one of the four background color registers. The color assigned to the color register selected will be the background color of the character. The following example will make this clearer: If you POKE a 1 onto the screen, you would expect to see the letter A displayed, and indeed you would. The change becomes apparent when you POKE the number 65 onto the screen expecting to see the graphic symbol that looks like a spade, but instead you get the letter A again, this time with a different background color. Congratulations! You have just used Extended Background Color mode.

The following table illustrates the association of character codes to background color registers:

Character Codes	Background Color Register Address
0-63	53281
64-127	53282
128-191	53283
192-255	53284

Extended Background Color mode is turned on by the following BASIC command:

POKE 53265,PEEK(53265)OR 64

Here is the BASIC command to turn the Extended Background Color mode off:

POKE 53265,PEEK(53265)AND 191

Extended Background Color mode is implemented in lines 320-330 of *Tablut*.

Line 320 assigns colors to each of the background color registers, and line 330 enables the Background Color mode.

Lines 240-310 insert programmable character definitions within the first 64 characters. We replace twelve characters, beginning with the exclamation point(!) and ending with the comma (,), but we can still generate all the words and numbers required in the program.

Continued on p. 151

The IBM PCjr version of *Tablut* uses the DRAW command to display the players' pieces. Both sides use pieces of the same shape, except for the king, which is crown-shaped. In this version, the king and his men are green, and the intruding pieces are red.

The shapes for the pieces are defined by string variables in line 190, and the king is defined in line 200. The strings contain a list of drawing commands that tell the computer where to draw. For example, the first part of the king's crown looks like this:

BR2BD2DRDRG

and would read like this:

Blank Right two, Blank Down two, Down, Right, Down, Right, Down and Left.

This part of the code goes right 2 pixels, then down 2 pixels without drawing anything; then down, right, down, right, and diagonally down and left, drawing in whichever color the system was told to use. The color is set with each DRAW command to give the program more control and flexibility. To make the color green, you would use a command string like this:

"C1;XK$;"

The **C1** sets the color to color #1 of the palette being used. XK$ tells the computer to eXecute **K$**. You can build drawing commands that include many different strings, each of which can be up to 255 characters long.

To save time calculating the board positions on the screen, the *x* and *y* screen coordinates for each of the board positions go into an array called TP( ). To find the *x* coordinate for the third column of squares on the board, you would use TP(0,3). To find the *y* coordinate for the 6th row of squares you would use TP(1,6).

Lines 950, 1070, and 1100 use the PLAY command to add sound effects. The PLAY command uses its own set of special commands to create music. The letters A through G represent the notes. The # sign denotes a sharp. The letter T followed by a number sets the tempo of the music, and L followed by a number sets the length of the notes that follow. The letter O followed by a number sets the octave. There are many other commands that can be used with the PLAY command to make it a very versatile instrument.

Continued on p. 152

Cannibals . . . from p. 59

Line 910 PEEKs the byte at joystick control port #2 and stores it in an integer variable (XT%).

Line 920 returns a value corresponding to a horizontal (X motion) position of the joystick. The following table lists the possible values for X1:

−1 = Left
1 = Right
0 = No movement

Line 930 checks to see if there was a horizontal input, and if so, the routine will skip checking for input on the vertical axis. This is because no diagonal movement is allowed Dr. Livingston when he is chased by the cannibals.

Line 940 checks for movement on the vertical (or *y*) axis. The following table lists the possible values for Y1:

−1 = Up
1 = Down
0 = No movement

Line 960 begins the routine that monitors the keyboard for direction if the joystick is not used. Address 197 will always contain a code unique to the key being pressed. If the value is 64, then no key is being pressed.

Line 970 checks whether the S key's was pressed. If it was, then X1 = −1.

Line 980 checks whether the D key's was pressed. If it was, then X1 = 1.

Line 990 checks whether the E key's was pressed. If it was, then Y1 = −1.

Line 1000 checks whether the X key's was pressed. If it was, then Y1 = 1.

If you would like to include a check on the joystick to see if the fire button is being pressed, you may use the following code:

FB = 1−SGN(XT% AND 16)

FB will be equal to 1 if the fire button is being pressed and will equal 0 if it is not.

Continued on p. 71

The Apple version of *Cannibals* uses the hi-resolution (hi-res) screen for its displays while leaving the four-line text window at the bottom of the screen to show the score. As is often the case, the program is large enough that if it were loaded normally (starting at $801) and RUN, the large array used to keep track of the screen locations of the characters would overwrite the hi-res screen and destroy the display. Line 170 solves this problem by PEEKing the pointer that points to the start of BASIC (locations 103 and 104) and comparing it to the location immediately above the hi-res screen. If the values in line 170 are not equal, then the new values are POKEd in, and the program is reloaded and RUN again. This time the hi-res screen is safe from being overwritten, and the program operates properly. (See our Tech Note for the Apple, "Protecting the Hi-Res Screen.")

Continued on p. 62

Cannibals . . . from p. 61

One tricky aspect of translating *Cannibals* to the Apple is getting the various music themes to play at the right times without excessive code. Because there is no sound generating chip on the Apple, all music is produced by "plucking" the speaker with the machine language routine POKEd in at the beginning of the program by the subroutine in lines 1700-1760. This routine is then CALLed each time one note is played by line 2000 after values are assigned to the two variables D (Duration) and P (Pitch). Because each melody is at least 16 notes long, and there are 4 different melodies, using individual assignment (LET) statements for each variable would be wasteful of memory and more difficult to code.

The TI-99/4A uses a FOR-NEXT loop coupled with READ and DATA statements to accomplish this task. Applesoft BASIC includes all of these features; however, the key to the TI approach is the RESTORE statement that lets you set the *pointer* to the DATA statements from BASIC. For example, the line

```
RESTORE 2000
```

in a TI-BASIC program means that the next READ statement would start READing from the first DATA statement in the program in line 2000 or greater. Applesoft BASIC, on the other hand, can only RESTORE to the first DATA statement in the program.

In order to use the READ-DATA method from Applesoft, we wrote a machine language program that we could CALL whenever we wished to RESTORE to a particular line number. It resets the DATA pointers used by Applesoft so that the next READ statement uses the correct DATA statement. This routine is POKEd in at lines 190-210 to address locations 36096 through 36231. Notice that line 190 also resets HIMEM to protect the routine from being overwritten by STRING variables. (String variables are stored starting at HIMEM and working downward in memory; see the *Applesoft BASIC Programmer's Reference Manual* for details.)

Whenever we need to RESTORE to a particular line number in the program, we branch to the linking Applesoft subroutine in lines 2020-2060, passing the line number as a parameter in the variable LN. This linking subroutine uses LN to calculate the values needed by the machine language routine to find the line number, POKEs them into memory, then CALLs the routine (lines 2030-2040). This machine language routine (which RESTOREs by line number) not only resets the proper pointers, but also checks to be sure that the line number in LN exists in the program and that it is indeed a DATA statement. If either of these conditions is not satisfied, then a flag is set by the machine language routine. The Applesoft section then checks the flag (identified by the variable ST) and halts the program, displaying an error message to identify the incorrect LN value.

If you wish to use this routine in your own programs, you should be aware that if you use the RENUMBER utility to resequence your program, the value of LN will *not* be changed. You must update the values of LN after you have RENUMBERed. If you don't do this, the error checking routine will determine an incorrect LN value. It is a good idea before RENUMBERing to note where each GOSUB to the linking program is, so that the LN values can be updated.

Continued on p. 78

The TI-99/4A version of *Cannibals* is a studied, measured game, almost like chess, with four cannibals jockeying around the bogs and advancing slowly. The PCjr version, on the other hand, is more hell-bent-for-leather, with cannibals streaming after you from every direction.

Some home computers (and the TI-99/4A is a good example) achieve remarkably good graphics speeds under certain circumstances by using a hardware chip, usually called a "sprite" chip, to handle the drawing and movement of shapes. A "dedicated" chip like this will be limited in what it can do, however, and it's somewhat isolated from the busy activity of the rest of the program. The sprite chip can't help much in a game like *Cannibals*, which involves a lot of

logic and coincidence checking when an object moves, so the computer is thrown back on the resources of its central processing unit, where these operations are carried out. Thus IBM, which has by far the fastest central processor in the home market, has the quickest cannibals.

In many applications, speed is all-important, but *Cannibals* seems to work well as either a deliberately paced strategy game or a frantic chase.

Some Notes on the Game

You'll be queried at the start about whether you want the program to play through the console or the monitor. The console has a tiny speaker in it, so use your monitor's sound unless it has no speaker, as is the case with the IBM Personal Computer Color Display.

You'll also be asked to choose between using the keyboard arrow keys or a joystick. If you're using a joystick, you should know that some joysticks will return different values when read by BASIC's STICK command. To cope with this, the program reads STICK(0) and STICK(1) at an early stage, before the joystick is in play, and uses these values as base constants for the rest of the program. This program was developed using a WICO Computer Command analog joystick and tested with an IBM stick, so you should be all right if you have one of these.

STICK(0) returns the value of the *x* coordinate of the stick (used to move back and forth), and STICK(1) returns the *y* coordinate (to move up and down). If your joystick decreases STICK(0) when you move to the right instead of increasing it as the WICO stick does (and increases instead of decreases when you move left, etc.), then you'll find your controls reversed. There may be some joysticks on the market that depart even further from the WICO stick, so you may need to modify your program. Don't feel shy about jumping in and making changes. . . Just remember to save the original version first.

The WICO is an analog stick, which means that the value it outputs will change smoothly and constantly as you move the stick one way or another. (It would be ideal for smoothly varying the speed of an object in a program.) A digital joystick, by contrast, outputs only discrete values. At the center position it might output 0; moved to the right, it might output +1 or perhaps +5. Because it was written for an analog stick, the code in this program looks for a range of values when it is trying to determine if the stick is in the center position. It reads the WICO's position at the beginning and knows what it ought to be (say, 56), but later, from jostling or handling or just being there, the WICO might be returning 57 or 55 or 54. A digital stick, however, will always return its discrete value... In other words, analog is wiggly and changeable.

One final note: Make sure that you aren't leaning on your stick when the program reads it at the very beginning, or you'll be able to run only one way.

Continued on p. 69

Cannibals. . . from p. 59

CANNIBALS (TI-99/4A)
Explanation of the Program

Line Nos.	
100-190	Program header.
200-370	Initialization and input option for joysticks or keyboard.
380-550	Display the title screen.
560-590	Time delay loop and clear the screen.
600-720	Define graphics and color for characters.
730-760	Display ocean.
770-820	Display ship.
830	Clear the jungle area of the screen.
840-870	Create swamps.
880-980	Place the cannibals and the doctor on the screen.
990-1020	Branch to read the keyboard or the joystick.
1030-1200	Move the cannibals.
1210-1290	Doctor makes it to safety on the boat.
1300-1330	Check to see if the doctor made it to the boat or the shark.
1340-1620	Shark makes a meal of the good doctor.
1630-1770	Cannibals place Dr. Livingston on the menu when they catch him.
1780-1820	Subroutine to wait for a key response.
1830-1920	End of the game messages and restart the game.
1930-2070	Display the score.
2080-2200	Subroutine to scan the keyboard for Dr. Livingston's movement.
2210-2370	Move Dr. Livingston.
2380-2410	Subroutine to scan the joysticks.
2420-2450	DATA containing graphics pattern information.
2460	DATA containing character set color information.

TI-99/4A

```
100 REM  *********************
110 REM  *        CANNIBALS    *
120 REM  *********************
130 REM  BY CARL CARROZZA
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM
170 REM  TI BASIC, OR
180 REM  TI EXTENDED BASIC
190 REM
200 GOSUB 380
210 DIM D(6,2)
220 CALL CLEAR
230 PRINT "TYPE 1- FOR JOYSTICK.":::
    :"TYPE 2- FOR ARROW KEYS.":::::
240 CALL KEY(0,K,S)
250 IF S=0 THEN 240
260 IF K=49 THEN 290
270 IF K=50 THEN 340
280 GOTO 240
290 AA=1
300 CALL CLEAR
310 CALL SCREEN(7)
320 PRINT "PLACE ALPHA LOCK UP."
330 GOTO 350
340 PRINT "PRESS ALPHA LOCK DOWN."
350 FOR DEL=1 TO 200
360 NEXT DEL
370 GOTO 560
380 CALL CLEAR
390 PRINT "**********************"
400 PRINT "*          *"
410 PRINT "*  CANNIBALS  *"
420 PRINT "*          *"
430 PRINT "**********************"
440 RESTORE 490
450 FOR NOTE=1 TO 29
460 READ A,B
470 PRINT
480 CALL SOUND(200*A,B/2.7,0)
490 DATA 1,1109,1,1661,1,1397,.5,1245,.
    5,1397,2,1109,2,831
500 DATA 1,1109,1,1661,1,1397,.5,1245,.
    5,1397,2,1109,2,831,1,44000
510 DATA 1,1109,1,1397,.5,1245,.5,1109,
    2,1245,2,1661,1,44000
520 DATA 1,1109,1,1397,.5,1245,.5,1109,
    2,1245,2,831,1,44000
530 NEXT NOTE
540 RANDOMIZE
550 RETURN
560 FOR T=1 TO 100
570 NEXT T
580 LI=0
590 CALL CLEAR
600 REM DEFINE GRAPHICS AND COLOR
610 RESTORE 2320
620 FOR X=1 TO 19
630 READ A,A$
640 CALL CHAR(A,A$)
650 NEXT X
660 FOR X=9 TO 14
670 READ A,B
680 CALL COLOR(X,A,B)
690 NEXT X
700 CALL COLOR(1,6,4)
710 CALL CLEAR
720 CALL SCREEN(4)
730 CALL HCHAR(19,1,128,192)
740 FOR X=1 TO 20
750 CALL HCHAR(RND*5+19,RND*28+3,129)
760 NEXT X
770 REM LOCATE SHIP
780 M=INT(24*RND+3)
790 CALL HCHAR(20,M,96)
800 CALL HCHAR(20,M+1,97,3)
810 CALL HCHAR(20,M+4,98)
820 CALL HCHAR(19,M+2,99)
830 GOSUB 1840
840 CALL HCHAR(1,1,32,576)
850 FOR X=1 TO INT(ABS(10-SAFE))*10
860 CALL HCHAR(INT(RND*18)+1,INT(RND*28
    )+3,33)
870 NEXT X
880 FOR R=1 TO 6
```

TI-99/4A

```
280 GOTO 240
290 AA=1
300 CALL CLEAR
310 CALL SCREEN(7)
320 PRINT "PLACE ALPHA LOCK UP."
330 GOTO 350
340 PRINT "PRESS ALPHA LOCK DOWN."
350 FOR DEL=1 TO 200
360 NEXT DEL
370 GOTO 560
380 CALL CLEAR
390 PRINT "**********************"
400 PRINT "*          *"
410 PRINT "*  CANNIBALS  *"
420 PRINT "*          *"
430 PRINT "**********************"
440 RESTORE 490
450 FOR NOTE=1 TO 29
460 READ A,B
470 PRINT
480 CALL SOUND(200*A,B/2.7,0)
490 DATA 1,1109,1,1661,1,1397,.5,1245,.
    5,1397,2,1109,2,831
500 DATA 1,1109,1,1661,1,1397,.5,1245,.
    5,1397,2,1109,2,831,1,44000
510 DATA 1,1109,1,1397,.5,1245,.5,1109,
    2,1245,2,1661,1,44000
520 DATA 1,1109,1,1397,.5,1245,.5,1109,
    2,1245,2,831,1,44000
530 NEXT NOTE
540 RANDOMIZE
550 RETURN
560 FOR T=1 TO 100
570 NEXT T
580 LI=0
590 CALL CLEAR
600 REM DEFINE GRAPHICS AND COLOR
610 RESTORE 2320
620 FOR X=1 TO 19
630 READ A,A$
640 CALL CHAR(A,A$)
650 NEXT X
660 FOR X=9 TO 14
670 READ A,B
680 CALL COLOR(X,A,B)
690 NEXT X
700 CALL COLOR(1,6,4)
710 CALL CLEAR
720 CALL SCREEN(4)
730 CALL HCHAR(19,1,128,192)
740 FOR X=1 TO 20
750 CALL HCHAR(RND*5+19,RND*28+3,129)
760 NEXT X
770 REM LOCATE SHIP
780 M=INT(24*RND+3)
790 CALL HCHAR(20,M,96)
800 CALL HCHAR(20,M+1,97,3)
810 CALL HCHAR(20,M+4,98)
820 CALL HCHAR(19,M+2,99)
830 GOSUB 1840
840 CALL HCHAR(1,1,32,576)
850 FOR X=1 TO INT(ABS(10-SAFE))*10
860 CALL HCHAR(INT(RND*18)+1,INT(RND*28
    )+3,33)
870 NEXT X
880 FOR R=1 TO 6
```

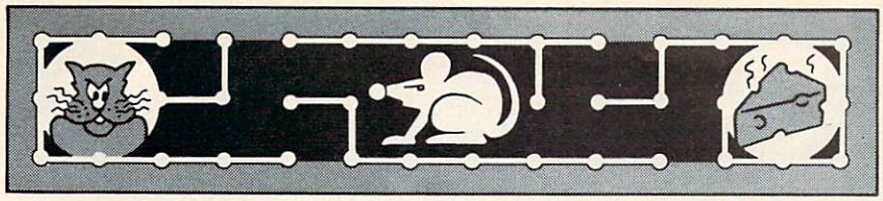
Continued on p. 64

Cannibals. . . from p. 63

TI-99/4A

```
 890 D(R,1)=INT(17*RND+1)
 900 D(R,2)=INT(28*RND+3)
 910 CALL GCHAR(D(R,1),D(R,2),CH)
 920 IF CH<>32 THEN 890
 930 IF R=5 THEN 980
 940 IF R=6 THEN 970
 950 CALL HCHAR(D(R,1),D(R,2),112)
 960 GOTO 980
 970 CALL HCHAR(D(R,1),D(R,2),104)
 980 NEXT R
 990 IF AA=1 THEN 1020
1000 GOSUB 1980
1010 GOTO 1030
1020 GOSUB 2280
1030 ZZ=ABS(ZZ-1)
1040 IF ZZ=1 THEN 990
1050 FOR X=1 TO 4
1060 MCX=(D(X,1)<D(6,1))-(D(X,1)>D(6,1))
1070 MCY=(D(X,2)<D(6,2))-(D(X,2)>D(6,2))
1080 IF (D(X,1)-MCX>18)+(D(X,1)-MCX<1)+(
     D(X,2)-MCY>31)+(D(X,2)-MCY<2)THEN 1
     190
1090 CALL GCHAR(D(X,1)-MCX,D(X,2)-MCY,CH
     )
1100 IF (CH=104)+(CH=32)THEN 1140
1110 MCX=INT(RND*3)-1
1120 MCY=INT(RND*3)-1
1130 GOTO 1080
1140 D(X,1)=D(X,1)-MCX
1150 D(X,2)=D(X,2)-MCY
1160 IF (D(X,2)=D(6,2))*(D(X,1)=D(6,1))T
     HEN 1630
1170 CALL HCHAR(D(X,1)+MCX,D(X,2)+MCY,32
     )
1180 CALL HCHAR(D(X,1),D(X,2),112)
1190 NEXT X
1200 GOTO 990
1210 REM MADE IT TO BOAT
1220 RESTORE 1270
1230 FOR I=1 TO 11
1240 READ P,N
1250 CALL SOUND(300*P,N,0)
1260 NEXT I
1270 DATA 1,932,1,932,2,1047,1,1047,1,10
     47,2,1047,1,1397,1,1245,1,1175,2,93
     2,1,44000
1280 SAFE=SAFE+1
1290 GOTO 1610
1300 REM CHECK FOR WIN
1310 FOR WIN=0 TO 4
1320 IF D(6,2)=WIN+M THEN 1220
1330 NEXT WIN
1340 REM SHARK EAT
1350 Z=D(6,2)+(D(6,2)=30)+(D(6,2)=31)*2
1360 CALL COLOR(9,16,5)
1370 CALL HCHAR(19,Z,137)
1380 CALL HCHAR(19,Z+1,139)
1390 CALL HCHAR(20,Z,138)
1400 CALL HCHAR(20,Z+1,140)
1410 CALL HCHAR(21,Z+3,141)
1420 FOR J=1 TO 6
1430 CALL HCHAR(20,Z,136)
1440 CALL SOUND(250,-5,0)
1450 CALL HCHAR(20,Z,138)
1460 CALL SOUND(250,44000,30)
1470 NEXT J
1480 RESTORE 1520
1490 FOR SHA=1 TO 18
1500 READ A,B
1510 CALL SOUND(300*A,B/2,0)
1520 DATA 1,1109,1,1319,1,1480,1,44000,3
     ,1480
1530 DATA 1,1109,1,1319,1,1480,1,44000,3
     ,1480
1540 DATA 1,1109,1,1319,1,1480,1,44000,3
     ,1480
1550 DATA 1,1661,1,1661,2,1480
1560 NEXT SHA
1570 DEAD=DEAD+1
1580 CALL HCHAR(21,Z+3,128,4)
1590 CALL HCHAR(20,Z+1,128,2)
1600 CALL HCHAR(19,Z+1,128,2)
1610 CALL HCHAR(19,1,128,64)
1620 GOTO 780
1630 CALL HCHAR(D(6,1),D(6,2),120)
1640 CALL HCHAR(D(6,1),D(6,2)+1,121)
1650 CALL HCHAR(D(6,1)+1,D(6,2),122)
1660 CALL HCHAR(D(6,1)+1,D(6,2)+1,123)
1670 RESTORE 1720
1680 FOR SMO=1 TO 11
1690 READ A,B
1700 CALL SOUND(400*A,B/2,0)
1710 NEXT SMO
1720 DATA 2,1109,1,1047,1,1245,1,1109,1,
     932,2,1480
1730 DATA 1,1397,1,1661,1,1480,1,1245,2,
     2218
1740 DEAD=DEAD+1
1750 CALL HCHAR(D(6,1),D(6,2),32,2)
1760 CALL HCHAR(D(6,1)+1,D(6,2),32,2)
1770 GOTO 1610
1780 REM SUB TO CONT.
1790 CALL KEY(0,K,S)
1800 IF S=0 THEN 1790
1810 CALL CLEAR
```

Continued on p. 69



MOUSER

A Review by Judy Sanoian
HCM Staff

Name:	**Mouser**
Program Type:	**Arcade Game**
Machine:	**IBM PCjr**
Distributor:	**IBM Software**
	P.O. Box 1328-S
	Boca Raton, FL 33432
Price:	**$35.00, cartridge**

	poor	fair	good	excellent
Performance			▬	
Engrossment				▬
Documentation			▬	

Remember the three blind mice? They taunted the farmer's wife until she grabbed the carving knife and hacked off their little tails. In *Mouser*, the PCjr's sound and graphics capabilities have been tapped to transform that grisly tale into a colorful and lively arcade-style game.

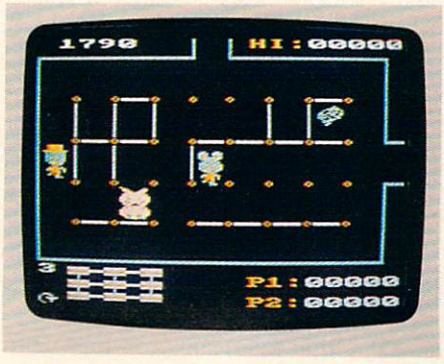
The scenario is a simple one: You play Farmer Wheatbread who moves through a maze of swinging walls, setting them up in boxes to trap a pack of ravenous mice. This is more difficult than it sounds. In the heat of the game you sometimes forget which way a wall will swing—determined by pressing the space bar or the joystick's fire button—when you push it. If it swings the wrong way, you may find yourself face to face with a *deadly* mouse (either they're rabid or Farmer Wheatbread is extremely fainthearted). Also there are nine separate rooms (screens) full of the treacherous rodents, and there is a reproduction factor—the female mice multiply if they reach the cheese. There is also a "dark room" (due to Farmer Wheatbread's unpaid light bill) in which your peripheral vision is severely limited, and a ticking clock is constantly working against you.

Three Mouser Cheers

Three things will immediately jump out at you when you begin to play *Mouser*. First, its response is excellent. Farmer Wheatbread responds to your (joystick or keyboard) commands with speed and precision. But this doesn't, however, make the game any easier. At times it even seems to make it more difficult: When things get tense, your slightest involuntary movement can send your farmer off and running in a direction that you hadn't intended.

The second feature you will notice is *Mouser*'s excellent use of sound. Actually, it is probably the first thing you'll notice because *Mouser*'s sound effects are truly unique. The noise accompanying the farmer's movement resembles a twanging rubber band with lots of echo. There are also harmonious, electronic-sounding tunes when you swing a wall or enter a room and a dramatic noise similar to clapping (or shutting a coffin, perhaps) when either the mouse or your farmer dies. The sound may get irritating after a while—especially when you're losing—and some players may prefer to turn it way down or off.

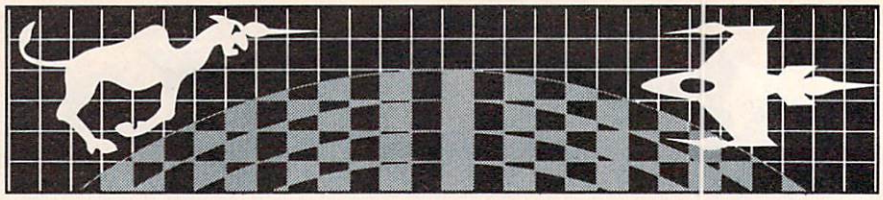
The third most striking aspect of *Mouser* is its graphics. The mice, cats, farmer—even the cheese—are detailed, clear and colorful. The PCjr can offer this superior color resolution because it is capable of producing one of 16 colors in each individual pixel on the screen when in medium resolution mode. This means you get more colors per character. But keep in mind that we were viewing them on the expensive IBM RGBI monitor. Viewing it on a regular TV screen, you lose some of the crisp, clear effect.



There is an especially nice use of color in the dark room of the house. When you enter it, all the colors change, and a circle of contrasting color (simulating the glow of a flashlight) surrounds Farmer Wheatbread as he hunts rats in the dark. The other eight rooms are all basically the same with a few more or less mice in each one. I would have enjoyed a bit more variety among the rooms—new predators, different obstacles or movements—to make entering each new screen more interesting.

While we're suggesting improvements, I would also recommend adding an optional "baby" level of difficulty for

Continued on p. 76



Attack of the Mutant Camels

A Review by Greg Roberts
HCM Staff

Name:	Attack of the Mutant Camels
Program Type:	Arcade
Machine:	VIC-20
Author:	Jeff Minter
Distributor:	HesWare Human Engineered Software 150 North Hill Drive Brisbane, CA 94005
Price:	$29.95

	poor	fair	good	excellent
Performance				
Engrossment				
Documentation				

The popular game *Gridrunner* has mutated into a new organism, taking on some bizarre and appealing traits in the process.

In *Attack of the Mutant Camels*, your monitor simulates a view from inside your space vehicle; the grid shows your position on the battlefield and that of the enemies. Use the joystick to move your fighter plane anywhere on the grid except the four top lines. You can hold down the fire button with no apparent loss of power. You're going to need lots of ammunition: The enemy droids come down in wave after wave. They attack in small groups or as a Space Invaders column that you must blow away before it reaches the bottom of the screen.

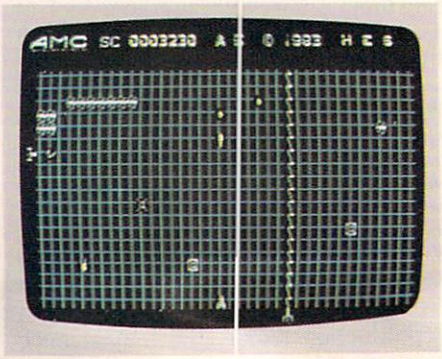
But there's more to worry about than columns of droids. Two laser zappers move around the periphery of the grid, firing at random. You must keep an eye on their positions at all times, making sure you don't get caught in their line of fire.

The camels? They first appear in Level 4 (of 20 levels), and they are best explained in the manufacturer's message to all prospective gridrunner pilots: "We also believe the droids are attempting some kind of bizarre psychological disorientation tactics—some pilots have reported sighting what appear to be camels running down the matrix." In the computer gaming world, that remark is not only plausible, but highly intuitive.

UFOs and Y-Beams

It's unfortunate that not all the features of *Mutant Camels* are explained so well. When you first run the game, it's hard to tell what's going on—and it won't help much to read the instructions. The leaflet does not mention half the images you see on your screen. You fire at every odd-shaped piece of garbage that comes floating past the grid. Some disintegrate, some don't. You're not sure who the enemy is. Tiny orange rectangles that look like breakfast cereal change shape when you hit them, but don't disappear. What does that mean? The leaflet talks about the dangers of "Y beams," but doesn't say what they are. Sometimes this game wipes you off the screen for no apparent reason, and you have no idea why it happened.

In an unprecedented humane gesture, the game does let you bypass levels 1-5. And, for this business, it's about time. The computer arcade is an arena where the tiniest slip-up can condemn you to square one. Computer adventure games let you pick up where you left off, so why shouldn't you be able to save your place in an arcade program? This game doesn't go that far either, but I appreciate the concession it does make.



Besides its razor-sharp, extremely fast graphics, *Mutant Camels* has the most outrageous sound effects of any home arcade game I've seen. The bangs and bleeps are unrelenting from your first touch of the fire button. It will delight games fanatics and will completely uncork everyone else unless the volume is set at zero.

Games enthusiasts who spot this bizarre title on the rack might expect something much different from other home computer games. Instead, they will find a game with strong roots in the Centipede family. Nevertheless, it is an expertly-programmed, fast-moving challenge that turns the VIC-20 into a mighty machine.

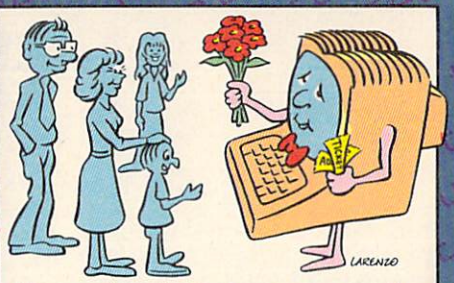
HCM

Volume 1, Number 2 $1.50

HOME COMPUTER™ DIGEST

News and Happenings in the Home Computer World

- Computer Peripherals: Small is Beautiful
- Techno War: Truth or Myth?
- Crystal Ball Market Gazing
- Big Mac Attack on Campus
- Summer Olympics Go Digital



More user friendly than ever. . . see page 3

CONTENTS

3 **Wooing the Home Computer Family** . . . . . *by Sharyn Lyon*
The family that computes together. . .

5 **Why Smaller Disk Drives?** . . . . . *by Roger Wood*
Just when you thought 5¼" was the industry standard!

8 **Preview of Summer's CES** . . . . . *by David Brader*
What's going to happen next?

12 **Computers Enable the Disabled** . . . . . *by Joan Killough-Miller*
New technology offers new independence.

16 **On the Road to Computer Literacy** . . . . . *Digest Staff*
Man's marvelous math machines.

24 **Celebrity Software and Software Celebrities** . . *by Erin O'Connor*
The famous and near-famous shine in the limelight.

DEPARTMENTS

2 On Line
6 Any Questions?
10 Software Trends
14 New Tech News
22 Industry Watch
26 Computers in Education
28 Novel Applications
30 Gameware Updates

Publisher/Editor-In-Chief
Gary M. Kaplan

Executive Editor
David G. Brader

Managing Editor
Walter Hego

Production Editor
Sharyn Lyon

Features Editor
Greg Roberts

Technical Editors
Roger Wood
Peter Bloch

Assistant Editors
Erin O'Connor
Joan Killough-Miller

Research Editor
Deborah Amity

Manuscript Editor
Dana Campbell

Production Manager
Norman Winney, Jr.

Design Coordinator
Eric Martin

Production and Design
Larenzo
Larry Fisk
Barbara Mickelson
Mohsen Zarrabi

Typesetting
Kathy Garcia
Julienne Laabs
Louisa Ada Hooven

Customer Relations
Denise McGuire
Tel. 503-341-1029

Bulk Distribution
Paula Hoiland
Tel. 503-341-1053

On Line

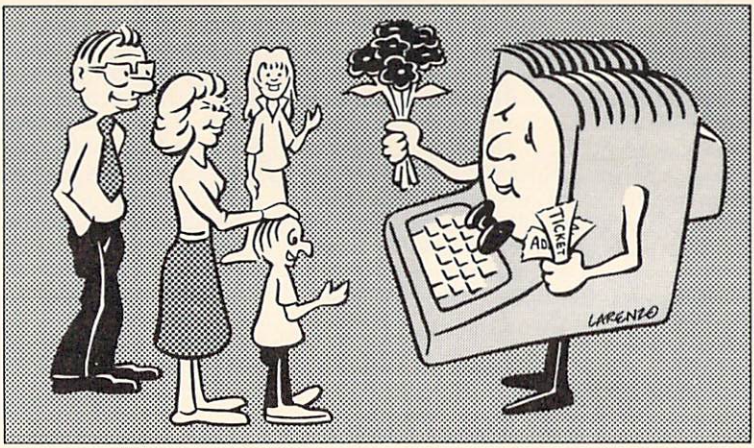
Have our science fiction fantasies come true? Is society speeding towards a head-on collision between computer technology and the needs of people? This month, *Home Computer Digest* explores the dynamic interface between those marvelous machines and their all-too-human users. Our features focus on a diverse group who are making computers a part of their lives.

Wooing the Home Computer Family explores what's behind the hardware and software industry's heightened pitch to Mom, Dad, and the kids. *Computers Enable the Disabled* reports on what electronic technology can do for people with special needs. Our look at *Celebrity Software and Software Celebrities* turns the spotlight on another group of special people who are making industry news. And the latest installment of *On the Road to Computer Literacy* brings you one step closer to a close rapport with the new machines in your future.

What's in store for us? *Why Smaller Disk Drives?* ventures some predictions on the emerging industry standards, and our *Preview of Summer's Consumer Electronics Show* takes a peek into the future of the home computer industry. And, as always, our regular features continue to bring you the last word on the present state of the home computing world.

HOME COMPUTER DIGEST is published by Emerald Valley Publishing Co., P.O. Box 5537, Eugene, OR 97405.

Wooing the Home Computer Family



The time is now! Soon we'll all be needing a home computer. And just in time—computer makers from Apple to Zenith are providing us with a hardware horn of plenty. New home computers are coming out of the woodwork, and each one hits us with a new promotion angle. Ads using price slashing, rebates, guilt-trips, and even appeals to Motherhood and Apple Pie parade before us. The Home Computing Family is definitely being courted, and each suitor claims to be smarter, better-looking, more modern, and more compatible with our interests than its predecessors. All this attention from industry honchos—who hope that theirs will be the most at-home home computer—is having unexpected results that will benefit home users as much as the industry itself.

You Don't Bring Me Flowers

Although computers are not being delivered to our doors by FTD, they are turning up on the backs of cereal boxes, on discount store sales flyers, and even in mass market magazine ads and TV commercials. Look at the way the PCjr was recently introduced by IBM. The first ad that ran in a general audience publication read like a birth announcement, complete with an old-fashioned baby carriage rocked by Big Daddy IBM (the Charlie Chaplin character). Below this compelling portrait (designed to grab the attention of the elusive female consumer?) was the clincher phrase, "Announcing A Proud Addition To Your Family." The magazine ad, like its television counterpart, takes you through Junior's life cycle, linking the growth of IBM's junior with the family's own flesh and blood Junior. Under the heading "GROWING UP WITH JUNIOR," the ad tells how easily you can expand the PCjr as your child's need for "a powerful tool for home, school or college" expands.

Seeing Is Believing?

With carefully selected props— overstuffed chairs, foot stools, and assorted cozy-looking antiques—IBM's TV ads visually communicate how comfortably junior fits into any home environment. These homey TV ads first aired during Super Bowl XVIII and have continued to fill prime-time slots surrounded by family-oriented specials, network news, sports, and morning talk/news shows. *Advertising Age* claims that IBM will spend about $40 million acquainting the American family with the PCjr. Why in-

vest so much time and money in one computer? Afraid that their "big business" image will turn away PCjr purchasers, IBM hopes, by the end of the third quarter, to establish the equation: Modern Family + Computer = You + PCjr.

It Followed Her To School One Day

Along with these family-oriented magazine/TV ad campaigns, IBM and Apple are entering the home through the back door—the schools. Plans to donate 2,000 PCs and PCjrs to private and public elementary and secondary schools in more than 25 US cities will increase consumer awareness of IBM in the private sector, an important market that has been long-dominated by Apple. IBM seems to be counting on the marketing axiom that each computer placed in our schools will ultimately be responsible for selling four or five additional machines to home users.

Both IBM and Apple have commitments from universities across the country. Apple recently formed the Apple University Consortium [see "Computers In Education" in this issue.—Ed.], giving discounted equipment to member institutions in return for software research and development. IBM has fielded a new strategy to counteract Apple's consortium campaign. Many of their computers are already in place at the university level. In fact, some business and technical schools are now requiring all students to have an IBM PC. Hardware retailers have mixed reactions to the low prices. Dealers in college towns are angry with Apple for discounting its machines because none can match a 60% discount. Nevertheless, some dealers predict that the larger user base will spur software sales among students and their families.

Superior Software for Students

Not to be outdone by college software R&D, commercial educational software producers have gotten serious about creating higher quality programs. The Learning Company is field testing programs in schools and homes throughout northern California. Their testing includes observation of play, in-school interviews and questionnaires, and in-home interviews with children. The revised programs then undergo a second round of testing with different children. With such extensive pre-marketing trials it is no wonder that IBM turned to The Learning Company to develop the first software for the PCjr.

While educational programs make inroads, producers of personal productivity software are kicking themselves for not anticipating the needs of home computer users sooner. But much catch-up work is underway, and we should start seeing many more home-sized versions of the popular accounting packages that were originally designed for business use. It appears that the line separating home computer from business computer concerns is fading. The cross-over is underway on both sides, as home users demand more sophisticated utility packages and business demands educational software—like the PLATO courseware—for on-the-job training and re-training programs.

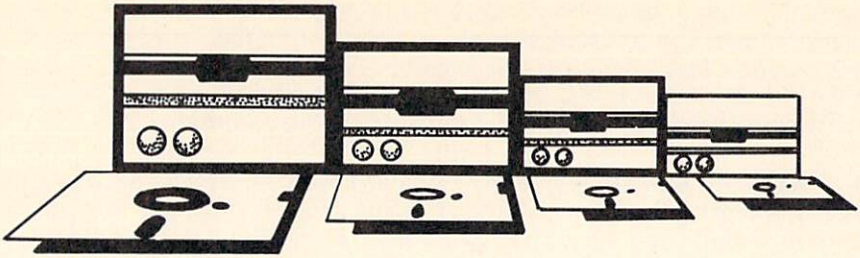
The Cycle Continues

The Home Computing Family shops for hardware and software to meet its many needs in an anxious marketplace. With weighty goals like a child's future in mind, the home computer consumer is a thorough collector of information. Hardware and software producers who are genuinely proud of their products are working to create a loyal and well-informed following.

Some manufacturers, like IBM, are trying to change their image of big business impersonality. Whether the motivation is genuine pride in their products or just profits and bigger market shares, the result is a more technologically receptive public. Ads that portray computers as friendly—even cuddly—inspire the consumer to think of the computer as more of an ally than an enemy. If the results of this careful courting of the home computing family are the demystifying of the computer, lower prices, higher quality software, better inter-industry communication, and easier access to technological information, then we have very little to lose—and much more to gain—from this new-found high-tech attention.

—Sharyn Lyon

Why Smaller Disk Drives?



In the past few years floppy disks have become the standard mass-storage medium in the computer industry. In the late 70's, the 8" floppy disk was the most common storage medium for mini- and micro-computers. But the 5 1/4" floppy quickly superseded the 8" as the principal storage device for microcomputers, both in business and in the home. The smaller physical size makes the 5 1/4" drive far more convenient for most applications, even though it holds less data than the 8". The 5 1/4" systems are capable of storing anywhere from 90K to 320K-bytes of data per disk, depending upon whether they support single-sided or double-sided disks, and whether data is stored on the disk using "single-density" or "double-density" encoding techniques. The 8" disk can store from 360K to more than one megabyte, depending upon the same considerations mentioned above.

More recently, "hard disk" drives (known as Winchester drives) have also come into their own. They are being used increasingly in situations where large amounts of data need to be regularly accessed. These drives are capable of storing 5 to 10 megabytes of data on a single 5 1/4" disk, but the disks are quite different from the 5 1/4" floppy. Generally, they are permanently installed in a computer system and are best suited to large data bases. Also, while ready-to-use floppy disk systems can be obtained for as little as $300, a hard disk system can cost $1500 or more.

Now, just as drives were becoming standardized, Apple's Macintosh comes equipped with a Sony 3 1/2" drive. Hewlett-Packard chose the same Sony disk for its HP-150 business computer, and many other manufacturers have introduced 3" and 3 1/4" drives as well. Why the change? Won't the smaller disks hold less data than the 5 1/4"? Is there really any advantage to the new smaller drives?

Smaller Drives Stack Up Well

Surprisingly, the smaller drives have quite a few advantages over their 5 1/4" big brothers. First, the smaller physical size of the drive provides a more compact package, thus allowing the computer itself to be more compact (witness both the HP-150 and the Macintosh). Another big advantage of the new, smaller disks is that they are not floppy. Instead, they are encased in a light plastic housing, making them far less susceptible to the damage that often befalls floppy disks. The smaller size and sturdier packaging also provide a more transportable medium for programs and data. The disk fits easily in a shirt pocket, or in a regular envelope for mailing (hand cancel, please). In addition, its small sliding metal shutter protects the storage medium when it's not in use, so you don't have to worry about an ill-placed fingerprint destroying this month's sales records, (as you might with the exposed medium of the larger disks). With all this

cont. on pg. 25

Any Questions?

The present global ignorance of computing may come, in part, from our natural aversion to asking simple questions—for fear of revealing only a shallow knowledge of vital topics.

Why not let someone else ask the questions while we sit back and benefit from the reply? That's the purpose of this column.

Q. What is an operating system?

A. An operating system is a program in the computer's memory that is responsible for controlling the basic functions of the system. A computer may use several kinds of operating systems, the most common of which is the Disk Operating System or DOS. It controls all interfaces with the disk drives, whether from programs or as user input. Another common operating system is known as the BIOS or Basic Input/Output Operating System. This one controls all I/O operations of the computer, including links to printers, modems, color graphics video circuitry, and other hardware.

Q. Why do some people bother with computer monitors when they can get the same picture on an ordinary television set?

A. The picture quality of a television set can't compare with that of a good monitor. This is, in part, because the TV must convert the video signals from Radio Frequency (RF) into video signals that represent the actual picture. With a monitor there is no need for a channel selector because it does not convert an RF signal. The signal supplied to the monitor is a direct video signal from the computer, with no messy frequency conversions necessary. Another performance advantage of the monitor is its ability to accept signals of a much broader bandwidth. In other words, the picture will have brighter and more definable colors and there will be less color bleed from one screen pixel to the next. The monitor doesn't offer non-computer entertainment, but the improved video image is easier on the eyes.

Q. I'm trying to decide which storage system to use—tape or diskette. Is the price of a disk operating system justified?

A. As the price of disk operating systems continues to drop, the lowly cassette player seems less and less attractive. Now that disk drives for some home machines are approaching the $200 mark, it is easy to justify that amount in time saved, not to mention tape costs. In order to be truly convenient, taped programs are best stored on individual cassettes, and that is a more expensive method than keeping multiple files on a diskette.

Q. Why buy more than one disk drive?

A. Most computer users can get by very well with one drive. However, there are many applications that require two drives, or operate much better with two. Many programs which handle data bases, such as word processors, or accounting software, need to have a program disk in one drive and the data disk in the other drive. Those programs that allow you to use one drive may have you swapping disks so often that you start wondering who's controlling whom. Dual drives are also very con-

venient for maintaining backup copies of an entire disk without having to copy the files individually.

Q. I have heard that the new remote keyboards used with the PCjr can only be used singly, not with several in one place, such as in a classroom. What is the problem, and what can be done about it?

A. When you place several keyboards in the same room, all using the infrared link to the main unit, you will probably run into an interference problem. This occurs when two people type on two different keyboards at the same time. The signals become mixed, and therefore unreadable at the computer. There are two solutions to this problem. If only one person were allowed to enter anything on a keyboard at any one time, then there would be no interference problem. But, there are definite logistical problems with this approach in a classroom. The second solution would be to use IBM's optional keyboard cable. The cable supplies a direct link to the main unit and turns the infrared transmitter off. The disadvantage here is that each keyboard will need a separate system unit. For group participation, the infrared keyboard will work quite nicely. It can be passed around from student to student as each one takes a try at the problems.

Q. Can I still get a disk drive for the TI-99/4A?

A. Several hardware makers are doing very well by continuing to support the two million or so 99/4A's now in use. Disk drives, 32K and 128K expansion cards, and other peripherals are described and reviewed in the pages of *Home Computer Magazine*.

Q. Would you recommend my buying a computer through a discount mail-order outfit?

A. If you are computing on a budget, you may find that the price variance between your neighborhood shop and a large mail-order house can mean the difference between owning the machine or going without. Mail order discount stores are usually reliable, in spite of the horror stories you may have heard about them. Remember, however, that your relationship with a discount house will usually last only as long as it takes them to cash your check and ship the box. You sacrifice the sort of dealer support you can get from a home-town shop—often crucial to a new computer owner. Then again, there are always users groups to help you learn to use the machine. In summary, there are good reasons for purchasing from either source. It just depends on your needs.

PREVIEW OF Summer's Consumer Electronics Show

Gazing into the crystal ball of last winter's Consumer Electronics Show reveals a clear vision of what to expect at this summer's product-introduction extravaganza.

Texas Instruments is on the sidelines. Commodore is changing course, IBM is gaining market share, and Apple is moving into the home. What's going to happen next? Will window technology become the predominant human/computer interface? Is entertainment software on the way out? Where is computer-aided instruction headed? Are computer prices going to come down even further?

Many of these questions will be answered at the Summer Consumer Electronics Show (CES) to be held in Chicago the first week in June. For now, by looking at the events of recent shows, we may be able to spot some trends and speculate on the future.

At the 1982 Summer CES, the home computer was beginning to be recognized as a desirable consumer product. Texas Instruments had paved the way with new people-friendly technology never before offered to consumers. And while TI had been concentrating on advanced work in the electronics lab, Atari, Commodore, and others had not been idle—they were busy planning intensive marketing campaigns to scoop up their share of the new market. The airwaves soon filled up with the celebrity-inspired sounds of the "new tech" advertising.

A year ago, CES was full of start-up third-party developers promoting their software for the various home computers. Manufacturers of both hardware and software began to ramp up production, and store shelves began to fill. In a battle for market share, prices of the home machines were repeatedly slashed, rebates became the norm, and a full-scale price war catapulted the market into a state of confusion.

Competition is really heating up in the home education market. DesignWare has offerings in this area for Apple, Commodore, and IBM.

The Learning Company was one of the first software companies to market programs for the IBM PCjr. Their popular education series is targeted to this new home user.

Then, on October 31, 1983, TI announced its withdrawal from the home computer market. One day later, IBM "unshelled" its PCjr (code-named "Peanut"). In the six months since then, we've seen Mattel and Timex pull the plug, Coleco run into quality control and shipment trouble, Commodore announce—then put on "indefinite hold"—a new line of machines, and Apple

begin to posture in the home market while unveiling its long-awaited Macintosh.

Beginning at this summer's CES we can expect to see some interesting marriages between home computers and other high-tech devices such as laser disks. There should also be evidence in abundance that home computers will be used more as tools, not toys. The maturing software industry is hard at work developing home productivity packages for the most popular computers. In the past, home computer software development centered mainly around arcade gaming. Now

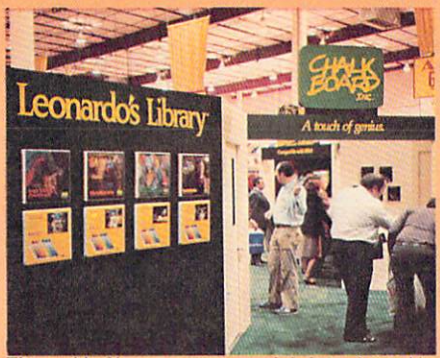
First Star Software, Inc., is one of the many new vendors now competing at the low end of the home market with several titles for Commodore and other machines.

the trend is towards entertaining educational programs and productivity tools.

The first productivity packages have, in fact, already arrived in the form of scaled-down office tools (word processors, spreadsheets, and the like). But the home environment demands its own set of productivity tools. Areas such as energy management, dietary and menu planning, physical conditioning, wardrobe coordination, and security await the creative genius of today's software developers. The Summer CES will shed light on new software that is more supportive of the home user's needs.

Industry watchers expect a continued explosion in Commodore add-on products and an abundance of new software for the IBM PCjr and the Apple Macintosh. There will also be more consumer-friendly software for the Apple IIe as a drop in price causes its user base to skyrocket.

New networking options will make these machines more attractive to schools and small businesses. Much of the latest productivity software really needs the speed and storage capacity of a "hard" disk drive (also called a Winchester) to be effective, but the



The world of home computer art is being enhanced by the light pens, mice, and touch pads offered by companies like Chalk Board, Inc. Adequate software support for new peripherals is crucial.

cost of putting one on each computer is too expensive. Networking with four or five computers that can share a printer and hard disk will provide an effective solution for small businesses and schools.

Prices on both computers and software should continue to drop during the remainder of 1984, though not as fast as during the recent price wars. Instead, look for more advanced products at prices close to those for earlier versions. We expect to see 128K-byte machines with a single disk drive start to become the *de facto* standard for the productivity home computer by year's end—with 64K-byte machines dominating the entertainment/educational niche.

Will all this come to pass? The answer lies in Chicago, where the Summer CES will set the stage for the final act of this year's home computer drama.

—Digest Staff

Entertainment software is certainly not dead, just more sophisticated and flashy. Adventure International has been producing challenging games for several years.

Software Trends

HOME-TRIAL OR PIRACY?

Software companies have resorted to paying for the consulting expertise of former pirates who help them devise ways to thwart program theft. Now that software rental companies threaten to cut into sales, the companies are more concerned than ever to develop un-copyable software. The software lease outfits argue that it makes sense for customers to rent software and try it out in its home setting before they buy it, and they'll give the customer a discount on the purchase price if the software works out. Outraged software companies counter that giving customers the opportunity to rent software doesn't create later sales. They say they're losing those sales altogether as the customers make their own copies of the software before they return it to the rental company.

FEMINIST NON-VIOLENCE

Elizabeth Stott and Lucy Ewell of Rhiannon believe personal and home computer software has always been designed and targeted with a male audience in mind. They're out to rectify the situation with a series of four adventure games designed expressly for girls. Addison-Wesley will publish the non-violent adventures—*Jenny of the Prairie*, *Chelsea of the South Sea Islands*, *Cave Girl Clair*, and *Lauren of the 25th Century*—for the Apple II, II+, and IIe this spring. Each game will retail for $39.95.

A MARRIAGE MADE IN JAPAN

TV and computing finally have more than a monitor in common. Sony Corporation has developed the technology for locally transmited games and data—including still pictures—over cable TV lines. Cable companies in Japan will be the first hardware customers this summer. Not much is scheduled here yet, with most U.S. companies just in the "interested stage," but Broderbund has granted cable rights to their software to The Games Network of Orange County, CA.

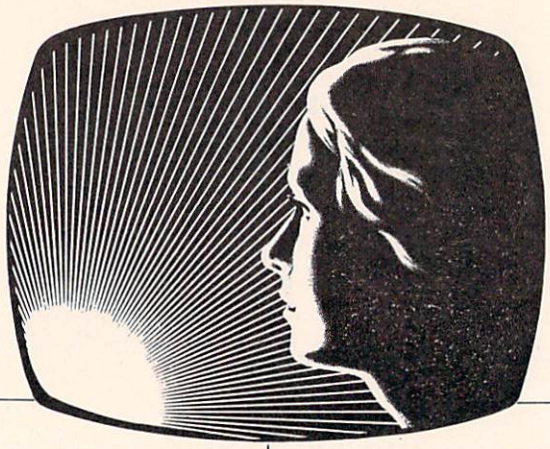
SOFTWARE HARDSELL

Software companies are adopting the aggressive marketing strategies that characterized the home computer hardware wars of last year. Sierra On-Line offers $5.00 rebates on its educational programs and will give away a free game for every purchase of two; a coupon from their *Quest for Tires* nets the buyer a free "B.C." poster. Spinnaker will give away a free program for every purchase of four; and Parker Brothers offers a $15.00 rebate for every purchase of two games. Epyx offers its "preview disks" of program excerpts at $2.50, and K-TEL courts dealers with budget-priced software ($10.99 for cassettes, $12.99 for diskettes) for guaranteed sale or exchange and offers to freshen dealers' wares every 90 days.

DIVISION OF LABOR: THE GOOD, BAD, & UGLY

The costs of maintaining programming staff have convinced CBS Electronics and Sirius Software that they should focus on marketing and buy programming designs from outside firms. Others have decided the reverse: Imagic will focus on program development and leave the marketing to other firms, and Synergistic will stop publishing and concentrate on developing software. Such neat divisions of labor don't always work out, though, especially when one or both of the parties continue (or decide later) to both produce and market software. A case in point is the bitter feud between Software Arts, developers of *VisiCalc*, and VisiCorp, the firm that has marketed the program since 1979. Software Arts claims that VisiCorp is marketing *VisiCalc* much less energetically now that VisiCorp's own windowing integrated spreadsheet, *VisiOn*, is out. It's now up to the courts to measure marketing "energy levels."

Computers Enable the Disabled



The day is coming when personal computers will be as important to handicapped people as their wheelchairs, hearing aids, and seeing eye dogs are now. Computer technology holds an exciting potential to allow handicapped people to participate freely in the mainstream of life. Unique applications of hardware and software can actually become the eyes, ears, voices, and hands of those who cannot use their own.

The concept may seem a bit strange to most home computer users, who depend on the conventional route of keying in programs, viewing monitor displays, and reading hard copy from a printer. How can a person with cerebral palsy who can't type, or a blind person who can't respond to screen prompts, use a computer? And what good is a personal computer to a quadraplegic, whose movement may be limited to nodding the head, raising the eyebrows, or even just breathing?

Handicapped users can take advantage of a computer's ability to receive and generate information in many forms. The computer doesn't really "read" the words we type in at the keyboard, or "talk" to us in the sentences we see on the screen. The Central Processing Unit works with electronic impulses: Input and output can be converted to whatever form is best for the user—whether it's visual, aural, or tactile. A computer can serve as an electronic translator that compensates for the user's disability.

The Future is Here and Now

This kind of "miracle technology" is not years down the road—the dream product of obscure and unfunded research. Speech synthesizers, which are already used in everything from videogames to talking Coke machines, can make verbal communication possible for people with severe speech difficulties. The user just types in a sentence, and the speech synthesizer produces an understandable message. Speech capabilites are also used by blind writers and students, who can have the computer "read back" what has been typed at the keyboard, thus allowing instant error-checking.

In many cases, accessories that were developed for commercial appeal are perfect for disabled users. Touch tablets such as the Koala Pad are sold as graphics aids and as an alternative for children too young to handle a joystick. But these digitizing devices were a big breakthrough for people with cerebral palsy. Someone who doesn't have the motor control to write legibly, but who can scrawl signs with some kind of regularity, can use the Koala Pad along with a program that recognizes those signs and translates them into print.

Light pens, also marketed as an artistic novelty or a child's toy, can be held in the mouth or strapped to any motile part of a paralyzed person's body. They can be used to select menu options, draw on the screen, or even "type" commands on a simulated keyboard display.

Much of the new "enabling" technology was pioneered by disabled programmers who wanted to continue or begin a career in computer science. Programmers who lacked the manual dexterity to type in their programs were the innovators behind the development of special keyboard overlays, breath-controlled switches, and levers. The Optacon, a device that translates output from a monitor into braille, is used by blind programmers in specialized vocational training programs. Programs such as *PC Speak*, which lets a blind user scroll through the program while a speech synthesizer reads each screen aloud, were originally developed to make spreadsheet programs like *VisiCalc* more accessible. Developments of this type can open doors for disabled people in all walks of life.

Better For Everyone

What's merely convenient for the average home computer user can be truly liberating to the handicapped. For example, the Apple "mouse," now used with the Lisa and Macintosh models to make word processing and spreadsheet programs faster and more fun, can also be used as a screen pointer to make standard software accessible to people who can't type on the keyboard. Likewise, the IBM PCjr's programmable keyboard can also make multiple keypresses manageable for a one-handed typist, or for a paralyzed person who uses a mouth stick to punch in commands. And a program that dials the telephone may be just a novelty for most people, but can mean more independence for a mentally retarded person who chooses to live alone.

Other trends, such as electronic mail, at-home shopping, and computerized information sources will provide a new way of life for people unable to go out or communicate by telephone. Communications networks and electronic bulletin boards are a boon to deaf people, who can work or socialize via the keyboard, instead of depending on teletype systems that can be cumbersome, and are usually owned only by other deaf people. The computerized systems will give them a standardized means to interface with a much broader segment of society.

Meeting the Challenge

Many new markets and industries have been opened up by this computer-hungry sector of the population. Some handicapped people have gone on to form small companies specializing in the development and marketing of hardware and software to meet the needs of disabled people. Others have initiated training programs to help disabled workers enter the job market. IBM and other corporations have turned to consulting firms for help in making their workplaces accessible to disabled employees, and computer skills are becoming increasingly important for rehabilitation counselors and trainers.

But many handicapped-rights advocates worry that progress is too slow and the equipment too expensive to benefit the people who need it. They are concerned that disabled people's needs will be bypassed in the rush to bring out profitable mass-market products. Dedicated companies that place need before profit and sell goods at cost do exist—but are not the norm. Disabled activists have organized to advise computer companies of their priorities, and to lobby for action. They have developed information networks and specialized publications to help people learn about and locate what they need.

So keep an eye on the desk next to yours. You may soon be working alongside a quadraplegic who uses a voice recognition module and microcomputer to thumb through files and write up a report, the same way you use yours to help keep your checkbook in balance.

—Joan Killough-Miller

New Tech News

A NEW MICRO-FLOPPY FROM JAPAN

The Apple Macintosh is making the Sony 3 1/2" floppy disk drive a standard in America, as the new micro-floppy gains in popularity. This same drive—although a fixed-speed version—has helped make the Hewlett-Packard 150 a very popular business computer. Some third-party developers have also come forward with 3 1/2" drives for other machines (e. g., the Apple II series and the IBM PC) Meanwhile, rumor has it that Mitsumi Electric in Japan has been perfecting a 2.8" drive which should prove to be very stiff competition for any of these products. The drive utilizes a single spiral track (as opposed to the standard concentric circle tracks) that allows it to access program and sequential files at incredibly fast speeds. It is said to load a 64K-byte program in about 8 seconds, but will be priced in the same range as an inexpensive cassette recorder. The new disk drive's medium is being developed by Hitachi, Ltd. (a subsidiary of Hitachi Maxell), and its reported $2 price tag makes it a prime candidate for home computer applications.

NEW APPLE IIe

Apple is rumored to be testing an Apple II series compatible "transportable" computer. It is said to be based on a 65C02 microprocessor and to have 128K of RAM, one 5 1/4" single-sided, single-density disk drive, and a built-in keyboard in a 13" x 17" x 4" unit.(Note that a monitor is not included.) All existing Apple software would run on the machine, but while the new microprocessor is a step up from the old 6502—featuring two new addressing modes and 27 more op-codes—it can't boast the power of newer 16- and 32-bit processors. No release date has been set, and while the price tag is expected to be below $1000, Apple so far will say only that it "does not comment on unannounced products."

WHAT WILL jr RUN?

Spinnaker, a leading third-party publisher of educational software, has once again distinguished itself by being the first publisher outside IBM to ship software for the PCjr. All three of the cartridges—*Facemaker* ($34.95), *Fraction Fever* ($34.95), and *Kindercomp*

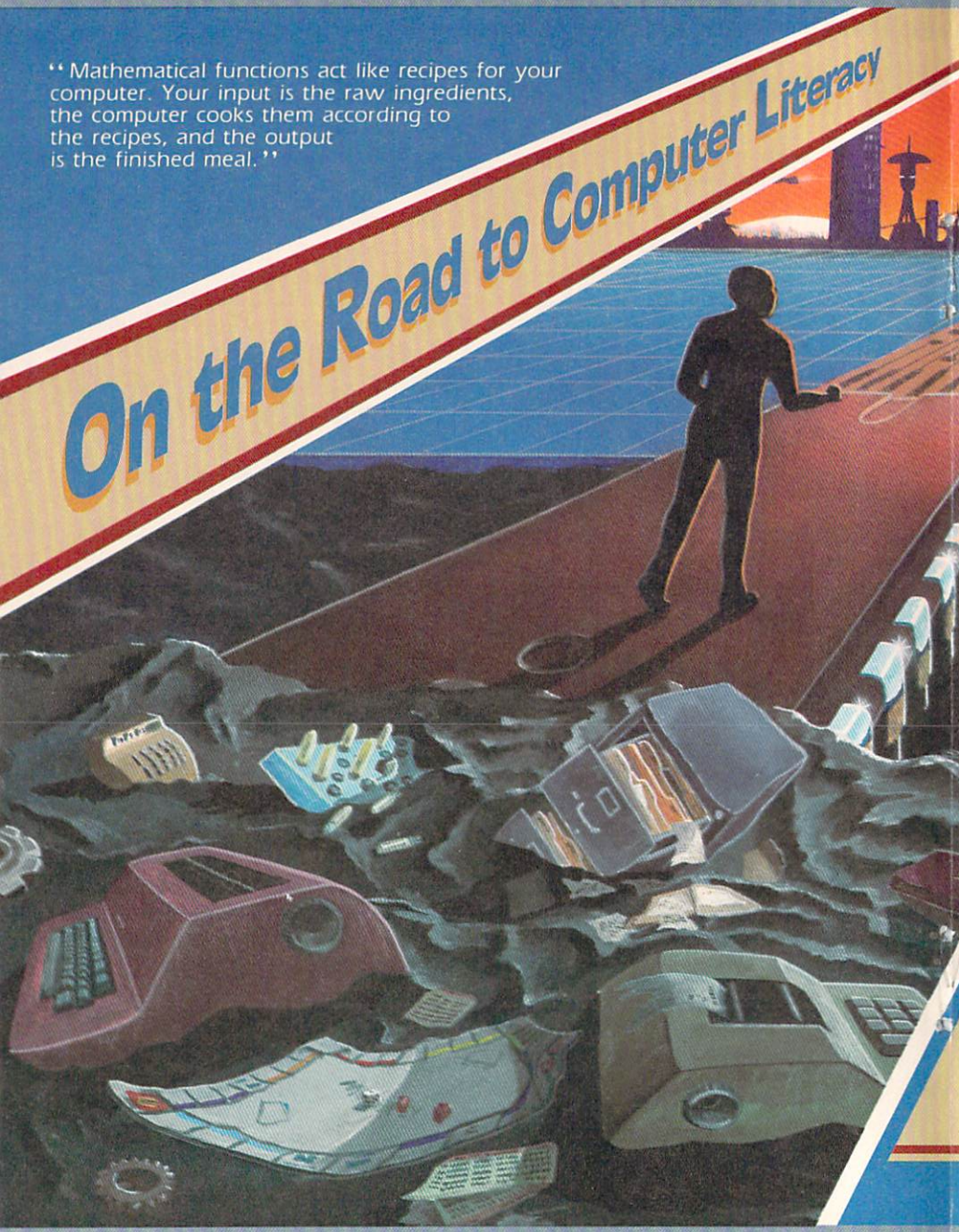
($29.95)—come from their educational software "will-sell" list and are the result of months of careful translations. Along with Sierra On-Line's *HomeWord*, these Spinnaker offerings should keep PCjr users RUNing until the second wave of PCjr software washes up on the retail beaches.

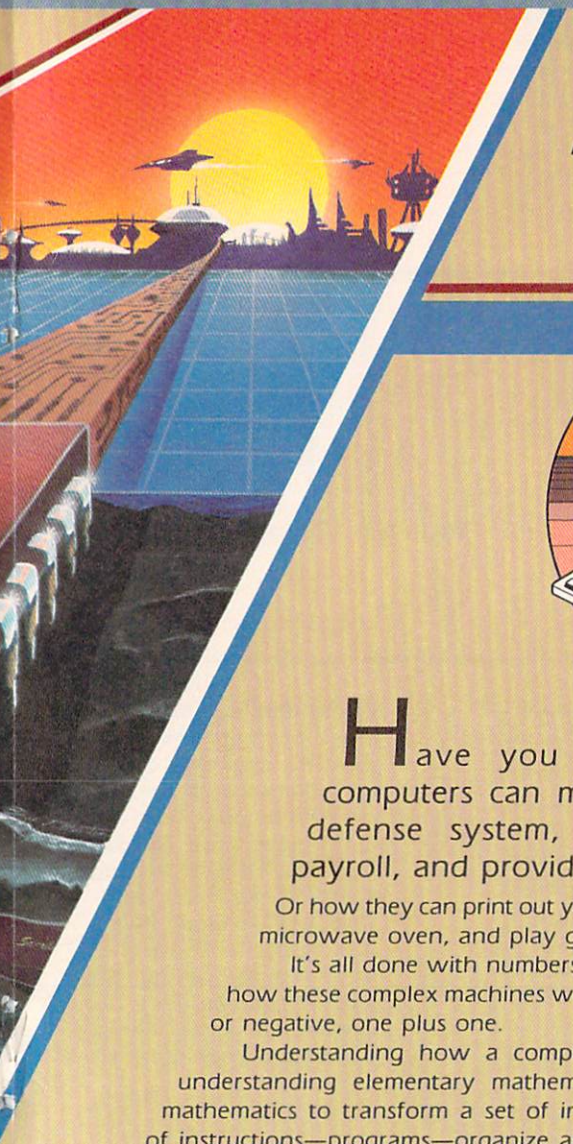
SINCLAIR PREPARES FOR QUANTUM LEAP

The company that brought us the first micro-micro for the home (the ZX-81) will soon be releasing the QL (for Quantum Leap) personal computer. Sinclair's 68008-based 128K RAM machine will include a dual continuous-tape-loop *microdrive* for up to 100K-bytes of data and sell for a mere $499. A .5-megabyte RAM add-on will follow. The operating system will be called Q-DOS, and the native language will be SuperBASIC (an enhanced version of Sinclair's Spectrum BASIC). In addition, a London-based software house, Psion Ltd., is working on applications for the new machine that include the four most commonly used business packages: word-processing, spreadsheet, database management, and graphics. The exact release date hasn't been announced, but third-quarter mail order availability in the U.S. is expected.

AND NOW. . .jr COMPATIBILITY

While many continue to ask, "Is it PC-compatible?" Mindset of California has introduced a new home computer that may start people asking, "But, is it compatible with the PCjr?" A fully configured PCjr lists for $1269. Mindset should be a faster and more powerful machine because it uses the 80186 (an upward-compatible relative of the the 8088 and 8086); it is PC-compatible at less than $1700. The System Unit, featuring a detachable 84-key low-profile keyboard, 2 cartridge slots, a custom video processor, Microsoft's GW Basic—but without a disk drive—carries a suggested retail price of $1099. Rumor has it that Mindset developers have plans for an adapter for the cartridge slots that will give Mindset access to PCjr's cartridge-based software. Looks like this new start-up firm has its mind set on giving Big Blue a run for the up-scale home computer market.





COMPUTERS:
MAGICAL MATH MACHINES

MAPPING & FUNCTIONS

Part 2

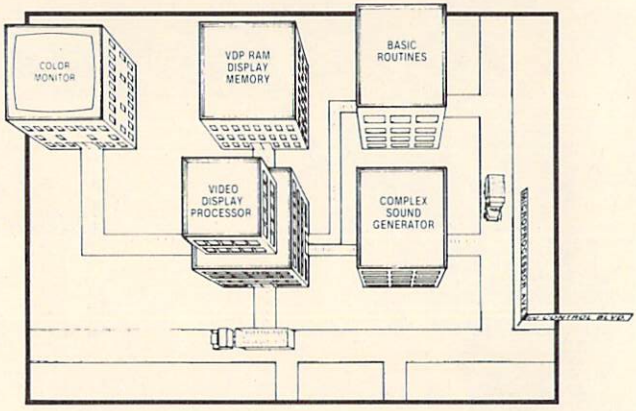
Have you ever wondered how computers can monitor a nation's missile defense system, generate a company's payroll, and provide medical diagnoses?

Or how they can print out your Christmas card list, control your microwave oven, and play games with your four-year-old?

It's all done with numbers—simple mathematics. The key to how these complex machines work is as basic as on or off, positive or negative, one plus one.

Understanding how a computing machine "thinks" involves understanding elementary mathematics. All computers use simple mathematics to transform a set of instructions into action. These sets of instructions—programs—organize and transmit data, solve problems, play games, and—through interfacing with other types of machines—even perform physical labor.

One of the most important mathematical ideas that all computers use is **mapping**. Computers use maps in much the same way we do. A city map isn't identical to the city itself—it's just a piece of paper after all—but once you understand what its symbols represent, it becomes a tool to help you find your way around. The map is useful because there is a one-to-one correspondence between the symbols on it and the actual places they represent. If you see the symbol for a mountain on a map and the symbols for the roads leading to the mountain, you can follow the actual roads and find the real mountain.



Computers use maps in much the same way we do. The map of a city isn't identical to the city itself—but once you understand what its symbols represent, it becomes a tool to help you find your way around. The map is useful because there is a one-to-one correspondence between the symbols on it and the actual places they represent.

Mapping in mathematics is very similar—one set of symbols is said to "map" a group of numbers or other symbols onto another group. It's a lot like writing a report. You have a group of thoughts to convey that you map onto a piece of paper in the form of notes or an outline. Because you do the mapping yourself, you know what thoughts the notes stand for. Later, you can map those notes into readable sentences and paragraphs so that your original thoughts are conveyed to your reader. We have rules of grammar and syntax that allow you to map the sentences and paragraphs back into your thoughts, so that in a sense you have "mapped" your thoughts onto those of your reader.

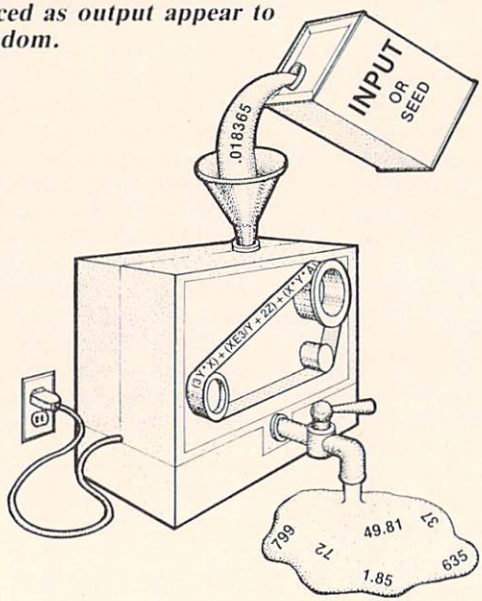
Mapping With Mathematics

Mathematics has a variety of rules for mapping. One major type of mathematical mapping is called a **function**. A function maps one set of things onto another in a unique way. You can think of these function maps as working like a recipe for baking a loaf of bread. The recipe calls for specific quantities of ingredients—say, 4 cups of flour, 2 table-

spoons of honey, 1 package of yeast, 2 teaspoons of salt, 2 cups of milk, etc. The recipe also includes instructions for combining the ingredients to make the bread. If you combine them according to these instructions, you will get the same amount of bread every time. In fact, if you follow the instrucions precisely, then you will get an identical loaf of bread every time (given identical ovens, relative humidities, temperatures, etc.).

Mathematical functions act like recipes for your computer. Your input is the raw ingredients, the computer cooks them according to the recipes, and the output is the finished meal. For a given set of numbers a mathematical function returns a **unique** answer. Because mathematical functions work like this, we know that when you use a specific function it will return its own unique answer every time and that any other answer is wrong.

A typical program takes advantage of a specialized function called a "random number generator." The function in your program accesses this and tells the computer to respond with a unique output. It does this by entering a number into a complex mathematical function. The series of numbers produced as output appear to be random.



The idea of a unique output (the bread or the answer, depending on what you put in) does not mean that the only way to get a different output is to use different inputs. Some functions give the same answer for several different sets of input. For example, the function y=0*x always has the same answer because any real number x multiplied by zero equals zero. There is, however, one critical test of a function: For any given input, a unique output results.

To understand functions better, let's look at how they are used in our lives. Income tax (a subject near and dear to our hearts) is a good example of how functions affect us all. The amount of tax you pay, the number of people you support, and your tax-deductible expenses are all functions of your income. If, for

a given income and set of conditions two people paid drastically different taxes, it would cause a lot of problems. Needless to say, the functions are so complex that it might seem they do pay drastically different amounts, but generally the mathematical functions used see to it that for a given income and deduction schedule a unique amount of tax is paid.

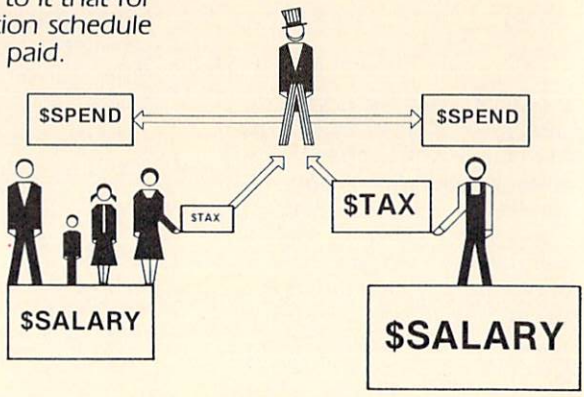
A Function Machine

The reason we've spent so much time explaining mapping and functions is that computers are primarily function machines, and their programs are the functions that they implement. When you tell a computer in BASIC to PRINT "HELLO" you know that it will put the word HELLO on the screen. That's because it understands that the BASIC function PRINT means to put on the screen the word or words that are contained within the quotes. Every time you tell it to PRINT it will do so, because the function PRINT has certain rules that make it always give a unique output for a given input.

When you write a program on a computer, it's like writing a whole series of connected mathematical functions that will give a unique output for a given input. When your program RUNs it will always give the identical output for identical input. The output will always be uniquely connected to the input by the functions that the computer implements in the program.

What about game programs? They change every time played (if they're any good), even if the same options are selected. Though this seems contradictory, there is a mathematical answer. Games employ a specialized function called a **random number generator**.

A function in your program accesses this specialized random number generator and tells the computer to respond with



The amount of tax you pay is a function of the number of people you support and salary. Mathematical functions used by the government see to it that for a given income and deduction schedule a unique amount of tax is paid. Some functions yield the same output (spendable income) for different sets of input (salary and deductions).

a unique output. A number called the **seed** of the random number generator is entered into a complex mathematical function, and a series of numbers is produced that seems quite random. The program then instructs the computer to make decisions based upon these seemingly random numbers, which, in turn, make the game operate differently every time.

"... functions can even be made to appear to be non-functions if you know how to use them."

In most game programs, if you could seed the computer's random number generator in the same way every time, you would always get the same game. Luckily, computer programmers see to it that a random seed is entered into the random number generator each time the program runs. This makes the game respond with a unique output each time. So you see, functions can even be made to appear to be non-functions if you know how to use them.

Now that you know computers are based on functions, you see how they can be so useful. If you want your computer to solve math problems for you, it's obvious that its mathematical functions will figure them out correctly every time—if you program the functions correctly. If you program them so the functions don't adhere to the rules of the problems, you will get a lot of wrong answers—though you can rest assured you will get the same wrong answer for the same problem every time because the computer will be consistent in giving you unique output for your input.

These function machines (computers) can also be interfaced to data storage devices that will keep track of vast quantities of data for you. They can be programmed to use functions to check for system errors. A computer's functions can even be made complex enough to control entire manufacturing processes—all because every given input will always produce a unique output. HCD

Industry Watch

COURSE CHANGE AT COMMODORE

Recent turnovers in the management at Commodore have kindled much speculation about the company's future. When Commodore's founder and president, Jack Tramiel resigned in January, four other top executives soon followed suit. Shortly afterward, the company announced that it would delay production of its new 264 and 364 computers. This decision may stem partly from management upheaval and partly from the critics' reaction to the new machines at the Winter Consumer Electronics Show. Many observers felt that the machine was not a significant improvement over the C-64. And the fact that it also wasn't directly software-compatible with the C-64 didn't help matters any. Clearly, Commodore is going through some growing pains, making many C-64 owners uneasy about continued support of their favorite machine. But it is unlikely that Commodore will soon scuttle its most popular product. The Commodore 64 is selling extremely well right now, and provides a healthy cash flow for a company in transition.

VIDEOTEX COMES CLOSER TO REALITY

Home computer-based shopping, entertainment, banking, and other kinds of information retrieval—videotex—has been in the formative stages for years, but only now does it seem ready to take off. Three huge backers—CBS, IBM, and Sears—have undertaken a joint venture intended to tap computer households. Other companies have attempted similar ventures, but none has had the combined clout that is possible with a conglomerate such as this. The companies predict that they will need several years to put their videotex plan into full operation, but that it could become a 30 billion-dollar industry within a decade. The road to riches won't be easy, however, because sales of that magnitude require at least half the homes in America to have a home computer. The "Big 3" expect that the availability of videotex services will be a strong enough incentive for consumers to buy a computer—thus making their sales forecast a self-fulfilling prophesy.

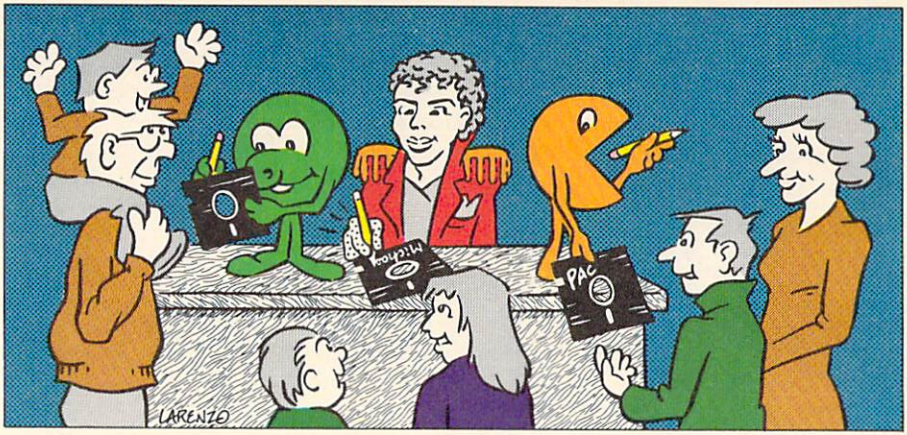
DATA DEWAR'S DELIGHT IN DOING

The Dewar's Profile of Computer Professionals is based on interviews with over 300 men and women in this field. According to the survey, job satisfaction is extremely high: over 90% would not leave the field. Over 80% would not even change to another area in data processing. It may come as a surprise that this "technical" industry has a high regard for language skills. Creative liberal arts graduates can do well in may areas of this business. The chief complaint heard among computer professionals is that there is still too much paper work. Although some observers predicted that computers would help eliminate this burden, it is still the bane of many people employed in the computer field.

TECHNO-WAR WITH JAPAN

Food for threat: First we nearly reached Sputnik-level hysteria over Japan's progress towards developing a super-powerful computer. Then we shuddered at reports of super-cheap, powerful micros soon to cross the Pacific to devastate the machines we know and love. The latest scare is seen as a Japanese infringement on software produced and copyrighted by U.S. firms. Japan's Ministry of International Trade and Industry proposes to enhance software development in that country by forcing foreign companies to license software to Japanese firms if such is in the national interest. American software companies see Japan's proposal as nothing more than piracy, and they are protesting it via the U.S. embassy in Tokyo. On another front, Japanese developers of machine tools for manufacturing integrated circuits are making great strides to capture the world market. In a market whose sales could approach six-billion dollars this year, the Japanese could easily slice off a third of the pie—more than doubling their share from last year. U.S. producers of such equipment are countering the Japanese attack with unprecedented expenditures for research and development.

Celebrity Software
and Software Celebrities



Celebrities have long been involved in the promotion of computers and software. One has only to think of Bill Cosby, Alan Alda, and Charlie Chaplin on TV, of Leonard Nimoy and even Minnesota Fats at recent computer shows. But the luster of celebrity will shine from within soon, as software companies marry expertise and famous names and scenarios with computer programming. There's even a movement afoot to create software's own heroes and celebrities.

While Japan diligently perfects the technology for expert, knowledge-based systems, software companies in the U.S. have blithely embarked on "expert-based" software production. The new software comes from the collaboration of famous and near-famous experts in their fields with software programmers. James Fixx, author of *The Complete Book of Running*, has contributed fitness lore and his name to MECA's *The Running Program*. Investment advisor and best-selling author Andrew Tobias has developed *Managing Your Money* with MECA's programmers.

We're not sure how extensive his actual contribution will be, but Mr. Rogers will at any rate lend his prestige to CBS Software's preschooler software, and Sesame Street's Big Bird, Cookie Monster, and Ernie will star in Atari's educational games aimed at the 3 to 7 years set and promoted by the ubiquitous Alan Alda. Other famous names and characters are coming on-line: the late Bruce Lee's name has launched Datasoft's new martial arts adventure game; Marvel Comics' Super Heroes(tm) will appear in software from both First Star and Adventure International; and Walt Disney characters will star in software from Atari.

Famous fiction writers will lend their illustrious names and scenarios to computer games too. Simon & Schuster has prevailed upon the likes of *Galaxy* editor and seven-times Hugo Award nominee James Baen to prevail upon the likes of John Chenault (*Snake!*) and Stephen Walton (*Starclash II*) for scenarios. And next season the Baen series will release Robert Heinlein's *Glory Road*, Stephen R. Donaldson's *Animal Love*, Poul Anderson's *The Game of Empire*, and Larry Niven's and Jerry Pournelle's *Inferno*. Simon & Schuster will also turn Douglas Adams' *The Hitch-hiker's Guide to the Galaxy* into an adventure game series. Epyx will release *Robots of Dawn*, based on the book by Isaac Asimov. Miyamoto Mushashi's 300-years-old *A Book of Rings*, recently a best-seller among businessmen who find

parallels between their concerns and those of a samurai warrior, will go on from that unlikely success to become an adventure game. Authors' contracts will have to include software rights as well as paperback and movie rights from now on.

Lest it run out of celebrities from other media, software has also begun to turn out its own notables. Scott Adams has been something of a cult hero among fans of his adventure games. Now his software company, Adventure International, will release the first of its series of adventure games featuring Marvel Comics Super Heroes as a Limited Edition Scott Adams Signature Series. Along with Adams, First Star Software's Head of Design and Engineering, Fernando Herrera, is among the first true software celebrities. His games have won awards, and he has been honored by both the Toronto Film Festival and UCLA's Video Game Conference. Herrera was the first software author to conduct an autograph signing, but he's not likely to be the last. This spring Simon & Schuster will launch their software writers on "author's tours." Their first candidate for the star system treatment is Sat Tara Singh Khalsa, a programming Sikh whose *Typing Tutor I* and *Typing Tutor II* for Kriya Systems are already best-sellers. Simon & Schuster will send Khalsa on a media blitz this spring to publicize *Typing Tutor III*(tm). In time-honored author's fashion he'll appear under Simon & Schuster's auspices on radio and TV talk shows, at press interviews, and at autograph signings in eleven major cities. And to think that only a year or so ago programmers were fixed in the public mind as those pasty-faced white-socked fellows happily programming in obscurity!

—Erin O'Connor

Why . . . cont. from pg. 5

added convenience and indestructible packaging, the disks sell for approximately the same price as premium-grade 5 1/4" diskettes.

In comparing the amount of data storage available on these smaller disks, Sony says its disks have 437.5K-bytes maximum capacity, which is more than a double-sided, double-density 5 1/4" disk on the IBM PC. Technological advances have made denser storage possible (i.e., more tracks per inch), due in part to the more rigid packaging and the increased stability of the way the disk is held in the drive.

The 3 1/2" disk drive on the Hewlett-Packard 150 uses a format such that the disks hold 270K-bytes. Meanwhile, the Macintosh puts 400K on each disk by utilizing a variable-speed motor in the drive.

You may be wondering why these advances have not been applied to 5 1/4" disks. Well, now they have. Kodak has just released a 5 1/4" disk drive that will allow a special disk to hold nearly 10 times the normal amount of data (3.3 megabytes). However, interfaces are not yet available for major computers (See *Home Computer Digest* 1.1 for details on this drive).

Even with all the advantages that the smaller disks have, a great furor has arisen over which of the small disks (3", 3 1/4", or 3 1/2") will be the "standard" micro-floppy in the industry. Many people say the Sony 3 1/2" has already become the de facto standard, but considering the lack of any standard format (Sony, H-P, and Apple all use different ones), one of the other disks may yet gain supremacy in the fight to become the standard disk. In April, the American National Standards Institute (ANSI) will vote on a resolution declaring one of these different micro-floppies as the *industry* standard, but it's our guess that the *marketplace* will determine the *true* standard.

—Roger Wood

Computers in Education

PLATO GETS NEA SEAL OF APPROVAL

Control Data Corporation's PLATO courseware—available on the Apple, IBM, and Texas Instruments home computers—has earned the coveted approval of The National Education Association. Evaluations conducted by teachers and programmers found PLATO's Math, Foreign Language, and Computer Literacy packages (16 specific titles in all) technically reliable, educationally sound, and easy to use for both teachers and students. The NEA's nearly two million members have yet to react to the news, but CDC is busy converting more titles to run on the Apple and IBM.

UNIVERSITIES HAVING MAC ATTACK

Twenty-four American universities have joined the Apple University Consortium (AUC) to get the $2500 Macintosh at a large discount. While retailers worry about how such discounts will affect their hardware profits, Apple is reportedly counting on $60 million in commitments from consortium members over a 3-year period. In return for their discounts, the universities are each expected to purchase $2 million in Apple products as well as develop curriculum, new applications, and educational software for the Mac. IBM will undoubtedly keep a close eye on what happens because some AUC members also have similar contracts with Big Blue. The winner in this battle for dominance in academia's hallowed halls may not be IBM, Apple, or commercial software producers. With educational software coming out of prestigious universities, the consumer will emerge the real victor.

A VIDEO GAME A DAY...

The new Basic Books publication *Mind At Play: The Psychology of Video Games* offers arcade fans new justification for the time spent in pursuit of the supreme score. Authors Geoffrey R. Loftus and Elizabeth F. Loftus applied the results of psychological experiments to video games—"electronic Skinner Boxes"—and found that they have a positive effect on memory retention. The same elements that keep you depositing coins also condition your mind to expect the unexpected, to remember details, and to respond quickly to each encounter. The book presents evidence that video games are fun, educational, have military training potential, and can even be substitute friends! Where else can you get all that for a quarter these days?

SCHOOL DAYS ON 25 DISKS OR LESS?

Kaypro, looking for ways to court the home computer market, may be eyeing the schools. At Winter CES, David Kay, Kaypro's vice-president of marketing and sales, implied that their next move could be a combination of putting large quantities of educational courseware on diskette and creating a low-cost, disk-equipped computer to lure educators and then parents to the Kaypro product line.

COMPUTERS: PRESCHOOL TOOL OR TRAUMA?

Are preschools with computers preschools with a plus, or is teaching technology to tots merely accelerated alienation? Experts feel that adult involvement makes all the difference. Certain child-development specialists have warned against long periods of passive, non-interactive computer use that utilizes the machine as no more than an electronic baby-sitter. On the other hand, if a responsible adult uses the computer with the child, the experience can increase his self-esteem, improve his social interaction, and sharpen his cognitive and motor skills.

SECRETARY BELL RINGS EDUCATORS' CHIMES

Hot on the heels of a Presidential directive to bring computers into every student's studies comes a message from Education Secretary Terrell Bell offering federal funds to educators and school systems to develop educational software. Bell plans to give software development a "very high priority" in hopes that quality software, generated by those who need it, will encourage in-school use of computers, and will be available to the public at low prices. This news is causing concern in commercial software houses who may claim that this is government's attempt to control the software market as well as what children learn in the classroom. Secretary Bell asserts that this funding program will benefit students, which should be the main concern of all educational software developers.

Novel Applications

AND NOW THE GOOD NEWS

The Good Book is now available on diskette for Apple users, and versions are in the works for IBM and CP/M users. Bible Research Systems of Austin, Texas, offers *THE WORD processor*, which contains the entire King James Version of *The Bible*. The software has built-in programs to analyze, display, cross-reference, annotate, and print Biblical text. The program will be a great boon for theologians, but somehow we don't think an END OF FILE message will ever replace "Here endeth the reading."

HIGH-TECH DECO

And now another reason to own a home computer: It can be your personal interior decorator. *Home Decorator*(tm), from SOFTRON, Inc., will take you in hand with expert advice on color, furniture, and room layout. With just a few keypresses you can install carpeting, paint walls, and rearrange the furniture to your heart's content. And there's no one to throw a fit if you want to try the baby grand in the far corner just one more time...

ON DISK AND READY-TO-WEAR

Looking for a lawn mower, a cruise to Curacao, or the latest new wave album for your nephew's Bar Mitzvah gift? Your computer may be able to show you just the thing, without the hassle of crowded malls or tedious walks through the yellow pages. CompuServe is pioneering an "Adformation" service that lets you shop at home through on-line catalogs of merchandise and services. More advanced than earlier "electronic shopping" schemes that listed only simple product descriptions, Adformation will contain a detailed data base on products in ten categories: books, magazines, credit card subscriptions, club memberships, gardening and lawn care, national newspapers, records and tapes, financial services, computer products, and travel/vacation. On-screen instructions will guide the shopper in making the purchase directly from the manufacturer or through a retail outlet.

THE COMPUTER THAT CLIMBED MT. EVEREST

Their tents are packed, the gorp is bagged, and the yaks are ready and waiting to go. This spring 16 American mountaineers and 15 Tibetan porters will attempt to retrace the "classic route" to the summit of Mt. Everest, following in the very footsteps of George Leigh-Malloy, the legendary British climber who wanted to climb Mt. Everest "because it's there." Stowed in the old kit bag, (among the seven tons of equipment the group will be hauling) will be an Okidata dot matrix printer hooked up to a Columbia VP portable computer. The system will facilitate biomedical research and financial accounting, functioning within a portable research tent, which will be heated to protect the hardware against the −30 degree chill.

The computer will coordinate the difficult logistics of the trip and insure that gear is unloaded in the proper order at each campsite. It will maintain a budget, monitor food consumption of both humans and yaks, and chart medical information such as pulse rates to warn against the danger of overexertion.

Dubbed *Ultima Thule* (a Greek and Latin phrase meaning "the outer limits of achievement"), the expedition is a far cry from the days when backpackers took the tags off their teabags to cut down on the weight of their packs.

WANNA MAKE A BET?

Hi there, sports fans. Are you tired of taking a beating in the office football pool? *Pro Sports Stats* is a software package from Eastern Computer Consultants that lets you play Monday morning quarterback before the weekend rolls around. The program gives you complete statistical records on professional football teams from 1970 to the present. Its data base includes win/loss records, point spread information, coaching history, records of playing surfaces, and other critical statistics. Using the operating program, you can plug the relevant information into your Apple, IBM PC, or Commodore 64 (plus several other popular models), consult your team's history, and pick a winner.

Gameware Updates

VIDEO GAMES SPIN OFF THE BIG SCREEN

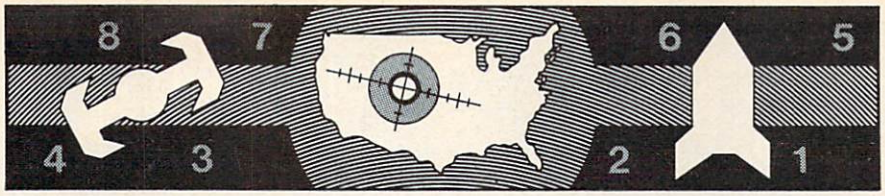
The movie release of *Dune*, Frank Herbert's science fiction classic, is expected to catch the fancy of many home computer users. This anticipation led a number of toy and video game manufacturers to vie for pre-release licensing agreements with Universal Pictures. Parker Brothers was the big winner in this contest and has been awarded the arcade and video game rights for the film, which is scheduled to be released sometime this year.

SAILING SOFTWARE COMING ABOUT!

Landlubbers and sailors alike can enjoy the excitement of an afternoon on the water with *Regatta*, a new sailing game by Howard W. Sams & Co. *Regatta*, for the Apple II, simulates sailing races on four different lakes while a clock on the screen shows elapsed race time. Penalties are recorded each time a boat hits a buoy or runs aground. The game comes complete with an illustrated instruction manual that defines some basic sailing jargon such as running, reaching, tacking and "yarning."

GO FOR THE GOLD—IN YOUR LIVINGROOM

Armchair athletes should be warming up the television sets and getting their sweatsuits out of mothballs in preparation for the summer Olympic Games. For those who want to start the action early, a variety of home computer and video arcade games is now available. *HES Games 84*, by Human Engineered Software, and *Summer Games*, by Epyx, have been added to the growing list that started with *Olympic Decathlon* from Microsoft. *HES Games 84* recreates a variety of summer games events including archery, springboard high diving, and weightlifting. *Summer Games* is actually a series of different games modeled after Olympic events. *Track and Field*, a surprisingly realistic arcade game that requires the player to use split second accuracy combined with a special touch on the keyboard, is expected to be adapted for the home computer in the near future.



COMPUTER WAR

A Review by Greg Roberts
HCM Staff

Name:	**ComputerWar**
Program Type:	**War Game**
Machine:	**VIC-20**
Distributor:	**Thorn-EMI** **1370 Avenue of the Americas** **New York, NY 10019**
Price:	**$39.95, cartridge**

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

The movie *WarGames* inspired this program that puts you in control of the computer at NORAD, the North American Air Defense System. You're dealing with a crisis: Someone has broken into the system and activated a war simulation program. Because the computer can't tell illusion from reality, it is about to fire retaliatory weapons, and that could mean the end of everything.

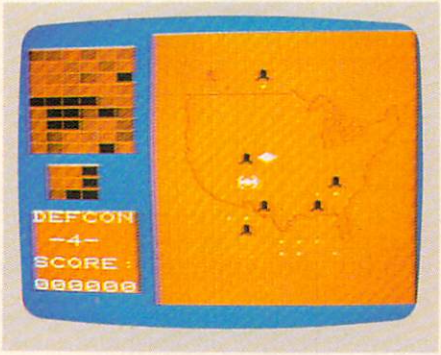
Your job is to make the computer believe you have eliminated the enemy missiles so that it won't launch the actual retaliation. You must wipe out the "attacking" missiles on the map. They are shown as white blips that move quickly towards their targets—missile silos in western North America. Move your cursor over the missile that seems closest to your missile base and press the fire button. Now the screen changes to a landscape that can scroll both horizontally and vertically, with arrows at the edges of the screen showing the direction in which the target is flying. Use your joystick to find the missile, set your sights on it, and fire. The targets are not hard to hit, but you must act quickly. While you are working on one missile, all the others are moving towards your bases. Once they knock out five bases, the computer responds with a real counter-attack and it's all over.

Should you succeed in destroying all the incoming missiles, there's more work to be done. You have to disable your own bases too, and you must break into the NORAD computer to do it. At the left of your screen are two grids, one much larger than the other, both filled with rows of small flashing squares. When the squares freeze, you have a few moments to set your cursor on the smaller grid and then try to place it over that exact configuration within the larger one. When you've matched the patterns, press the fire button. Now you can move the map cursor over your own missile bases and shut them down.

No Pushovers, These Missiles

The game is difficult, in my opinion. Everything happens so fast, it will take much dedicated play in order to get good at this one. But there's plenty of incentive to practice. The game's design is good, involving three distinctive elements of play: map, landscape, and matching grids.

All three screens are programmed in bright scarlet, and can be a bit much to stare at for long. The U.S. map stage of the game is a good piece of graphics work, although it doesn't feature much action—simply the tiny missile dots lurching across the landscape. The grid display blinks on and off so rapidly it is a real challenge to match the patterns. Finally, the rocket flying across a red, featureless landscape can be very difficult to track and destroy, but it explodes nicely when you do get a hit.



My negative reactions are not to the game's playability or level of engrossment, but to the story line itself. We are not engaged in the destruction of funny little aliens here—these are American towns. The map of your own country bristling with missiles is a grim reminder of reality—that we are all at the mercy of "leaders" who could destroy us in minutes. Perhaps the British designers of this game can detach themselves more easily from that aspect of the program—after all, the map is of the U.S.—but even they must realize that we are all in this together.

People need to escape from everyday pressures once in a while, and video games can be a useful medium for escape. But this game's handling of nuclear disaster puts it outside the normal guidelines for entertainment.

HCM

CASTLE WOLFENSTEIN

A Review by the HCM Staff with David Fulton

Name:	Castle Wolfenstein
Program Type:	Graphic Adventure
Author:	Silas Warner
Machine:	Commodore 64, Apple II series
Distributor:	Muse Software 347 N. Charles St. Baltimore, MD 20201
Price:	$29.95, diskette

System Requirements:
Disk Drive

	poor	fair	good	excellent
Performance				
Engrossment				
Documentation				

World War II is blasting its way across Europe and into the history books. You are a spy working in extremely dangerous territory held by the Germans.

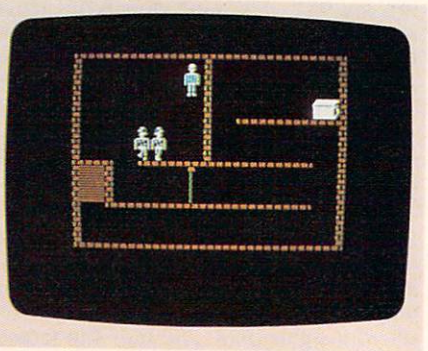
Achtung!! You are under arrest and are to be taken to Castle Wolfenstein. There your life will probably end, but if you show extraordinary skills, you may be able to escape. Inside the castle, awaiting your doom, you see a glimmer of hope as a cellmate slips you a pistol wth a full clip. He also tells you of secret Nazi plans known as "Operation Rheingold"—plans hidden somewhere in the castle, of enormous value to the Allies if they could only get them. As your cellmate is dragged away, never to be seen again, you resolve to find the plans and escape.

This game is hard to classify. It has many elements of an adventure game, but doesn't require typed-in commands. Instead you use joystick or keyboard to move your hero from room to room, looking for the usual valuables; in addition, you shoot up plenty of enemies, as if you were in an arcade game.

Graphics are spare. You see floor plans of the castle's rooms, each furnished with a chest that may hold items to help you in your quest. The only animation is that of the hero and the Nazi soldiers, and their movements are a bit crude and jerky. In this case, the Nazis are their old selves, rigidly goose-stepping across the screen. When they suddenly appear and start shouting at you, in German, it's enough to raise the hair on your joystick. The voices are not exactly as clear as Ella Fitzgerald on Memorex, but you can make out the *Achtungs* and *Schweinhunds*. You get a great deal of realism for a vocabulary of only eight words.

The play is further enhanced by stage props such as the mystery footlockers that contain anything from grenades to wine; you will also find bullet-proof vests and Nazi uniforms. These elements keep your interest keen in a game that can go on endlessly. There are roughly thirty rooms in the castle, and you will have to keep a careful record of your moves if you want to escape with the enemy's secret plans. When you do get out, you will be promoted for your bravery, then sent back to the castle on another mission even more difficult than the last. And so it goes, through eight levels.

Castle Wolfenstein offers two vehicles for play on the Commodore 64: keyboard or joystick. The keyboard is awkward to use. To direct the figure, you choose among the eight keys surrounding the F key. In firing at the enemy, you have to take aim by using the circle of keys around the L key, using the latter as your fire button. You are much better off with a joystick, but still you have to combine the joystick with various key presses. This is necessary, of course, because you must choose among a variety of weapons and actions, and your joystick has only one button.



This package includes a booklet offering you the basics necessary to load and play the game, but you are given few if any hints for success in winning the game. Only your own experience will teach you how to avoid or kill the enemy, and how to take advantage of locker contents.

The action is macabre. You are constantly killing people and searching them for useful supplies. If a German soldier surrenders to you, you will probably not know what to do—so you blow him away. This one is not meant for the Mr. Rogers crowd. No one can deny, however, that the challenge of *Castle Wolfenstein* is never boring, however lengthy the quest may be.

HCM

Cannibals . . . from p. 64

TI-99/4A

```
1820  RETURN
1830  REM  SCORE  DISPLAY
1840  A$="CANNIBALS:"&STR$(DEAD)
1850  PX=1
1860  GOSUB  1940
1870  A$="LIVINGSTON:"&STR$(SAFE)
1880  PX=2
1890  GOSUB  1940
1900  FOR  TD=1  TO  200
1910  NEXT  TD
1920  CALL  HCHAR(1,1,32,64)
1930  RETURN
1940  FOR  X=1  TO  LEN(A$)
1950  CALL  HCHAR(PX,X+3,ASC(SEG$(A$,X,1))
      )
1960  NEXT  X
1970  RETURN
1980  CALL  KEY(1,K,S)
1990  MX=0
2000  MY=0
2010  IF  (S<>0)*(K<6)THEN  2030
2020  RETURN
2030  ON  K+1  GOTO  2040,2020,2060,2080,202
      0,2100
2040  MX=1
2050  GOTO  2110
2060  MY=-1
2070  GOTO  2110
2080  MY=1
2090  GOTO  2110
2100  MX=-1
2110  IF  (D(6,1)=1)*(MX=-1)THEN  2020
2120  IF  D(6,1)+MX>18  THEN  1310
2130  CALL  GCHAR(D(6,1)+MX,D(6,2)+MY,CH)
2140  IF  CH=33  THEN  2020
2150  D(6,1)=D(6,1)+MX
2160  D(6,2)=D(6,2)+MY
2170  CALL  HCHAR(D(6,1)-MX,D(6,2)-MY,32)
2180  IF  D(6,2)>1  THEN  2210
2190  D(6,2)=31
2200  GOTO  2260
2210  IF  D(6,2)<32  THEN  2240
2220  D(6,2)=2
2230  GOTO  2260
2240  IF  D(6,1)>0  THEN  2260
2250  D(6,1)=1
2260  CALL  HCHAR(D(6,1),D(6,2),104)
2270  RETURN
2280  CALL  JOYST(1,MY,MX)
2290  MX=SGN(-MX)
```

TI-99/4A

```
2300  MY=SGN(MY)
2310  GOTO  2110
2320  DATA  96,00000000003F1F0F07,97,FFAAFFF
      FFFFFFFFF,98,00000000FFFFFFFEFC,99,00
      0000000C3C3C3C3
2330  DATA  104,1050503C113A2440,112,1010F
      C1028452210,120,0303010701F1F0F,12
      1,8080000E000F8F8F0
2340  DATA  122,0F0F07,123,F0F0E0,128,00,1
      29,00001028 44,136,7F3F1F0F0F070FFFFF,
      137,E0F0F8FCFEFFFFFB
2350  DATA  138,7F3F3F3F1F0F0F0703,139,00000
      0000000B0C0,140,E0F0F8FCFEFFFFF,14
      1,FF7E3C18,33,00187EFFFF7E18
2360  DATA  15,5,5,4,2,4,7,4,16,5,15,5
```

HCM

CANNIBALS (PCjr) Explanation of the Program

Line Nos.	
100-190	Program header.
200-580	Initialization.
590-730	Move cannibals.
740-900	Coincidence checks.
910-970	Scoring.
980-1060	Move explorer.
1070	Adjust joysticks.

IBM PCjr

```
100   '*********************
110   '*      CANNIBALS      *
120   '*********************
130   '  BY CARL CARROZZA
140   '  HOME COMPUTER MAGAZINE
150   '  VERSION 4.2.1
160   '  PCJR CARTRIDGE BASIC
170   '
180   '
190   '
200   KEY OFF:SCREEN 1:COLOR 2,1,:RANDOMI
      ZE TIMER:DEFINT A-Z:DIM D(3,2):DIM
      P(22,18):SOUND OFF:BEEP ON
```

Continued on next page

Cannibals. . . from p. 69

IBM PCjr

```
210 CLS:LOCATE 11,1:PRINT TAB(8) "TYPE 1- FOR JOYSTICK." TAB(8) "TYPE 2- FOR ARROW KEYS.
220 V$=INKEY$:IF V$="2" THEN 240 ELSE IF V$="1" THEN AA=-1:AX=STICK(0):AY=STICK(1):GOTO 240
230 GOTO 220
240 CLS:LOCATE 10,1
250 PRINT "**********"
260 PRINT "*"
270 PRINT "*          CANNIBALS"
280 PRINT "*"
290 PRINT "**********"
300 PLAY "MB T200ML03C4G4E4D8E8C2O2G2O3C4G4E4D8E8C2P2P4C4E4D8C8D2G2P4C4E4D8E8C2"
310 LI=0 ' DEFINE GRAPHICS AND COLOR
320 SH$="BR3R10H1L9H1R9H1L9H1R9H1L9U1"+"R8H1L7U1R4BR1R1H1L5U1"+"R4H1L3U1R2H1L1U1BD17BR17"+"R8G1L5F1R3G1L1"
330 MOUTH1$="BL1R3":MOUTH$="BU2F2"
340 BOG$="BR2BU1R10E1L12H1R14U1L14E1R12H1L10"
350 EX$="BR6BU4U3R1D1F1E2BG3G1R2D3R1BH3H1L1D1L1BR3D2L2D1" ' EXPLORER DEFINED
360 CN$="BR7BU4U3L1D1G1L1U2BF3E1F1E1U1BG2G1D2R2D1H3G1D2L1" ' CANNIBAL DEFINED
370 POT$="BR4R6U1L5U1L1R7BL3U2L4R7BL3U3L1D1" ' COOKING POT
380 FOR X=0 TO 22:FOR Y=0 TO 18:P(X,Y)=0:NEXT Y:NEXT X
390 CLS:LINE (0,145)-(320,200),1,BF
400 FOR Z=0 TO 30:X=(RND*(300))+10:Y=(RND(1)*(40))+145 ' WAVES
410 FOR N=0 TO 2:LINE (X-N*2,Y+N)-(X+N*2,Y+N),3:NEXT:NEXT
420 M=INT(RND*(220))+50 ' LOCATE SHIP
430 LINE (M,155)-(M+40,165),3,BF ' SHIP BODY
440 FOR N=1 TO 3:FOR Y=155 TO 150 STEP -1:P=M+(N*10)+50-Y/3:LINE (P,Y)-(P+5,Y),3:NEXT ' SMOKESTACKS
450 FOR Y=155 TO 165:P=M-(165-Y)*1.5:LINE (P,Y)-(M,Y),3:NEXT:NEXT ' PROW,STERN
460 FOR Y=160 TO 165:S=M+(165-Y)*.75+40:LINE (S,Y)-(M,Y),3:NEXT
470 S=M+44:FOR Y=155 TO 159:LINE (S,Y)-(M,Y),3:NEXT
480 LINE (M-2,155)-(M+2,143),2,B:LINE (M-1,154)-(M+1,144),3,BF ' GANGWAY
490 GOSUB 950
500 FOR N=1 TO ABS(10-SAFE)*10
510 X=RND*21:Y=RND*17+1:IF X=INT(M/14) AND Y>16 THEN 510 ELSE PSET (X*14,Y*8),0:DRAW "C1"+BOG$:P(X,Y)=-1:NEXT
520 Y=16*RND+2:X=21*RND+1:IF P(X,Y) THEN 520 ' *X AND Y INDICATE POSITION ON THE SCREEN
530 PSET (X*14,Y*8),0:DRAW "C2"+EX$
540 FOR R=1 TO 3
550 D(R,2)=RND*16+2:D(R,1)=RND*21+1:IF P(D(R,1),D(R,2)) THEN 550
560 IF (D(R,1)>X+4)+(D(R,1)<X-4)+(D(R,2)>Y+4)+(D(R,2)<Y-4) THEN 570 ELSE 550
570 PSET (D(R,1)*14,D(R,2)*8),0:DRAW "C3"+CN$:P(D(R,1),D(R,2))=-1:NEXT
580 Z=0
590 IF AA THEN GOSUB 1070:GOTO 610
600 GOSUB 980
610 IF Y1>18 THEN Y1=9:GOTO 760 ' FALL IN OCEAN
620 Z=Z+1:IF Z=4 THEN Z=1 ' CANNIBAL MOVE
630 DX=(D(Z,1)>X)-(D(Z,1)<X):DY=(D(Z,2)>Y)-(D(Z,2)<Y)
640 D1=D(Z,1)+DX:D2=D(Z,2)+DY
650 IF NOT(P(D1,D2)) THEN 680
660 DX=RND*2-1:DY=RND*2-1:D1=D(Z,1)+DX:D2=D(Z,2)+DY
670 IF (D2<1)+(D2>18)+(D1<0)+(D1>22) THEN 660:ELSE IF P(D1,D2) THEN 590
680 P(D(Z,1),D(Z,2))=0:IF (D1=X)*(D2=Y) THEN 850
690 P(D1,D2)=-1
700 PSET (D(Z,1)*14,D(Z,2)*8),0:DRAW "C0"+CN$:PSET (D1*14,D2*8),0:DRAW "C3"+CN$
710 D(Z,1)=D1:D(Z,2)=D2
720 IF AA THEN IF K<2 THEN K=K+1:GOTO 620 ELSE IF K=2 THEN K=0:GOTO 590
730 GOTO 590
740 PSET (X*14,Y*8),0:DRAW "C0"+EX$ ' MADE IT TO BOAT
750 PLAY "O3C8C8D4D8D8D4G8F8E8C8":SAFE=SAFE+1:GOTO 380
760 FOR WIN=0 TO 1:IF X=WIN+INT(M/14) THEN 740:NEXT ' CHECK FOR WIN
```

IBM PCjr

```
770 PSET (X*14,Y*8),C0:DRAW "C0"+EX$:PS
    ET (X*14,Y*8),C0:DRAW "C3"+SH$ ' SH
    ARK EAT
780 FOR R=0 TO 4:PSET (X*14,Y*8),C0:DRA
    W "C3"+MOUTH$ ' **CHEW
790 FOR T=0 TO 8:FREQ=RND*4000+100:SOUN
    D FREQ,1:NEXT
800 PSET (X*14,Y*8),C0:DRAW "C0"+MOUTH$
    :PSET (X*14,Y*8),C0:DRAW "C3"+MOUTH
    $
810 FOR T=0 TO 8:FREQ=RND*4000+100:SOUN
    D FREQ,1:NEXT ' **MOUTH CLOSED
820 PSET (X*14,Y*8),C0:DRAW "C0"+MOUTH$
    :SOUND 32767,0:NEXT:PSET (X*14,Y*8
    ):DRAW "C3"+MOUTH$
830 PLAY "MB T255O2MSA4O3C4D8P8P4D8P8P4
    P2 O2A4O3C4D8P8P4D8P8P4 P2 O2G4A#4
    O3D8P8P4D8P8P4 P2 E4E4D8" ' MAC THE
    KNIFE
840 DEAD=DEAD+1:GOTO 380
850 FOR R=1 TO 3:PSET(D(R,1)*14,D(R,2)*
    8),0:DRAW "C0"+CN$:NEXT
860 PSET(X*14,Y*8),0:DRAW "C0"+EX$:PSET
    (X*14,Y*8),0:DRAW "C3"+POT$
870 PLAY "O3C2O2B4O3D4C4O2A4O3F2" ' **SM
    OKE GETS IN YOUR EYES
880 FOR T=0 TO 1300:NEXT
890 DEAD=DEAD+1:PSET(X*14,Y*8),0:DRAW "
    C0"+POT$:GOTO 380
900 IF INKEY$="" THEN 900 ELSE CLS:RETUR
    N ' **TO CONTINUE
```

IBM PC & PCjr

```
910 IF DEAD>SAFE THEN 930:LOCATE 12,57:
    PRINT " ***** YOU WIN *****"
920 LOCATE 14,6:PRINT "WELCOME ABOARD,
    EXPLORER...":LOCATE 15,5:PRINT "WHAT
    'S OUR NEXT DESTINATION?":GOTO 940
930 LOCATE 12,33:PRINT " ** YOU LOSE **
    :LOCATE 14,25:PRINT " PRESS ANY KE
    Y TO RESTART"
940 IF INKEY$="" THEN 940 ELSE 210
950 A$="CANNIBALS:"+STR$(DEAD):PX=1:PRI
    NT A$:A$="EXPLORER:"+STR$(SAFE):PX=
    2:PRINT A$ ' SCORE DISPLAY
960 FOR TD=1 TO 3000:NEXT:LOCATE 1,1:FO
    R X=1 TO 160:PRINT CHR$(32);:NEXT
970 RETURN
980 K$=INKEY$:IF K$="" OR LEN(K$)=1 THE
    N RETURN ' KEYBOARD CHECK
990 K=ASC(RIGHT$(K$,1)):DEF SEG=0:POKE
    1050,PEEK(1052)
1000 DX=(K=75)-(K=77):DY=(K=72)-(K=80)
1010 X1=X+DX:Y1=Y+DY
1020 IF Y1<1 OR Y1>18 THEN RETURN
1030 IF X1<0 THEN X1=22:ELSE IF X1>22 TH
     EN X1=0
1040 IF P(X1,Y1) THEN RETURN
1050 PSET (X*14,Y*8),C0:DRAW "C0"+EX$:PS
     ET (X1*14,Y1*8),C3:DRAW "C2"+EX$
1060 X=X1:Y=Y1:RETURN
1070 SX=STICK(0):SY=STICK(1):DX=(SX<AX-4
     )-(SX>AX+4):DY=(SY<AY-4)-(SY>AY+4):
     GOTO 1010
```

HCM

Cannibals . . . from p. 59

CANNIBALS (C-64)
Explanation of the Program

Line Nos.

100-160	Header.
170-230	Move character definitions to RAM.
240-280	Read through all DATA statements to store restore variables.
290-320	Read programmable character definitions.
330-440	Read sprite definitions.
450-490	Initialization.
500-580	Main program loop.
590-880	Subroutine to move cannibals.
890-1010	Subroutine to get inputs from keyboard or joysticks.
1020-1030	Subroutine to find random position on upper part of game screen.
1040-1050	Subroutine to find random position on lower part of game screen.
1060-1100	Subroutine to position ponds on screen.
1110-1140	Subroutine to position ocean waves on screen.
1150-1180	Subroutine to place boat at random position.
1190-1230	Subroutine to place cannibals on screen.
1240-1270	Subroutine to place man on screen.
1280-1570	Subroutine to move man on screen in response to player input.
1580-1620	Secondary sound routine.
1630-1700	Subroutine to restore DATA statement pointers.
1710-1800	Primary sound subroutine.
1810-1850	End of game sequence subroutine.
1860-2040	DATA statements containing notes for program tunes.

COMMODORE 64

```
100 REM ***********
110 REM * CANNIBALS *
120 REM ***********
130 REM BY CARL CARROZZA AND THE HCM ST
    AFF
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM C64 BASIC
170 POKE 52,56:POKE 56,56:CLR
180 POKE 56334,PEEK(56334)AND254
190 POKE 1,PEEK(1)AND251
200 FOR I=0 TO 511:POKE I+14336,PEEK(I+
    53248):NEXT
210 POKE 1,PEEK(1)OR4
220 POKE 56334,PEEK(56334)OR1
230 POKE 53272,(PEEK(53272)AND240)OR14:
    POKE 53281,1
240 T=0:R=0
250 DIM D1(5),D2(5),D3(5),D4(5),MN(1,2)
    ,CN(4,2)
260 READ A:IF A=-100 THEN GOSUB 1630
270 IF A<>-333 THEN 260
280 RESTORE
290 FOR I=14848 TO 14847+(6*8):READ A:P
    OKE I,A:NEXT
300 DATA 255,255,255,255,255,255,255,25
    5,255,255,255,239,215,187,125,254
```

COMMODORE 64

```
310 DATA 0,60,126,255,255,126,60,0,88,8
    9,45,10,24,40,46,98
320 DATA 25,26,20,112,220,20,116,70,24,
    24,8,255,8,255,126,126
330 FOR I=15360 TO 15359+(3*64)
340 READ A:IF A<>-444 THEN 360
350 READ A:FOR J=I TO I+A-1:POKE J,0:NE
    XT:I=I+A-1:GOTO 370
360 POKE I,A
370 NEXT
380 DATA -444,21,1,131,0,1,131,0,1,131,
    0,15,255,128,15,231,224
390 DATA 26,165,96,255,255,254,127,255,
    252,63,255,252
400 DATA 31,255,252,-444,13
410 DATA -444,21,192,0,0,224,0,0,240,0,
    0,248,0,0,252,0,0,110,0,0
420 DATA 63,0,32,31,128,96,15,192,224,-
    444,16
430 DATA -444,21,192,0,0,224,0,0,240,0,
    0,248,0,0,124,0,0,46,0,0
440 DATA 31,0,32,255,128,96,15,192,224,
    -444,16
450 INPUT "■CTRL BLK■■ SHIFT CLR■■USING J
    OYSTICKS? ENTER Y OR N";J$
460 DV=1:S=54272:LV=0
470 FOR I=S TO S+24:POKE I,0:NEXT
480 PRINT "■SHIFT CLR■":POKE 53281,1
490 FOR I=1824 TO 2024:POKE I,64:POKES+
    I,4:NEXT
500 GOSUB 1060
510 GOSUB 1110
520 GOSUB 1150
530 GOSUB 1190
540 GOSUB 1240
550 GOSUB 890
560 RX=0:GOTO 1280
570 GOTO 590
580 GOTO 550
590 SA=(MN(1,1)*40)+983+MN(1,2)
600 MY=INT((SA-984)/40):MX=SA-983-(MY*4
    0)
610 FOR I=1TO 4:CT=0
620 IF MY>CN(I,1) THEN Y1=1:GOTO 650
630 IF MY<CN(I,1) THEN Y1=-1:GOTO 650
640 Y1=0
650 IF MX>CN(I,2) THEN X1=1:GOTO 680
660 IF MX<CN(I,2) THEN X1=-1:GOTO 680
670 X1=0
680 SS=((CN(I,1)+Y1)*40)+983+CN(I,2)+X1
    :NP=PEEK(SS):CT=CT+1
690 TP=(CN(I,1)*40)+983+CN(I,2)
700 IF NP<>32 THEN 740
710 CN(I,1)=CN(I,1)+Y1:CN(I,2)=CN(I,2)+
    X1
720 POKE TP,32:POKE SS,67:POKE S+SS,0
730 GOTO 870
740 IF NP<>66 AND NP<>67 THEN 830
750 ON CT GOTO 760,770,780,790,800,810,
    820
760 Y1=Y1:Y1=0:GOTO 680
770 X1=X1:Y1=Y1:X1=0:GOTO 680
780 X1=X1:Y1=Y1:Y1=Y1+1:GOTO 680
790 Y1=Y1:X1=X1:X1=X1+1:GOTO 680
800 X1=X1-1:GOTO 680
810 X1=X1:Y1=Y1-1:GOTO 680
820 GOTO 870
830 IF NP<>68 AND SA<>SS THEN 870
840 POKE SS,69:POKE S+SS,9:CN=CN+1
850 T=1:R=4:GOSUB 1630:D=1:DV=2:GOSUB 1
    710
860 GOTO 1810
870 NEXT I
```

Continued on p. 72

Cannibals. . . from p. 71

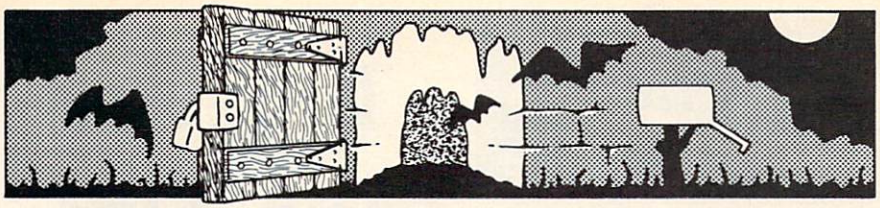
COMMODORE 64

```
 880 GOTO 580
 890 X1=0:Y1=0
 900 IF J$="N" THEN 960
 910 XT%=PEEK(56320)AND31
 920 X1=SGN(XT% AND 4) - SGN(XT% AND 8)
 930 IF X1<>0 THEN 950
 940 Y1=SGN(XT% AND 1) - SGN(XT% AND 2)
 950 RETURN
 960 KY=PEEK(197):IF KY=64 THEN RETURN
 970 IF KY=13 THEN X1=-1:RETURN
 980 IF KY=18 THEN X1=1:RETURN
 990 IF KY=14 THEN Y1=-1:RETURN
1000 IF KY=23 THEN Y1=1:RETURN
1010 RETURN
1020 CL=INT(RND(0)*40)+1:RW=INT(RND(0)*20)+1
1030 SA=983+CL+(40*RW):RETURN
1040 CL=INT(RND(0)*40)+1:RW=INT(RND(0)*5)+1
1050 SA=1784+CL+(40*RW):RETURN
1060 IF LV>9 THEN 1100
1070 FOR I=1 TO INT(ABS(10-LV))*10
1080 GOSUB 1020:Z=PEEK(SA):IF Z<>32 THEN 1080
1090 POKE SA,66:POKE S+SA,4:NEXT
1100 RETURN
1110 FOR I=1 TO 25
1120 GOSUB 1040:Z=PEEK(SA):IF Z<>64 THEN 1120
1130 POKE SA,65:POKE SA+S,4:NEXT
1140 RETURN
1150 BP=INT(RND(0)*265)+24:POKE 53269,0:POKE 53264,0:POKE 53249,206
1160 BZ=BP:IF BP>255 THEN BP=BP-255:POKE 53264,1
1170 POKE 53248,BP:POKE 53287,1:POKE 53277,1:POKE 2040,240:POKE 53269,1
1180 RETURN
1190 FOR I= 1 TO 4
1200 GOSUB 1020:Z=PEEK(SA):IF Z<>32 THEN 1200
1210 POKE SA,67:POKE S+SA,0
1220 CN(I,1)=RW:CN(I,2)=CL:NEXT
1230 RETURN
1240 GOSUB 1020:Z=PEEK(SA):IF Z <> 32 THEN 1240
1250 POKE SA,68:POKE SA+S,6
1260 MN(1,1)=RW:MN(1,2)=CL
1270 RETURN
1280 TP=(MN(1,1)*40)+983+MN(1,2)
```

COMMODORE 64

```
1290 SA=((MN(1,1)+Y1)*40)+983+MN(1,2)+X1:NP=PEEK(SA)
1300 IF NP<>66 AND SA>1023 THEN 1320
1310 GOTO 1560
1320 MN(1,1)=MN(1,1)+Y1:MN(1,2)=MN(1,2)+X1
1330 IF NP<>32 THEN 1350
1340 POKE TP,32:POKE SA,68:POKE S+SA,6:GOTO 1560
1350 IF NP<>67 THEN 1380
1360 POKE TP,69:POKE S+TP,9:T=1:R=4:GOSUB 1630:D=1:DV=2:GOSUB 1710
1370 CN=CN+1:GOTO 1810
1380 IF NP<>64 AND NP<>65 THEN GOTO 1560
1390 W1=INT((SA-1023)/40):WP=(SA-1023)-(W1*40)
1400 BZ=INT(BZ/8)-2
1410 IF WP<BZ OR WP>BZ+5 THEN 1450
1420 POKE TP,32:POKE SA,67:POKE S+SA,6
1430 R=2:T=1:GOSUB 1630:D=1:DV=1:GOSUB 1710
1440 LV=LV+1:GOTO 1810
1450 W1=INT((SA-1023)/40):WP=(SA-1023)-(W1*40)
1460 SP=WP*8+24:IF SP>255 THEN POKE 53264,PEEK(53264)OR2:SP=SP-255
1470 POKE 53250,SP:POKE 53251,206:POKE 53288,1:POKE 53277,PEEK(53277)OR2
1480 POKE 2041,241:POKE 53269,PEEK(53269)OR2
1490 R=3:T=1:GOSUB 1630:DV=2:D=1:GOSUB 1710
1500 FOR R=1TO10:IF HF=2 THEN HF=37:GOTO 1520
1510 HF=2
1520 IF PEEK(2041)=241 THEN POKE 2041,242:GOTO 1540
1530 POKE 2041,241
1540 GOSUB 1580:NEXT
1550 CN=CN+1:GOTO 1810
1560 RX=RX+1:IF RX=2 THEN 570
1570 GOTO 1280
1580 POKE S+5,0:POKE S+6,249
1590 POKE S+1,HF:POKE S,163
1600 POKE S+4,129:FOR ZX=1TO150:NEXT
1610 POKE S+4,128
1620 RETURN
1630 IF T=1 THEN 1680
1640 R=R+1
1650 D1(R)=PEEK(63):D2(R)=PEEK(64)
```

Continued on p. 78



ZORK

A Review by HCM Staff with Randy Wilson

Name:	**Zork I**
Program Type:	**Adventure Game**
Machine:	**Apple IIe, C-64, IBM PC, TI-99/4A**
Distributor:	**Infocom 55 Wheeler St. Cambridge, MA 01238**
Price:	**$39.95, diskette**

	poor	fair	good	excellent
Performance				
Engrossment				
Documentation				

Zork is a classic text adventure for Apple, Commodore, and IBM computers, among others, and may well be the most popular adventure game yet invented. It takes you through a forest, a mansion, a vast cavern complete with maze, the entrance to Hades, a dam, a river, and more. You have to fight a host of enemies in the Great Underground Empire.

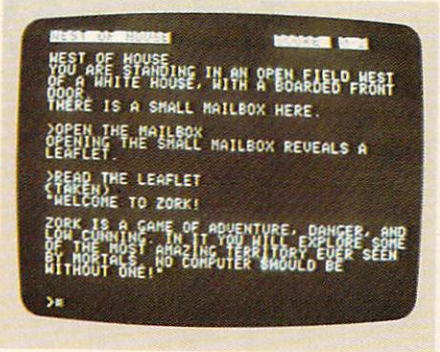
The game is not only ancient in its setting and storyline, it dates from a relatively early period in computing. *Zork* originated on the DEC PDP-10, the combined effort of Tim Anderson, Marc Blank, Bruce Daniels, and David Lebling. For years it was passed back and forth among programmers, long before computers became common household objects.

From the beginning, *Zork*'s programmers were careful to provide flexibility of input. Players could immerse themselves in the scenario; they didn't have to "talk" to the computer in unnatural phrases that could detract from the fantasy. Even today most adventure games operate on two-word commands from the player, for example: "GET SWORD", "TAKE LAMP", "GRAB ROPE"; but *Zork* can handle all three objects at once, e.g., "TAKE THE SWORD, LAMP AND ROPE", or "TAKE ALL". You can even list several complete commands on a single line, as long as you separate them with periods: "GET ALL EXCEPT THE ROPE AND THE PLASTIC. LIGHT LAMP. W." (west) "KILL THE TROLL WITH THE SWORD." Note that the article "the" used with commands does not throw off the game's logic, and it can help the imagination fly more freely.

But there must be more to it than that; a program doesn't become a hit just because it's comfortable to "talk" to. Zork's real drawing card is its story line—with its endless intrigues and surprises. The vast underground labyrinth is filled with treasure and populated with bizarre characters, most of them dangerous. More important, the elements of play are well-integrated. That is, as the game evolved, each new object or action was thoroughly analyzed to see that it was compatible with the rest of the plot. Thus, the game's action is logical, and there isn't a lot of clutter. Whatever the object or the event you must deal with, the program responds in a sophisticated, "knowledgeable" fashion.

Among some other little luxuries of this game are displays of your current score and number of moves, along with the room you are in. Most adventure players enjoy building a score as a reward for their fighting and treasure-finding skills, as opposed to a simple live-or-die contest.

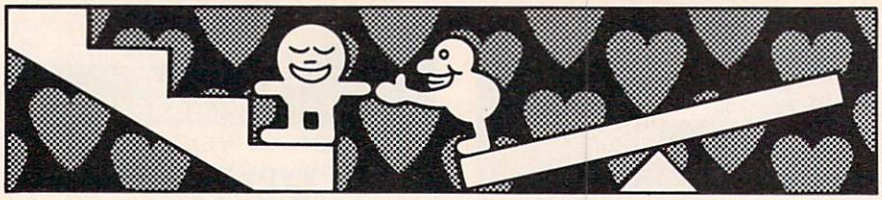
In addition to its friendly input structure, *Zork* also accepts a long list of commands that keep you well-informed of your progress. The command BRIEF, for example, describes any newly-encountered rooms or objects, whereas VERBOSE shows you a long description of the objects in any room you enter. Naturally, there are commands for SAVing the game, RESTARTing it, taking INVENTORY of your possessions, plus a dozen other useful commands.



The game comes with an excellent little booklet that tells you how to use the commands, save your game, and fight your enemies—without giving away any secrets that could spoil the challenge. If you get completely bogged down, you may be interested in seeking help from certain maps and booklets sold by Infocom. An order form is included in the package.

The traditional text-only adventure does have its good points. Not obliged to paint pictures, it can devote all its bytes to telling the story, and it processes the plot quicker too, letting you cover ground faster. And there is that matter of letting your mind wander freely. Yes, a good case can be made for the text adventure, and nobody makes a stronger case than *Zork*.

HCM



Tag Tom and Fire

A Review by Greg Roberts

HCM Staff

A few decades ago, in a darker age of education, most children were not given ample resources for developing their minds until they reached first grade. Now we find that the two-year-olds are capable of much more than we gave credit for. It turns out that they love learning, and they have no time for anyone who would deny and deaden their fine curiosity. Fortunately, the adult world is finally responding to the children with an outpouring of "read-early" books, television programs such as *Sesame Street*, and a widespread interest in Suzuki-style music education. And yet, the computer world has lagged in meeting the needs of pre-school children. Very little software has been designed for this group—and the truly good programs are as spotty as the selection of bordeaux in an all-night mini-market.

Tag Tom and *Fire* are packaged on a single diskette for pre-school children. Each game is designed to be entertaining, while teaching spatial relationships necessary for reading. For example, children must understand right and left before they can be expected to follow words printed on a page. This basic concept, plus shape recognition, sequencing, hand/eye coordination, and opposites are some of the considerations that went into the design of these programs.

Tag Tom

Tag Tom's basic scenario is to make one figure tag another on the playground and around the house. The game offers four variations in play. On the simplest level, a touch of the keyboard is the only input required. No matter what key you press, your man Timmy goes right over to tag Tom. The menu offers a second version that is faster than the first game, and it too is meant only for the very young.

The joystick mode is more challenging. You have a limited time to direct Timmy over to Tom. A speeded-up version of the joystick game is your fourth option. In all four levels of play, the screen flashes one-

---

Name: Donkey Kong
Program Type: Arcade game
Machine: TI-99/4A
Distributor: Atarisoft
Atari, Inc
P.O. Box 61657
Sunnyvale, CA 94088
Price: $34.94, cartridge
$44.95, diskette

System Requirements:
Joysticks

	poor	fair	good	excellent
Performance				
Ease of Use				
Documentation				

Donkey Kong

A Review by Greg Roberts

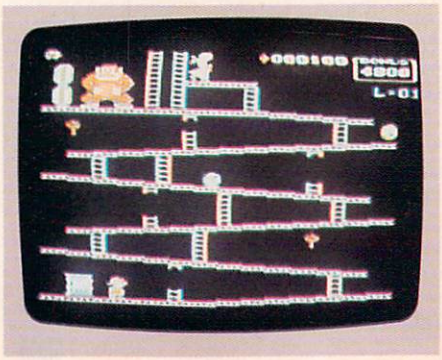
HCM Staff

This game is one of ten cartridges now manufactured by Atari for the TI-99/4A. The original *Donkey Kong* is a landmark, a fundamental institution of the arcade game cult; therefore, any new version will be scrutinized to determined how close it is to that revered icon known as the Arcade Original.

This one is very close. The programmers have meticulously constructed the framework girders and rolling barrels to look and act as they do in the original game. The barrels move in the same random fashion. They are deadly and unpredicitable in the extreme.

No matter what your version of *Donkey Kong*, learning the basic barrel-jumping maneuver can be frustrating to a beginner. The player has to press the fire button and move the joystick handle in unison at the precise moment to carry out a successful leap. To get much out of this game you have to be willing to practice a bit. You also need a good responsive joystick. I found the play almost impossible with the TI joystick. My score went up only after I plugged in a third-party joystick which was clearly more responsive weapon.

Once you get used to the jump maneuver, you are in for a game that is not likely to get dull for many, many plays. The scenario is uncomplicated except that a barrel reaching the end of the track turns into a fireball that can move around and kill you. And there is the option of reaching up and grabbing a hammer that quickly smashes the approaching barrels. Moving your man through the barrel-strewn obstacle course to the top of the girders, you proceed to a second level with new challenges, including gaps in the framework and more fireballs. These last are so overwhelming, I found them impossible to jump over. But this must be my lack of skill—other players faced this challenge with delight. The upper levels of this game, with their fast-moving elevators and conveyer belts, look impossible too.



My overall impression is that this is a major-league game for serious videacs, and one which could permanently frustrate some people. This is a matter of personal preference, of course. A good player could easily argue that anything less than *Donkey Kong* is boring baby stuff. However you perceive the game, its large variety of screens is the mark of a top-flight program.

Donkey Kong comes with a colorful, but vague brochure. It alludes to the dangers our hero must confront in order to rescue his girlfriend, and lets you know the point value of various achievements within the game, but there are few hints on strategy. This leaves much to be discovered through trial and error, a basic tenet of sophisticated games design, and one that is meant to keep you playing for a long, long time.

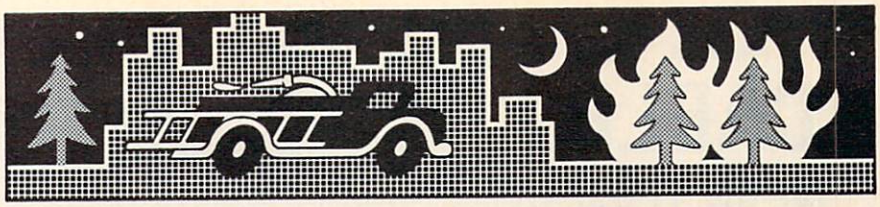
If you are looking for an arcade atmosphere in your own home, all you need is the dim lights, cigarette smoke, and a few loiterers. The TI-99/4A *Donkey Kong* very nicely provides the rest.

HCM

Name:	Tag Tom and Fire
Program Type:	Early Learning
Machine:	TI-99/4A
Distributor:	Software Advances P.O. Box 11409 Eugene, OR 97440
Price:	$15.95, diskette, cassette

System Requirements:
Extended BASIC, Joystick recommended

	poor	fair	good	excellent
Performance				■
Engrossment			■	
Documentation				■



word descriptions of Tom's location within the scene—up, down, upon, etc. These words may be heard through a speech synthesizer, but the silent version also does a thorough job of teaching and entertaining. The movements of the two figures are accompanied by cheerful sound effects that should appeal to any child.

Fire

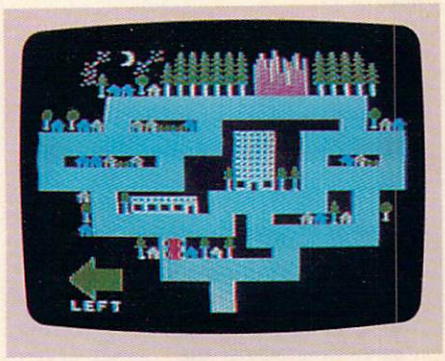
On the same diskette is *Fire*, a colorful little maze game that asks the player to direct a fire truck through the streets to a blaze at the top of the display. On its lowest level, you simply press a key on the right or left side of the keyboard to send the fire engine to its destination. A more complex option uses up and down arrows. Finally, you can choose joystick mode to guide the fire truck, again using all four directions.

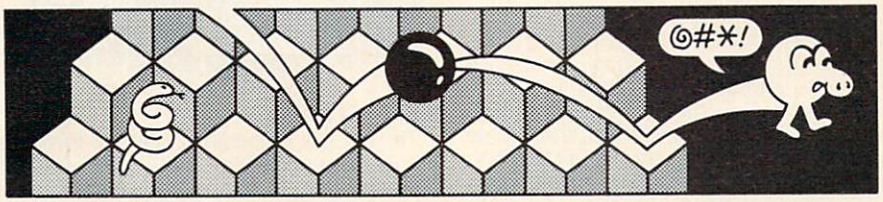
This program provides stick-on arrows for the keyboard. The arrows come in four colors, corresponding to the arrows that come up on the screen showing the proper direction to guide the truck. It is a good exercise for giving very young children some experience with basic directions.

Both games use simple graphics showing a colorful, albeit static background with only one moving object at a time. A little more action would probably not be lost on a child, but the existing graphics certainly do captivate the typical pre-schooler. Both games will be quickly outgrown, but that is a quality of most playthings for this fast-growing age group. And that is why a package like this has been priced reasonably at $15.95.

These games come with thorough documentation, including an explanation of the educational concepts behind their design. They are simple games, of course, and would not hold much interest for anyone but a child—but that is what makes them special. *Tag Tom* and *Fire* are worthy offerings for a previously neglected group of computer users.

HCM





Q*bert

A Review by Joan Killough-Miller

HCM Staff

Name:	Q*bert
Program	Arcade Game
Machine:	TI-99/4A, Apple II series, Atari, C-64 and VIC-20
Distributor:	Parker Brothers P.O. Box 1012 Beverly, MA 01915
Price:	$39.95, cartridge

System Requirements:
Joystick

	poor	fair	good	excellent
Performance			■	
Engrossment				■
Documentation			■	

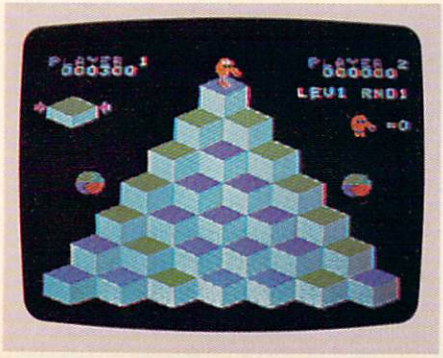
For those of us who aren't ready to be hurtled through a universe of exploding asteroids or crammed into the cockpit of a destroyer weapon and charged with the fate of the universe, there is *Q*bert*. Home computer owners who've come to know *Q*bert* in the arcade finally have their own version to bring home for the family.

*Q*bert* is a shy, lovable little character who moves in quick, nervous hops down the steps of a three-dimensional pyramid of stacked cubes that resembles a "baby's blocks" patchwork quilt. His mission is to change the color of every block by hopping on it. Life is out to get *Q*bert*, it seems, and hazards come from all directions. You have to be quick on the joystick to avoid the red and purple balls that come rolling after him. If a purple ball reaches the bottom, it uncoils into Coily, a bouncing snake who pursues *Q*bert* with zeal. In the later rounds, other demons maneuver to ambush him and undermine his work by changing the blocks back to their starting color.

Oh, @!#?!!

*Q*bert* works with dogged patience: When squashed by a ball or jumped by the snake, he utters an innocent oath— @!*?!!— and goes back to work. The round ends when you have completed the pyramid or used up all of your allotted "*Q*berts*." But *Q*bert* is not without resources. He can hop a pinwheel-patterned disc that twirls him back to the top of the pyramid. In a rather cunning ploy, he can jettison the snake by luring him to the edge of the pyramid and jumping a disc at the last minute, a routine reminiscent of the old Roadrunner cartoons. If *Q*bert* can catch a green ball, the screen freezes for a moment, and our timid hero can grab some quick action while the enemies are paralyzed.

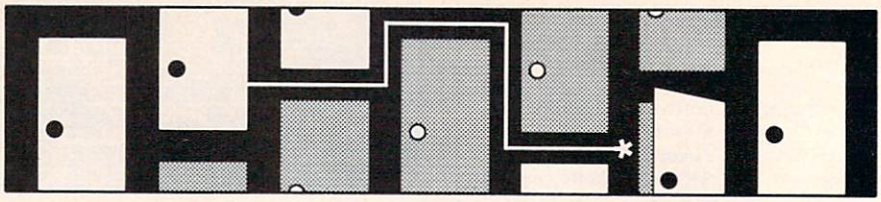
For a simple, non-aggressive game, *Q*bert* builds rapidly in speed and complexity as the enemies become faster and more plentiful. Although all of the screen displays are basically the same (only the cube colors change), new creatures and challenges lurk on every level.



We play-tested Q*bert extensively on the TI-99/4A. With the machine's fast 16-bit microprocessor, the action was very fast, and the joystick was *very* responsive—perhaps even a bit too sensitive, as an overly enthusiastic flick will send *Q*bert* toppling off the edge of the pyramid. It may take a while to fine tune your touch on the joystick and learn how and when to jump the flying discs.

There is a bit too much predictability in the course of the balls and timing of the enemies—you will quickly learn a set of stock maneuvers to dupe the snake and ditch the demons. But having the preliminary rounds under control is an advantage here, because the TI version of *Q*bert* sends you back to the very beginning every time you deplete your reserve of *Q*bert* figures. After you've finally reached Level 2 and found a whole new dimension (you now have an intermediate color as well as a destination color, so that each square must be jumped on twice), you may resent having to plod through the lower levels all over again—or you may see it as useful practice that sharpens your reflexes for the faster,

Continued on next page



TI-MAZE

A Review by G. R. Michaels

HCM Staff

Name:	TI-MAZE
Program Type:	3-dimensional maze game
Machine:	TI-99/4A
Distributor:	Gadget Software Post Office Box 98 Port Moody, B.C. V3H 3E1
Price:	$16.95 US, cassette $19.95 Canada, cassette

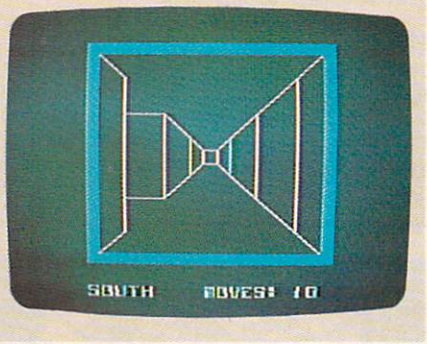
System requirements:
Cassette Recorder

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

Do some people possess an innate sense of direction? Experiments with a simple blind fold have shown that, so far, the answer is no. A certain few, however, have developed their powers of observation to find their way out of the most bewildering terrain. If you take pride in keeping your wits about you as you wander through seemingly endless passages, you will take to the challenge of *TI-MAZE*.

This program can generate ten trillion (10,000,000,000,000) mazes. You help determine the configuration of each new maze by entering a "lucky number" into the computer. Select the level of difficulty by specifying the maze's dimensions—its width and length as measured on a scale of 2–10. The maze is presented as a series of rather spare images made up of vertical and horizontal lines, but clearly simulating a corridor, including side passages and a distant wall. For this review we used a TI monitor, model PHA4100A, but the game can also be played on a standard television set, in spite of a slight degrading of the resolution.

My first effort was a medium-sized 6×8 rectangle. Picking my way through the corridor with single moves of the joystick, I was further guided by a screen display that gave me my direction (N, S, E, W), and by a map of the path I'd already traveled (revealed with a touch of the fire button). Nevertheless, after dozens of moves, I could not find the door. Abandoning that game, I tried a baby-level 2×3 rectangle and got out in six moves. Clearly, unless you have the homing resources of a Daniel Boone or an abandoned cat, you will save much frustration by choosing a postage stamp for your first attempt, just to get your confidence up.



Spare Graphics

The intrigue of the game, then, is in your mental involvement with the maze problems—not in fast-paced action or fancy graphics. The graphics have been kept very simple because, with each move, the computer must display a new picture to simulate your advancement through the corridors. As it is, each new scene appears rather slowly, even with the sparest of stage props. This is a characteristic of TI BASIC, and nothing can be done about it.

This is a large program. It will run only on a skeleton system made up of console, monitor, and cassette deck. unless other peripherials are turned off, any attempt to load TI-MAZE will result in a loading error.

The game must be loaded in two parts. On one side of the cassette tape is the TI-MAZE game itself; on the other side is the MASTER DATA. After you load the first side of the tape and type RUN, the program

*Q*bert* . . . from p. 75

more intricate levels, where jumping on the intermediate or destination colors will undo what *Q*bert* has already accomplished.

Standard Documentation

Parker Brothers' documentation covers the basics: a rundown of the various demons, the point value system, and the function of the flying discs. A clear and concise chart shows you the precise results of hopping on the starting, intermediate, and destination colors on each of the nine levels, although this information is difficult to retain in the heat of the game.

*Q*bert* offers a lot of flexibility for such a simple game. You can play an opponent, or play alone. You can even engage a foursome and alternate turns with a partner. You can play for mere survival, or you can study the manual and play for points: Conserving the flying discs, catching the green ball, and throwing the snake all have their rewards in point values. The game accommodates a variety of personal styles and allows players to develop unique strategies.

*Q*bert* has a lot to offer on all levels: Novices, children, and arcarde wizards can get started without a lot of explanation and stay with it for a long time. There's plenty of room for skill development in *Q*bert*, and you'll soon find yourself cursing along with the little orange guy, and doing whatever you can to ease his way in a world of continuous bombardment and slithering evil, where every harried step seems crucial. Don't we all have days like that? HCM

automatically goes to its own version of the familiar OLD CS1 routine necessary to add on the MASTER DATA. This procedure may seem a bit cumbersome to those who are used to cartridges. But detailed loading instructions are provided in a litttle booklet that also explains how to play, whether with joystick or using the arrow keys.

In some places the game could have benefited from more intense graphics. For example, painting the exit door a dramatic glowing shade would have made it a richer reward (after all, reaching it can take forever). Nevertheless, the present simple outline does serve its purpose.

The understated nature of this game may not appeal to players looking for multicolored moonscapes and Pac tracks; but those who like to work out a complex, long-lasting puzzle will find plenty to absorb them here. With ten trillion possibilities offered in one game, *TI-MAZE* fans will get plenty of play for the money. HCM

Mouser . . . from p. 65

new (or incompetent) players who get tired of having to lose and restart the game every three minutes. And there are two slight bugs that should be corrected: If a wall hits a mouse as it is swinging shut, the wall is partially obscured on the screen. So even after the mouse moves away, the wall appears as a fragmented collection of dots, rather than a solid wall. It does, however, return to its solid shape after you swing it again. The second flaw occurs if a mouse catches the farmer at the spot where he normally enters the first screen (approximately the nine o'clock position at far left). On each game's new farmer, the mouse is right on the spot—ready to pounce—and finishes the poor guy off each time before you can move him.

Punishing Staying Power

The documentation discusses the rules, obstacles, and moves of the game; it also offers a few helpful hints for improving your score. Although adequate, we feel that with some additional instruction on the "fine points of play"—i.e., staying alive a little longer—the game would be more appealing and less frustrating for more players.

We all know that a game's superior graphics and sound "don't mean a thing if it ain't got that swing." But *Mouser* does. It is truly a fun and challenging game. Its main action—manipulating characters to trap moving objects—is a refreshing change from the many shoot-'em-ups on the market And because *Mouser* is so difficult, it has staying power. After several rounds without capturing even one mouse, I was frustrated but I couldn't put down the joystick. *Mouser* has that undefinable quality that keeps drawing you back for more punishment. With constant electrical failures and mice that multiply like rabbits, you could be down on Farmer Wheatbread's farm for a long, long time. HCM

2049'er . . . from p. 57

offers new backdrops and new challenges.

For example, after negotiating the first level with its complex arrangement of ladders and shafts, you find that the second level features transporters capable of firing your man to various floors within the strata of the screen. When Bounty Bob enters his Star-Trek type "beamer" and shoots to the top of the tube, you witness some unparalleled graphics.

Farther along, you run into tanks of radioactive waste, pulverizing devices, and a cannon you can load with TNT in order to shoot Bounty Bob to glory. With so much to see, it is no wonder that Tigervision provides us, at the beginning of the game, with a sneak preview of all eight levels. They are a real draw to the prospective buyer. The complex moves of this well-designed cartridge game look good on an ordinary color television, as well as on a dedicated monitor.

A few elements of this game, however, are a little frustrating: The mine shafts or slides don't line up well with their entrances, so it is difficult to tell when your man is at the edge of a shaft. He appears to walk in space, then fall. This lack of precision is a real distraction that adds nothing to the game. In fact, it arbitrarily ruins the player's strategy. Nevertheless, once you establish where the shaft entrances begin and end, you can resume play and really enjoy this game.

Miner 2049'er's sound effects are typical computer Blippenese, but they are well integrated with the play. The only aural bugbear in this program is the primitive Clementine melody played during the sneak preview of the screens, and it can be mercifully squelched with the touch of a button.

The game comes with a detailed brochure that explains each screen, with a few hints on playing. The documentation is vital in a game as complex as this one.

It comes as no surprise that *Miner 2049'er* is available for every major home computer. It certainly deserves a wide audience. HCM

Cannibals . . . from p. 72

COMMODORE 64

```
1660 D3(R)=PEEK(65):D4(R)=PEEK(66)
1670 GOTO 1700
1680 POKE 63,D1(R):POKE 64,D2(R)
1690 POKE 65,D3(R):POKE 66,D4(R)
1700 RETURN
1710 READ A:IF A=99 THEN 1800
1720 IF A=-100 THEN 1710
1730 READ F:IF F=0 THEN HF=0:LF=0:GOTO 1
     750
1740 F=INT(F/DV):FD=INT(F/.06097):HF=INT
     (FD/256):LF=FD-(256*HF)
1750 POKE S+5,169:POKE S+6,179
1760 POKE S+1,HF:POKE S,LF
1770 POKE S+24,15:POKE S+4,33
1780 FOR ZX=1TO(200*D):NEXT
1790 POKE S+4,32:GOTO 1710
1800 RETURN
1810 PRINT"■SHIFT CLR■2CRSRDOWN■■CRSRRI
     GHT■LIVINGSTON=";LV
1820 PRINT"■CRSRDOWN■■CRSRRIGHT■CANNIBAL
     S=";CN
1830 PRINT:PRINT:PRINT:PRINT "< PRESS AN
     Y KEY TO CONTINUE >"
1840 Z=PEEK(197):IF Z=64 THEN 1840
1850 POKE 53269,0:GOTO 480
1860 REM SOUND1 /2.7 CANNIBALS THEME S
     ONG
1870 DATA -100,1,1109,1,1661,1,1397,.5,1
     245,.5,1397,2,1109,2,831,1,0
1880 DATA 1,1109,1,1661,1,1397,.5,1245,.
     5,1397,2,1109,2,831,1,0
1890 DATA 1,1109,1,1397,.5,1245,.5,1109,
     2,1245,2,831,1,0
1900 DATA 1,1109,1,1397,.5,1245,.5,1109,
     2,1245,2,1661,1,0,99
1910 REM SOUND2 /1 LIVINGSTON MAKES
     IT BACK TO BOAT
1920 DATA -100,1,932,1,932,2,1047,1,1047
     ,1,1047,1,1047,2,1047,1,1397
1930 DATA 1,1245,1,1175,2,932,1,0,99
1940 REM SOUND3 /2 MACK THE KNIFE
1950 DATA -100,1,1109,1,1319,1,1480,1,0,
     3,1480
1960 DATA 1,1109,1,1319,1,1480,1,0,3,148
     0
1970 DATA 1,1109,1,1319,1,1480,1,0,3,148
     0
1980 DATA 1,1661,1,1661,2,1480,99
1990 REM SOUND4 /2 SMOKE GETS IN YOU
     R EYES
2000 DATA -100,2,1109,1,1047,1,1245,1,11
     09,1,932,2,1480
2010 DATA 1,1397,1,1661,1,1480,1,1245,2,
     2218,99
2020 REM SOUND5 /1 SHARK CHOMPING
2030 DATA -100,2,880,2,220,2,880,2,220,2
     ,880,2,220,2,880
2040 DATA 2,880,2,220,2,880,2,220,2,880,
     2,220,2,880,99,-333
```

HCM

CANNIBALS (Apple)
Explanation of the Program

Line Nos.

100-160	Program Header.
170	Protect hi-res screen.
180-220	Poke Line# RESTORE routine.
230-240	GOSUBs to POKE Shape Table and Music routine.
250	GOSUBs to display title screen and play title tune.
260	Joystick or keyboard control determined.
270-300	POKE score and control method variable and CLEAR all variables; then PEEK control method variable back into J$.
310-340	DIM arrays and initialize variables.
350	GOSUB to draw main screen
360-610	Get player's input and set direction of movement.
620-660	Move player.
670-920	Move cannibals and check for action.
930-1020	Eliminate person and check for replay.
1030-1060	Display final score and END program.
1070-1150	Player goes into water; see if win or lose.
1160-1180	Keyboard input routine.
1190-1510	Draw main screen and place characters routine.
1520-1590	Play title tune routine.
1600-1630	Display title screen routine.
1640-1690	POKE Shape Table routine.
1700-1760	POKE Music routine.
1770-1840	Play Shark routine.
1850-1910	Play Smoke routine.
1920-1980	Play Columbia routine.
1990-2010	Make sound routine.
2020-2060	Line # RESTORE link routine.

APPLE II Series

```
100 REM  *********
110 REM  * CANNIBALS *
120 REM  *********
130 REM  BY CARL CARROZZA AND THE HCM S
    TAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  APPLE II SERIES APPLESOFT
170 IF PEEK (103) < > 1 OR PEEK (104
    ) < > 64 THEN POKE 103,1: POKE 10
    4,64: POKE 16384,0: POKE 16385,0: P
    OKE 16386,0: PRINT CHR$ (13) CHR$
    (4)"RUN CANNIBAL"
180 REM  POKE RESTORE PROGRAM
190 HIMEM: 36096: FOR I = 36096 TO 36623
    1: READ PV: POKE I,PV: NEXT
200 DATA 0,0,131,0,0,0,72,138,72,152,72
    ,165,103,133,133,133,165,104,133,134,16
    2,0,161,133,141,4,141,32,129,141,161,16
    1,133,141,5,141,32,129,141,161,133,
    205,0,141,208,49,32,129,141,161,133
    ,205,1,141,208,39,32,129,141,161,13
    3,205,2,141,208
210 DATA 57,165,133,133,125,165,134,13
    3,126,173,0,141,133,123,173,1,141,1
    33,124,169,0,141,3,141,104,168,104,
    170,104,96,174,4,141,208,13,173,5,1
    41,208,11,169,1,141,3,141,76,87,141
    ,173,5,141,134,133,133,134,76,19,14
    1,169,2,141,3,141
220 DATA 76,87,141,230,133,208,2,230,13
    4,96
230 GOSUB 1640
240 GOSUB 1700
250 GOSUB 1600: GOSUB 1530
260 GET J$: IF J$ < "J" OR J$ > "K" THE
    N 260
270 POKE 798, ASC (J$)
280 POKE 796,0: POKE 797,0:
290 CLEAR
300 J$ = CHR$ ( PEEK (798))
310 DIM SL%(39,14),PL%(1),CL%(4,1),OC(4
    )
320 HOME : VTAB 21: HTAB 16: PRINT "CAN
    NIBALS"
330 PRINT "LIVINGSTON: "; PEEK (796);"
    "
340 PRINT "CANNIBALS: "; PEEK (797);"
    "
350 GOSUB 1200
360 REM  GET PLAYER'S INPUT
370 DX = 0: DY = 0
380 IF J$ < > "J" THEN 480
390 M0% = PDL (0): FOR DE = 1 TO 10: N
    EXT DE
400 M1% = PDL (1)
410 REM  MAKE JOYSTICK INTO DIRECTION
420 IF M0% < 100 THEN DX = -1: GOTO 5
    30
430 IF M0% > 250 THEN DX = 1: GOTO 530
440 IF M1% < 100 THEN DY = -1: GOTO 5
    30
450 IF M1% > 250 THEN DY = 1: GOTO 530
460 IF DY = 0 AND DX = 0 THEN 670
470 GOTO 530
480 GOSUB 1160: IF ST = 0 THEN 670
490 IF KY = 69 THEN DY = -1: GOTO 530
500 IF KY = 68 THEN DX = +1: GOTO 530
510 IF KY = 83 THEN DX = -1: GOTO 530
520 IF KY = 88 THEN DY = 1: GOTO 530
530 IF DX = 1 AND PL%(0) = 39 THEN NX =
    0: GOTO 560
540 IF DX = -1 AND PL%(0) = 0 THEN NX
    = 39: GOTO 560
550 NX = PL%(0) + DX
560 IF DY = -1 AND PL%(1) = 0 THEN NY
    = PL%(1): GOTO 590
570 IF DY = 1 AND PL%(1) = 14 THEN 1070
580 NY = PL%(1) + DY
590 IF SL%(NX,NY) = 0 THEN 620
600 IF SL%(NX,NY) < 5 THEN 930
610 GOTO 680
620 HCOLOR= 1: DRAW 05 AT PL%(0) * 7,PL
    %(1) * 8: SL%(PL%(0),PL%(1)) = 0: PL%
    (0) = NX: PL%(1) = NY: SL%(NX,NY) = 5
630 IF (PL%(0) / 2) = INT (PL%(0) / 2)
    THEN 05 = 4: GOTO 650
640 05 = 1
650 HCOLOR= 3: DRAW 05 AT PL%(0) * 7,PL
    %(1) * 8
660 D = 2: P = 100: GOSUB 2000
670 REM  MOVE CANNIBALS
680 FOR N = 1 TO 4
690 DX = 0: DY = 0
700 IF CL%(N,0) < PL%(0) THEN DX = 1: I
    F INT ( RND (1) * 2) GOTO 730
710 IF DX THEN 750
720 IF CL%(N,0) > PL%(0) THEN DX = -1
    : IF INT ( RND (1) * 2) THEN 750
730 IF CL%(N,1) < PL%(1) THEN DY = 1: G
    OTO 750
740 IF CL%(N,1) > PL%(1) THEN DY = -1
750 NX = CL%(N,0) + DX: NY = CL%(N,1) +
    DY
760 IF SL%(NX,NY) = 0 THEN 850
770 IF SL%(NX,NY) = 5 THEN 930
```

Cannibals

APPLE II Series

```
780 IF SL%(NX,CL%(N,1)) = 0 THEN NY = C
    L%(N,1): GOTO 850
790 IF SL%(CL%(N,0),NY) = 0 THEN NX = C
    L%(N,0): GOTO 850
800 DX = INT (RND (1) * 3) - 1
810 NX = CL%(N,0) + DX: IF NX < 0 OR NX
    > 39 THEN 800
820 DY = INT (RND (1) * 3) - 1
830 NY = CL%(N,1) + DY: IF NY < 0 OR NY
    > 14 THEN 820
840 IF SL%(NX,NY) > 0 THEN 800
850 HCOLOR= 1: DRAW OC(N) AT CL%(N,0) *
    7,CL%(N,1) * 8:SL%(CL%(N,0),CL%(N,
    1)) = 0:CL%(N,1) = NY
    :SL%(NX,NY) = N
860 IF INT (NX / 2) = NX / 2 THEN OC(N
    ) = 4: GOTO 880
870 OC(N) = 1
880 HCOLOR= 0: DRAW OC(N) AT NX * 7,NY
    * 8
890 D = 4:P = 2: GOSUB 2000
900 D = 4:P = 3: GOSUB 2000
910 NEXT N
920 GOTO 370
930 REM PERSON DIES
940 HCOLOR= 4: DRAW 3 AT PL%(0) * 7,PL%
    (1) * 8: GOSUB 1850
950 POKE 797, PEEK (797) + 1
960 VTAB 22: PRINT "LIVINGSTON: "; PEEK
    (796);"
970 PRINT "CANNIBALS: "; PEEK (797);"
980 PRINT "WOULD YOU LIKE TO PLAY AGAIN
    ?Y/N";
990 GOSUB 1160: IF ST = 0 THEN 990
1000 IF KY = 89 THEN 290
1010 IF KY = 78 THEN 1030
1020 GOTO 990
1030 TEXT : HOME : PRINT "FINAL SCORE:"
1040 PRINT "LIVINGSTON: "; PEEK (796)
1050 PRINT "CANNIBALS: "; PEEK (797)
1060 END
1070 REM PERSON GOES INTO WATER
1080 IF PL%(0) > SC% - 1 AND PL%(0) < SC
    % + 6 THEN VTAB 22: POKE 796, PEEK
    (796) + 1;"
    EEK (796);"
    : GOTO 980
1090 HCOLOR= 7: IF PL%(0) = 39 THEN PL%(
    0) = 38
```

APPLE II Series

```
1100 IF PL%(0) / 2 = INT (PL%(0) / 2) T
    HEN 1120
1110 DRAW 6 AT PL%(0) * 7,(PL%(1) + 1) *
     8: GOTO 1130
1120 DRAW 6 AT (PL%(0) * 7) + 1,(PL%(1)
    + 1) * 8
1130 GOSUB 1770: POKE 797, PEEK (797) +
    1: VTAB 23: PRINT "CANNIBALS: ";
    EK (797);"
1140 GOTO 980
1150 END
1160 REM READ KEYBOARD****
1170 POKE -16368,0: FOR DE = 1 TO 100:
     NEXT DE: KY = PEEK (-16384): IF
    KY > 127 THEN ST = 1:KY = KY - 128:
     POKE -16368,0: RETURN
1180 ST = 0: POKE -16368,0: RETURN
1190 REM DRAW FIRST SCREEN
1200 HGR : SCALE= 1: ROT= 0: HCOLOR= 1:
     HPLOT 0,0: CALL 62454
1210 REM DRAW WATER
1220 HCOLOR= 6: FOR R = 120 TO 159: HPLO
    T 0,R TO 279,R: NEXT
1230 FOR N = 1 TO 15:X% = ( INT ( RND (1
    ) * 20)) * 2:Y% = INT (( RND (1) *
     4)) + 16: HCOLOR= 7: DRAW 2 AT 7 *
    X%,8 * Y%: NEXT
1240 REM PLACE SHIP ON SCREEN
1250 SC% = RND (1) * 18:SS% = SC% * 14:
    SC% = SC% * 2: HCOLOR= 7
1260 FOR Y = 121 TO 124: HPLOT SS% + 15,
    Y TO SS% + 21,Y TO SS% + 6,Y TO SS%
     + 2
1270 Y = 125: HPLOT SS% + 6,Y TO SS% + 2
    9,Y: Y = Y + 1: HPLOT SS% + 6,Y TO
    SS% + 12,Y TO SS% + 13,Y: HPLOT SS%
     + 16,Y TO SS% + 17,Y: HPLOT SS% +
    20,Y TO SS% + 21,Y: HPLOT SS% + 24,
    Y TO SS% + 29,Y
1280 Y = Y + 1: HPLOT SS% + 6,Y TO SS% +
     37,Y: Y = Y + 1: HPLOT SS% + 1:YF
    = SS% + 37: FOR N = Y TO 132: HPLOT
     YF,N TO SS%,N: NEXT
1290 REM PLACE OBSTACLES
1300 FOR N = 1 TO 30
1310 X% = RND (1) * 40:Y% = RND (1) *
    15: SL%(X%,Y%) = 6: X% = X% * 7: Y% =
    Y% * 8
```

Continued on p. 80

Cannibals. . . **from p. 79**

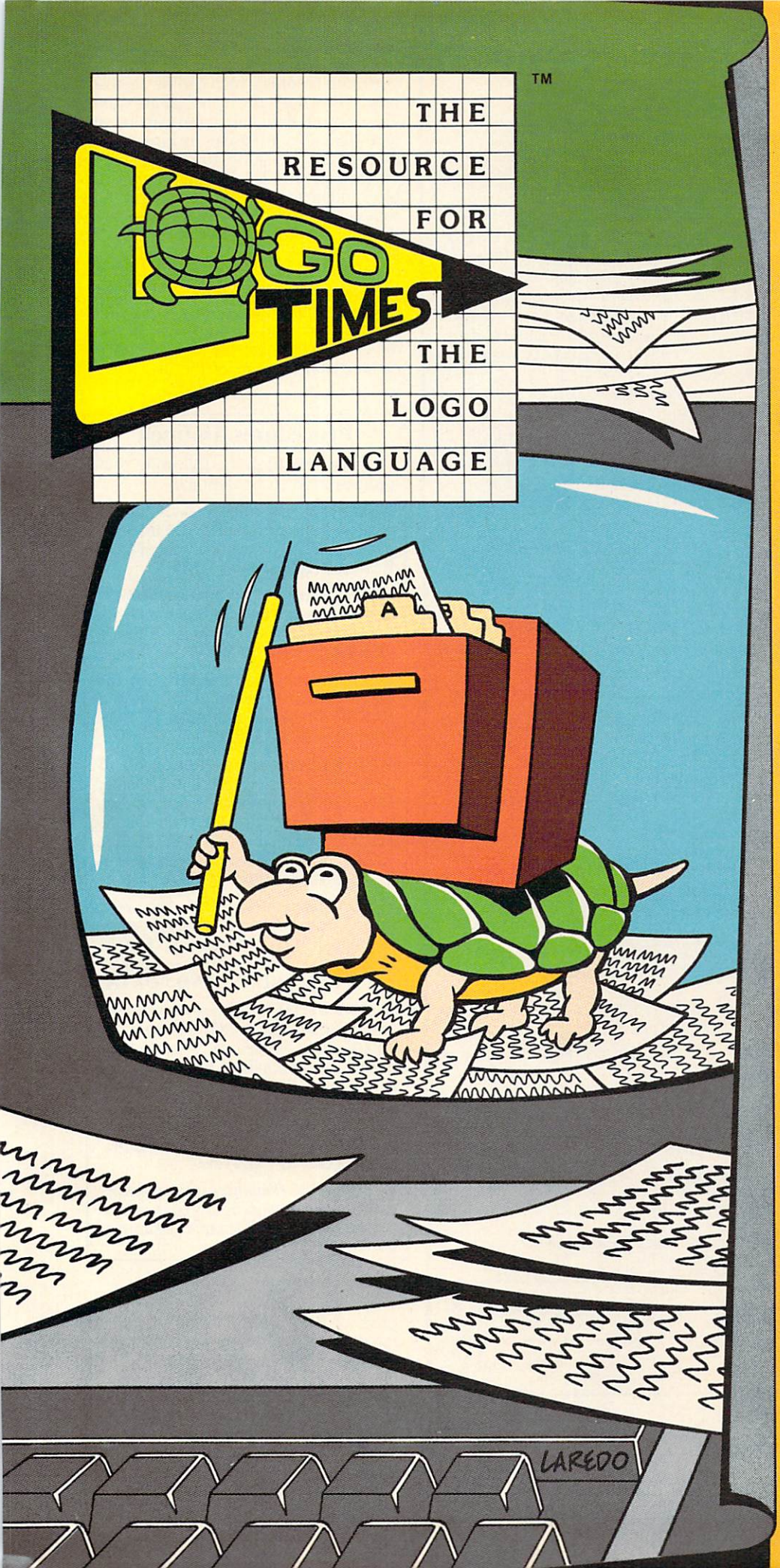
APPLE II Series

```
1320 HCOLOR= 6: FOR I = 0 TO 5: HPLOT X%
     ,Y% + I TO X% + 4,Y% + I: NEXT : NE
     XT
1330 REM  PLACE CANNIBALS ON SCREEN
1340 FOR N = 1 TO 4
1350 X% = RND (1) * 40:Y% = RND (1) *
     15: IF SL%(X%,Y%) > 0 THEN 1350
1360 SL%(X%,Y%) = N:CL%(N,0) = X%:CL%(N,
     1) = Y%
1370 IF X% / 2 = INT (X% / 2) THEN OC(N
     ) = 4: GOTO 1390
1380 OC(N) = 1
1390 HCOLOR= 0: DRAW OC(N) AT X% * 7,Y%
     * 8
1400 NEXT N
1410 REM  PLACE MAN ON SCREEN
1420 X% = RND (1) * 40:Y% = RND (1) *
     15: IF SL%(X%,Y%) > 0 THEN 1420
1430 SL%(X%,Y%) = 5:PL%(0) = X%:PL%(1) =
     Y%
1440 IF X% / 2 = INT (X% / 2) THEN O5 =
     4: GOTO 1460
1450 O5 = 1
1460 HCOLOR= 3: FOR N = 1 TO 8
1470 DRAW O5 AT X% * 7,Y% * 8: IF INT (
     N / 2) = N / 2 THEN HCOLOR= 1: GOT
     O 1490
1480 HCOLOR= 3
1490 D = 2:P = 100 + N * 9: GOSUB 2000
1500 NEXT N
1510 RETURN
1520 REM  PLAY TITLE THEME
1530 LN = 1590: GOSUB 2030
1540 FOR I = 1 TO 26
1550 READ D,P
1560 GOSUB 2000
1570 NEXT I
1580 RETURN
1590 DATA 4,187,4,218,4,206,2,197,2,206
     ,8,187,8,155,4,187,4,218,4,206,2,19
     7,2,206,8,187,8,155,4,187,4,206,2,1
     97,2,187,8,197,8,218,4,187,4,206,2,
     197,2,187,8,197,8,155
1600 REM  TITLE SCREEN
1610 HOME : VTAB 10: HTAB 14: PRINT "****
     ********": HTAB 14: PRINT "* CANN
     IBALS *": HTAB 14: PRINT "********
     ****"
1620 VTAB 23: HTAB 10: PRINT "PRESS J FO
     R JOYSTICKS": HTAB 10: PRINT " PRESS
      K FOR KEYBOARD";
1630 RETURN
1640 REM  SHAPE TABLE DATA
1650 LN = 1680: GOSUB 2030
1660 POKE 232,0: POKE 233,96
1670 FOR PA = 24576 TO 24692: READ PV: P
     OKE PA,PV: NEXT PA: RETURN
1680 DATA 6,0,14,0,27,0,37,0,50,0,62,0,
     76,0,18,45,36,53,46,181,58,39,60,62
     ,55,5,0,9,45,45,149,39,63,63,191,45
     ,0,146,42,45,45,62,63,63,46,45,45,
     2,22,63,0,82,45,36,53,46,181,58,39,
     60,62,63,0,146,10,45,45,53,63,63,55
     ,45,45,245,59,7,0
1690 DATA 173,63,46,45,45,21,63,223,55,4
     5,45,45,62,63,63,179,41,45,45,44,16
     4,36,8,12,53,21,63,63,46,45,53,63,6
     3,46,45,45,62,63,63,63,0
```

APPLE II Series

```
1700 REM  POKE SPEAKER ROUTINE
1710 FOR I = 768 TO 795
1720 READ IN
1730 POKE I,IN
1740 NEXT I
1750 RETURN
1760 DATA 0,210,172,1,3,174,1,3,232,208
     ,253,169,4,32,168,252,173,48,192,13
     6,208,239,206,0,3,208,231,96
1770 REM  PLAY SHARK THEME
1780 LN = 1840: GOSUB 2030
1790 FOR I = 1 TO 15
1800 READ D,P
1810 GOSUB 2000
1820 NEXT I
1830 RETURN
1840 DATA 4,187,4,202,8,210,8,210,8,210,4,187
     ,4,202,8,210,8,210,4,176,4,192,8,21
     0,8,210,4,222,4,218,16,210
1850 REM  PLAY SMOKE
1860 LN = 1910: GOSUB 2030
1870 FOR I = 1 TO 11
1880 READ D,P
1890 GOSUB 2000
1900 NEXT I: RETURN
1910 DATA 8,187,4,182,4,197,4,187,4,170
     ,16,210,4,206,4,218,4,210,4,197,16,
     234
1920 REM  PLAY ON BOARD SHIP
1930 LN = 1980: GOSUB 2030
1940 FOR I = 1 TO 10
1950 READ D,P
1960 GOSUB 2000
1970 NEXT I: RETURN
1980 DATA 4,170,4,170,8,182,4,182,4,182
     ,8,182,6,206,2,197,4,192,8,170
1990 REM  MAKE SOUND ROUTINE
2000 POKE 768,D: POKE 769,P: CALL 770: R
     ETURN
2010 HCOLOR= 6: HPLOT X% + 2,Y% + 1: HPL
     OT X% + 1,Y% + 2 TO X% + 3,Y% + 3:
     HPLOT X%,Y% + 4 TO X% + 4,Y% + 4: H
     PLOT X%,Y% + 5 TO X% + 5,Y% + 5: HPLOT
     X% + 1,Y% + 6 TO X% + 3,Y% + 6: HPL
     OT X% + 2,Y% + 7
2020 REM  DO LINE# RESTORE
2030 LH = INT (LN / 256): POKE 36097,LH
     :LL = LN - LH * 256: POKE 36096,LL
2040 CALL 36102: ST = PEEK (36099): IF
     T = 0 THEN RETURN
2050 IF ST = 2 THEN PRINT "NOT A DATA S
     TATEMENT IN LINE ";LN: END
2060 IF ST = 1 THEN PRINT "LINE NUMBER
     ";LN;" NOT FOUND": END
```

HCM



FILES IN LOGO

A Response to Muller's Challenge II

By Roger B. Kirchner

As many readers will recall, Jim Muller, President of the Young People's LOGO Association, formulated three challenges to give LOGO fans an opportunity to answer criticisms of the LOGO language. In our July 1983 issue Roger Kirchner answered Muller's third challenge with a tessellation procedure for "multi-colored mosaics that graphically represent multiple depths of recursion." Then in August, Kirchner responded to Muller's first challenge, demonstrating that TI LOGO could perform word-processing functions on either a 40- or 80-column screen and print out in 40 or 80 columns. Now Kirchner completes his defense of TI LOGO by demonstrating that the unfairly maligned language can even be made to access data files as BASIC can.

Jim Muller's second LOGO challenge calls for an interactive graphics program which includes data file access such as is offered by BASIC. TI LOGO has a limitation that makes data file access a challenge: Although you can access a disk while in the LOGO direct mode—with the SAVE and RECALL commands—no one has found a way to access the disk from procedures, i.e., while a program is running. Thus, our solution will be to write procedures that set up and maintain files in immediate memory. The files can then be saved as procedures.

It is possible to meet this challenge because LOGO is an extensible language. Commands and operations can be defined and used just like primitives. So new capabilities can constantly be created.

A file is a nameable collection of elements called *records*. Each record may be a simple data item or a structured collection of data. Information is written to and read from a file one record at a time.

The file commands we'll introduce here are similar to the file commands in Pascal. Readers familiar with Pascal will recognize this at once, and others will learn something about how files are used in Pascal. Standard Pascal is limited to the simplest kind of files, *sequential* files. They

Introduction

LOGO Times is an information resource for users who want to create their own *personal* languages—languages that will easily allow them to communicate with the computer in a totally new audiovisual realm of applied imagination, exploration, and self-discovery. The articles on these pages concern the use of the LOGO language, but readers do *not* need any additional software or equipment (or even a computer) to understand and learn from the material presented here.

If readers want to actually *experience* a LOGO environment, they will need a computer, the requisite software and/or cartridges, and any additional hardware required for a particular implementation. A disk drive is required for some LOGO implementations, but in other cases, a user's work may be saved on cassette tape, or copied into a notebook (for later re-keyboarding).

The varieties of LOGO we'll consider include—but are not limited to—Terrapin LOGO for the Apple II, II+ or IIe and the Commodore 64, TI LOGO for the TI-99/4A, and LOGO Computer Systems LOGO for the IBM PC and PCjr.

- **Apple:** Terrapin LOGO requires an Apple II, II+ or IIe with 64K of RAM, one disk drive with controller, and a blank, initialized disk.
- **Commodore 64:** Terrapin LOGO requires a Commodore 64 with a VIC-1541 Disk Drive and a blank, initialized disk.
- **TI-99/4A:** TI LOGO requires the TI LOGO or TI LOGO II cartridge and a compatible 32K memory expansion unit. A cassette recorder may be used for storage, but a compatible disk system is recommended for convenience.
- **IBM PC or PCjr:** LOGO Computer Systems LOGO requires the PC or PCjr with 128 bytes of RAM, one disk drive, and a blank, initialized disk.

In each issue, one or more of the articles may refer to or build upon the topics discussed in a previous article. It is therefore recommended that for maximum benefit and understanding, new readers obtain the appropriate back issues of *Home Computer Magazine* containing *LOGO Times* articles.

LOGO Listings

As you enter LOGO statements, the last thing you do at the end of every statement is to press [ENTER] on the TI and IBM (the key with the ↵ symbol), or [RETURN] on the Commodore 64 and Apple. This signals the system to begin a new line. In our typeset listings, single LOGO statements may carry over from one line to the next without ending. The end of a LOGO statement is marked with a curved arrow (↩) to indicate that you press [ENTER] or [RETURN] at that point.

Notice

LOGO Times is actively soliciting articles. Manuscripts should be typed double-spaced, and accompanied by a cassette tape or disk if containing any lengthy procedures or graphics.

Send all materials to:

LOGO Times Editorial Dept.
Home Computer Magazine
1500 Valley River Dr., Suite 250
Eugene, OR 97401

All mail directed to the Letters-to-the-Editor column (*Letters on LOGO*) will be published in accordance with the conditions set forth on *Home Computer Magazine's* Masthead page.

Contents

Volume 4, Number 2

81. Files in LOGO — *by Roger Kirchner and the HCM Staff*
Muller's last Challenge is answered

83. LOGO Spans the Generation Gap — *by Richard and Sarah Haller*
A Review of Commodore LOGO

84. FROGO: LOGO Invades the Arcade — *by Ted Barnicoat*
Frogo Meets LOGO Turtle

consist of records which can be accessed only in the order in which they are stored. The records in a Pascal file can be any type definable in Pascal, as long as they are all of the same type. In LOGO, records can also have this arbitrary structure, and in addition, the uniform file type requirement is lifted: The structures of individual records in a file may vary. So LOGO files are slightly more flexible than those of Pascal.

The salient feature of sequential files is that a marker indicates the record which can be read next. Records can be written to a sequential file only when the marker is at the end of the file.

Procedures Defined

To process files, the following procedures need to be defined:

REWRITE *filename*—creates (or recreates) a file with name *filename* and prepares it so data can be written to it. Example: **REWRITE "DATA.**

RESET *filename*—prepares *filename* so that data can be read beginning with the first record. Example: **RESET "DATA.**

EOF *filename*—outputs **TRUE** if the file marker is at the end of the file, and **FALSE** otherwise. Example: **IF EOF "DATA THEN STOP.**

WRITE *filename data*—causes *data* to be stored as the next record in *filename*. *Data* can be a number, word, or list. Gives an error if **EOF** *filename* is **FALSE.** Examples: **WRITE "DATA 1234,** and **WRITE "NAMES [[C.Q.UMBER][SILICONGULCH]].**

FIND *filename*—outputs the next record from *filename*. Gives an error message if **EOF** *filename* is **TRUE.** Each read advances the file marker. Example: **MAKE "X FIND "DATA.**

SCANTOEND *filename*—causes successive reads until **EOF** *filename* is **TRUE.** This procedure is used when more information is to be added to an existing file. There is no corresponding Pascal procedure, but it is included for convenience and as an example of the use of file commands. Example: **SCANTOEND "DATA.**

CLOSEFILE *filename*—causes information in the file to be stored as a procedure with name *filename*. Example: **CLOSEFILE "DATA.**

Using the Files

To illustrate the use of files, we'll suppose we have a list of names of people together with the cities in which they live. Let's call the file NAMELIST and consider a record to be a list of two lists: the first a list of a person's first and last names, the second a list of the words in the city's name. We can activate all procedures directly, as well as from within another procedure. Let's create the file using direct commands as follows:

Continued on p. 94

LOGO Spans the Generation Gap

A REVIEW OF COMMODORE LOGO

Name:	**LOGO for the Commodore 64**
Program Type:	**Logo Language**
Machine:	**Commodore 64**
Developer:	**Terrapin, Inc.**
Distributor:	**Commodore Software 1200 Wilson Drive West Chester, PA 19380**
Price:	**$60.00, diskette**

System Requirements:
Disk drive

	poor	fair	good	excellent
Performance				
Ease of Use				
Documentation				

by Richard and Sarah Haller

One of my goals in purchasing a home computer (a Commodore 64) was to give my daughter Sarah a head start in this Computer Age. After all, I make our living as a programmer. Shouldn't the cobbler's daughter have shoes? But despite exposure to a couple of books intended to teach people her age (13) to program in BASIC, Sarah appeared to prefer going barefoot.

Then about a year after we got our computer, LOGO became available for the Commodore 64 for only $60! Well, more like $360 since we needed a disk drive too, but I had wanted a drive anyway. So with the help of MasterCard, I became the proud possessor of Commodore 64 LOGO.

This is not a review of the LOGO language. *Mindstorms*, a book by LOGO's creator, Seymour Papert, is available in paperback (Basic Books), and is an exciting, if necessarily biased, discussion of the philosophy and history of LOGO. As you will see, I do highly recommend LOGO to anyone who wants to learn to program or improve their programming skills, but my focus here is on the values and defects of this particular version for the Commodore 64.

The Neophyte Perspective

First, I wanted to see if this LOGO package was designed with the beginner in mind. Since I am an experienced programmer, I asked my daughter (now 14) to sit down with the 350-page tutorial that comes with Commodore LOGO and work through the material as much as possible on her own. Besides, I supposedly got it for her in the first place! Her comments follow:

"Since we first had a computer, my father taught me the basic and not-so-basic things about computers. When he finally bought LOGO and a disk drive, I thought, 'This is a good chance to learn more about computers.' He told me about the LOGO review and asked me to try some of it out.

"I don't know anyone my age who would want to sit down and read *LOGO: A Language for Learning*, unless they were very serious about computers. To me, the book seems like an endless 'follow the leader.' I really enjoyed the Turtle program [in the Commodore LOGO Tutorial], though. It gives you the chance to learn to command the computer to make pictures.

"The little signs in the book like Rabbit, Elephant and Turtle are great, too. They give different kinds of people a chance to go at their own pace. But honestly, the book has me baffled. At times I feel that they think I am 3, and at others, it sounds as though they think I am majoring in computers.

"The graphics part was a bit repetitious while the sprite system stuff was much too complicated for younger or even older kids. The demos of the runner were very tedious. I don't mean to say that the book is no help at all; it is. I just think a shorter and medium-knowledge range version should be published."

The Programmer's Viewpoint

Now for my review. The materials supplied with the package are one copy of the LOGO system itself on a single disk, a second disk with over fifty demonstration or utility procedures written in LOGO, and a 350-page beginner's tutorial and reference manual, entitled *The Commodore 64 LOGO Tutorial*. I think that the manual is excellent. It is aimed at secondary school students and above. A LOGO primer aimed at elementary students is reportedly under development. The motivated student should be able to successfully master all the material with virtually no previous background in computers and with only occasional assistance.

"Hey, Dad, didn't you read *my* review?"

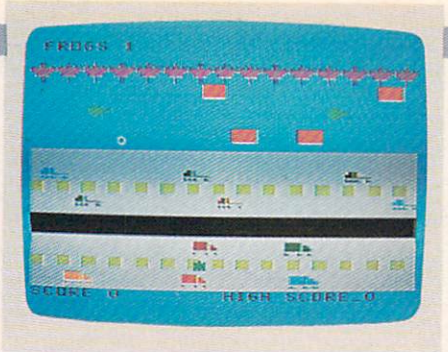
"Not before I wrote this!"

The quicker student may find, as Sarah did, that some of the sections move rather slowly. On the other hand, most people will find the Sprite section hard going. This has less to do with LOGO than with the fact that sprites are a complicated subject. My recommendation is to be patient and be careful about skimming, even though, as Sarah noted, the little animal signs will help you do so if you wish.

C-64 LOGO was developed for Commodore by Terrapin, Inc., who earlier developed a highly regarded version for the Apple II. Happily, in addition to providing all the the LOGO primitives and procedures one would expect, C-64 LOGO supports convenient use of the 64's unique sprite graphics and three-voice sound synthesizer. As a matter of fact, it is infinitely easier to write programs for exploiting these features in LOGO than it is in Commodore's own BASIC. Similarly, the DOS command allows one to issue Disk Operating System commands from within LOGO more conveniently than in BASIC.

The screen editor embedded in LOGO is superior to that provided with BASIC. Combine this with intelligible error

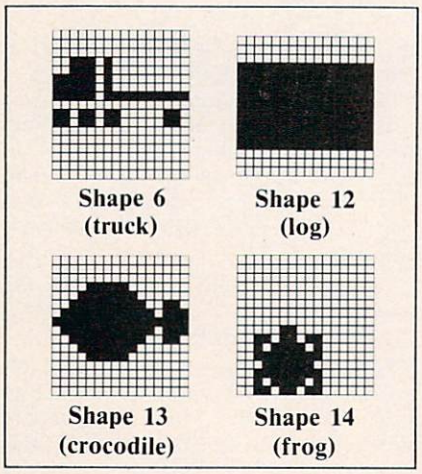
Continued on p. 92



FROGO: LOGO Invades the Arcade

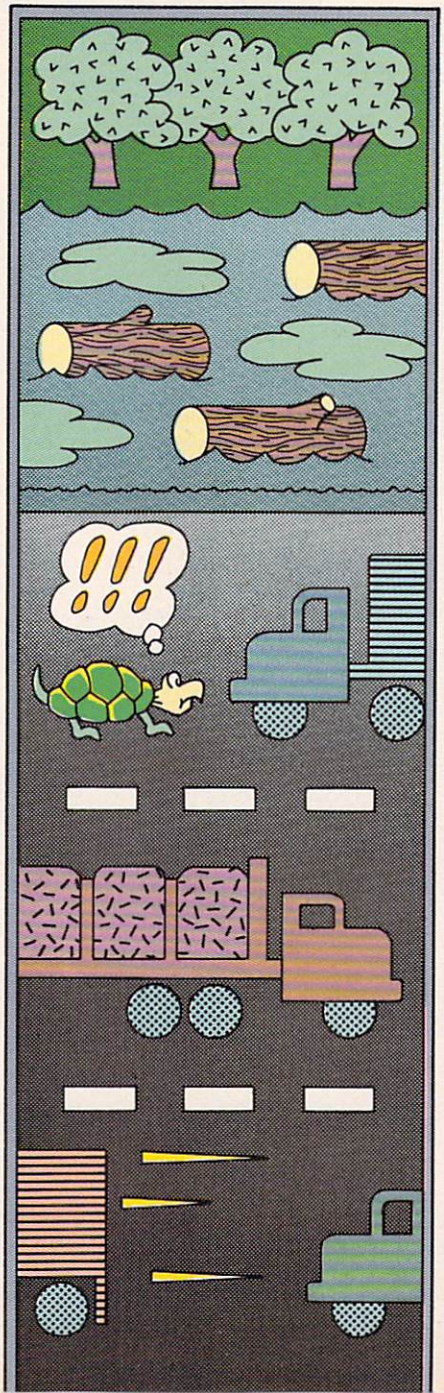
by Ted Barnicoat

I had been looking for quite some time for a vehicle to teach computer literacy to my children, aged 8 and 10. I eventually selected the TI-99/4A as an inexpensive but powerful computer with the promise of great graphics; and an article by Roger Kirchner called "LOGO Has Style" (*99'er HCM*, November 1982) convinced me that TI LOGO was the learning tool I needed. LOGO not only promotes logical problem-solving habits, but it also allows you to test each unit, subroutine, or procedure (whatever you like to call it) independently. This is a marvelous feature for kids, who want to see progress right away.



FROGO came out of a small neighborhood club I have formed for my kids and some of their friends. Called the "Logomaniacs," its aim is to give them a step forward in computer literacy while they have a bit of fun along the way. (We also play lots of table tennis and commercial video games.) FROGO was our first project, and we designed it to show ourselves that we could use LOGO and come up with a good approximation of an arcade game. At the same time, we wanted to learn about sprites, tiles, and recursive procedures.

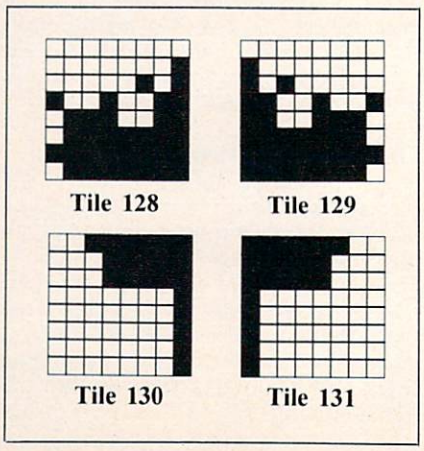
FROGO was designed *top down*. In the first two-hour session we developed the mainline program, using dummy procedures for all except the hopping, skipping, and scoring routines. The kids also completed the TITLE and ROAD procedures, which taught us all a good deal about tiles and rows and columns on the screen. In the second session we tackled TRAFFIC and learned some valuable lessons about *x* and *y* coordinates. After that, the kids took off and programmed the other sprites on their own. Our hardest task was the coincidence routines, COINC and ONIT, which only goes to prove that we should re-read our *LOGO Times* before starting our projects. Both Vol. 1, No. 5, and the December 1982 issue specifically explain the how-to's of these routines.



FROGO'S familiar scenario asks you to hop a frog across four lanes of traffic, with cars moving at different speeds in the four lanes. If your frog survives this ordeal, he must make his way home—to some lilypads on the opposite side of the river—by skipping onto two sets of logs floating down the river, and then onto some crocodiles swimming upstream. Each time he succeeds, you score, the frog is returned to the start, and the sprites are speeded up.

After you type in the procedure definitions, you will have to define the tiles using MAKECHAR and the shapes using MAKESHAPE. Then to start the game, type FROGO. Use the E,S, and D keys to move the frogs. Remember that the keys don't auto-repeat in LOGO and that LOGO stores keypresses in the keyboard buffer. This is why the frogs may hop or skip even though you think you haven't pressed a key—it's a stored keypress in the keyboard buffer that's responsible.

This game proved to be a valuable learning experience for all of us in understanding rows and columns, *x* and *y* coordinates, and direction and speed. We had fun devising and playing it, and I hope that you will too.



FROGO
Explanation of the Program

Procedure Title	
FROGO	Start game.
TITLE ROAD LILYPADS	Use tiles to lay out title and game screens.
SETUP	A collection of commonly used procedures to re-initialize all moving sprites and the score.
START	Re-initialize the game.
TRAFFIC LOGS SNAPPER FROGS	Set up moving sprites with specific X and Y coordinate positions with different directions and speeds.
GAME	Recursive procedure that allows the game to be played.
HOP	Move frog 1 row forward or sideways as a result of key input.
SKIP	Move frog 2 rows forward or sideways as a result of key input.
COINC ONIT	Check for coincidence of frog with traffic or logs/crocodiles. If "yes" for trucks or "no" for logs/crocodiles, the frog comes to a sticky end.

TI-LOGO — TI-99/4A

```
TO FROGO
TITLE
ROAD
LILYPADS
SETUP
PRINT [SCORE ]
GAME
END

TO TITLE
CB 15
CS
TELL :ALL SC 0
TELL TILE 96 SC [9 9 ]
REPEAT 30 [PC 96 ]
REPEAT 5 [PRINT [ ] ]
REPEAT 10 [PC 32 ]
PRINT [F R O G O ]
PRINT [ ]
REPEAT 13 [PC 32 ]
PRINT [BY ]
PRINT [ ]
REPEAT 7 [PC 32 ]
PRINT [THE LOGOMANIACS]
REPEAT 8 [PRINT [ ] ]
REPEAT 30 [PC 96 ]
MAKE "R 1
REPEAT 20 [PUTTILE 96 1 :R PUTTILE 96 30 :R - 1 MAKE "R :R + 1 ]
PRINT [PRESS E S OR D TO PLAY ]
CALL RC "J
END

TO SCORE
MAKE "T :T + 1
PUTTILE :T 8 23
TELL [1 2 3 ] SS SPEED + 5
TELL [4 5 6 ] SS SPEED + 5
TELL [7 8 9 ] SS SPEED + 5
TELL [10 11 12 ] SS SPEED + 5
TELL [16 17 19 20 ] SS SPEED + 3
TELL [22 23 ] SS SPEED + 1
END
```

Continued on p. 86

Frogo . . . from p. 85

TI-LOGO — TI-99/4A

```
TO TRAFFIC
TELL [1] 2 [3]
SH 90 SS 5 CARRY 2
TELL 1 SC 7 SXY 70 ( -
75 )
TELL 2 SC 8 SXY 0 ( - 7
5
TELL 3 SC 9 SXY - 70 (
- 75
TELL [4] 5 [6]
SH 90 SS 10 CARRY 2
TELL 4 SC 1 SXY - 70 (
- 50
TELL 5 SC 6 SXY 40 ( -
50
TELL 6 SC 12 SXY 100 (
- 50
TELL [7] 8 [9]
SH 270 SS 10 CARRY 6
TELL 7 SC 12 SXY 100 (
- 15
TELL 8 SC 7 SXY 40 ( -
15
TELL 9 SC 6 SXY - 70 (
- 15
TELL [10] 11 [12]
SH 270 SS 5 CARRY 6
TELL 10 SC 7 SXY 110 (
5
TELL 11 SC 1 SXY 40 ( 5
TELL 12 SC 12 SXY - 60
(
END

TO LOGIC
TELL [16] 17 [19] 20 ]
SH 270 SS 5
CARRY 12
SC 8
TELL 16 SXY 70 36
TELL 17 SXY 30 36
TELL 19 SXY 120 68
TELL 20 SXY 10 68
END

TO SNAPPERS
TELL [22] 23 ]
SH 90 SS 3 CARRY 13
3
TELL 22 SXY 110 52
TELL 23 SXY 0 52
END
```

TI-LOGO — TI-99/4A

```
TO ROAD
CS
CBELL 7
CTELL TILE 104 SC [14 14
REPEAT 10 [PRINT [ ] ]
REPEAT 390 [PC 104 ]
RTELL TILE 112 SC [10 10
MAKE "C 15
REPEAT 15 [PUT TILE 112
: C 20
MAKE "C 15
REPEAT 13 [PUT TILE + 2
: C ]
TELL TILE 120 SC [1 1 ]
END

TO LILYPADS
TELL TILE 128 SC [13 7
]TELL TILE 129 SC [13 7
]TELL TILE 130 SC [13 7
]TELL TILE 131 SC [13 7
]MAKE "C 15
REPEAT 15 [PUT TILE 128
: C + 2 ]
MAKE "C 15
REPEAT 13 [PUT TILE 130
: C + 2 ]
END

TO TRAFFIC
TELL [1] 2 [3]
SH 90 SS 5 CARRY 2
TELL 1 SC 7 SXY 70 ( -
75
TELL 2 SC 8 SXY 0 ( -
75
TELL 3 SC 9 SXY - 70 (
- 50
TELL [4] 5 [6]
SH 90 SS 10 CARRY 2
TELL 4 SC 1 SXY - 70 (
- 50
TELL 5 SC 6 SXY 40 ( -
50
TELL 6 SC 12 SXY 100 (
- 15
TELL [7] 8 [9]
SH 270 SS 10 CARRY 6
TELL 7 SC 12 SXY 100 (
- 15
TELL 8 SC 7 SXY 40 ( -
15
TELL 9 SC 6 SXY - 70 (
- 15
TELL [10] 11 [12]
SH 270 SS 5 CARRY 6
TELL 10 SC 7 SXY 110 (
5
TELL 11 SC 1 SXY 40 ( 5
TELL 12 SC 12 SXY - 60
(
END

TO LOGIC
TELL [16] 17 [19] 20 ]
SH 270 SS 5
CARRY 12
SC 8
TELL 16 SXY 70 36
TELL 17 SXY 30 36
TELL 19 SXY 120 68
TELL 20 SXY 10 68
END

TO SNAPPERS
TELL [22] 23 ]
SH 90 SS 3 CARRY 13
3
TELL 22 SXY 110 52
TELL 23 SXY 0 52
END
```

HCM

HERE'S AN OFFER YOU CAN'T REFUSE

Join The 99/4A Program Exchange Users Group for a one-time fee of only $10. Then pay only $2 each for programs. That's right. Only $2 each.

Membership provides. . .

. . . **LOW PRICES.** Programs for 99/4A users on cassette or diskette. $2 each.

. . . **QUALITY & DIVERSITY.** Over 1000 programs for 99/4A users. Games. Demonstrations. Education. Business. Home. Application. Utility.

. . . **NEWSLETTER** filled with hints, program reviews, problem solvers, & much more.

. . . **5 for 1 OPTION.** Submit a program to our software library & receive 5 FREE programs in exchange.

That doesn't mean you need programming experience to become a member.

. . . **1-TIME MEMBERSHIP FEE.** One guy we know about refused to join. He disappeared. Get the picture? We expect to hear from you. And soon. Besides, it's the best deal in town.

O.K. I GET THE PICTURE. HERE'S MY $10, THIS ENTITLES ME TO **PROGRAMS FOR $2 EACH** AND YOUR MONTHLY NEWSLETTER.

☐ Check or Money Order ☐ Visa/Mastercard

Card No. _____ Exp. _____

Name _____

Address _____

City _____

State _____ Zip _____

The 99/4A Program Exchange, P.O. Box 3242, Torrance, CA 90510

RENT&PLAY™ your favorite games for $9.95.

Until now, you could play a computer game one of two ways—borrow a friend's or buy it. Either way, someone gets stuck paying for an expensive cartridge.

But with RENT&PLAY™ you don't have to buy to play.

HERE'S THE LOWDOWN.

Join RENT&PLAY computer game club and rent the games you want from our ever increasing list of titles. It only costs you $9.95 each game.

You can keep and play a game as long as you like—just return it to us when you're ready for something new. You can even rent as many titles as you want.*

When you rent from us you get real cartridges. No disks or cassettes to fool with. Just plug in and play.

RENT&PLAY is now available for TI 99/4A owners. But, we will have more computer formats soon.

To join all you do is pay our lifetime membership fee of $25.00. This includes your first game and a one year subscription to the RENT&PLAY newsletter.

FREE OFFER!!
If you are one of the first 200 new members responding to this ad we will send you VIDEO VEGAS by Creative Software—*free*! So don't wait.

WHOSE GAMES DO WE HAVE?

Glad you asked. We feature games from publishers such as Creative Software, Funware, Romox and others. And this is just the beginning. Soon you'll have a virtual supermarket of game titles and publishers to choose from.

BUST THE GROM.

If you have the new V2.2 TI console you haven't been able to play most games from other publishers. Now you can. Our *GROM Buster* is the answer. It plugs directly into the I/O port on the side of your console. With the push of a button you will transfer control to the cartridge plugged into the game slot.

The GROM Buster costs only $39.95 and gives you full control of your new console.

Call today, (408) 866-8579. VISA and MasterCard are welcome.

Or write us: Navarone Industries, Inc., 510 Lawrence Expressway, #800, Sunnyvale, California 94086. If ordering by mail, send Cashier's check or Money Order payable to Navarone Industries, Inc. (Personal checks take 10 days to clear.)

All orders *must* include 5% of total for shipping and handling. Orders sent outside the U.S. will require a minimum shipping charge of $10.00. California residents add 6.5% sales tax.

NAVARONE

Navarone Industries, Inc., 510 Lawrence Expressway, #800, Sunnyvale, CA 94086

*Each additional game cartridge requires a refundable deposit of $20.00.

WHAT'S A FLIPPER/SNAPPER™?

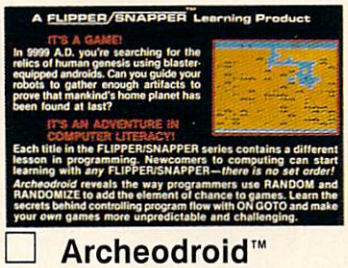
IT'S A 3-IN-ONE VALUE PACK!

- *A quality, ready-to-run game on cassette tape or diskette.*
- *A complete, easy-to-use programming lesson on a deck of colorful flip cards.*
- *A durable and attractive collector's case for your software library.*

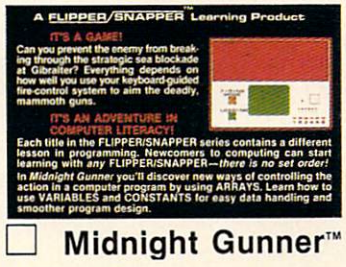
TI-99/4A cassette versions now available! Apple, Commodore, and IBM disk & cassette versions coming soon.

DISCOVER HOW MUCH FUN LEARNING CAN BE!

MAKE YOUR SELECTION



☐ **Archeodroid™**



☐ **Midnight Gunner™**

☐ **Simon's Saucer™**

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

☐ Please send me 1 FLIPPER/SNAPPER for $10.95 plus $2.00 shipping and handling.

☐ Please send me 2 FLIPPER/SNAPPERs for $19.95 plus $3.00 shipping and handling.

☐ Please send me all 3 FLIPPER/SNAPPERs for $29.95 plus $3.50 shipping and handling.

BE SURE TO SELECT YOUR TITLE CHOICE(s) BY CHECKING APPROPRIATE BOX(es) AT LEFT.

Bill my ☐ **VISA** ☐ **MasterCard**

Expiration Date Tel. No.

Signature

☐ **Check or Money Order Enclosed**

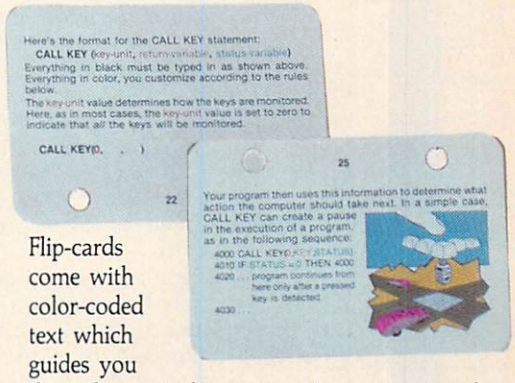
MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Foreign surface add $3.00 for shipping.

Enclose check, money order or credit card information to:

WORLD CLASS SOFTWARE
P.O. Box 5537
Eugene, OR. 97405

Use VISA/MasterCard and order Toll-Free
1-800-828-2212
In Oregon, Alaska, and Hawaii
telephone (503) 485-8796



Flip-cards come with color-coded text which guides you through essential programming practices and concepts in an easy-to-follow lesson format.

WORLD CLASS™ SOFTWARE

An Emerald Valley Company

Upgrading Your Commodore Disk Drive

If you have upgraded your home computer from a VIC-20 to a Commodore 64, you have probably found that your old 1540 disk drive does not work well with the C-64. However, that is no reason to retire the old machine. By swapping just one chip in the 1540, you can upgrade it to a 1541. It can be done by any full-service Commodore dealer, or you can try asking a student from a local junior college electronics department to do the job for you. Or, if you dare, you could even try upgrading it yourself.

If you choose to try it yourself, first read the entire procedure below. Then unplug the drive, take the top cover off, and reread this procedure while identifying each part inside the drive. If you have the slightest doubt about your ability to upgrade the drive, leave it to a dealer. Dealers will charge around $40 for the upgrade (including all parts and labor) while the average repair bill is around $120. You decide which bill you would rather pay.

If you decide to go ahead with the project, first order Commodore part #901229-03 from your dealer. When you ask the price, be sure to find out if it includes labor. If it doesn't, ask how much it would be to install the part, and compare offers.

Now, get out the drive and unplug it. This is very important. To get the cover off, flip the drive over on its back and remove the four phillips head screws, one in each corner. Holding the two halves together with your hands, turn it right-side-up. Lift the cover off and put it aside. Even though it's unplugged, don't touch anything unless you must. The chip you want is UAB5, which is located on the right side, next to the big silver voltage regulators. To be certain that it is the right chip, check it for number 325303-xx (xx being any number; it doesn't matter as long the chip has the first numbers). There may be other numbers on it too, but you should disregard them.

Look at the unit right-side-up, with the drive door facing you. You will see a metal lid which covers about three-fourths of the green-colored printed circuit board (PC board). Depending on how dexterous you are, you may need to remove this lid. If you can get to IC chip UAB5 unhindered, then just leave the lid in place. But if you must remove it, watch out for a little booby trap. On the left side of the lid are two wires which go to the LED on the case. You will have to remove one side of the cover, reach under it, and unplug this connector to get the cover completely off. Be sure to notice how the plug is situated, since it will have to be replaced in the same fashion if the LED is to work properly.

Before you pry UAB5 from its socket, notice the small notch cut in the chip. There will be a notch just like it on the new ROM you bought, so when you install the new one, make sure the notch is facing the same direction as the one you removed. If you forget to check the notch position before you remove the chip, look at the socket. Under it, printed in white on the PC board, is a drawing of the chip with the cutout clearly marked. If power is applied to an improperly inserted chip, it will probably ruin the chip, so it is important to have the notches properly lined up.

Use caution when inserting the new chip. Its tiny legs bend easily, and once bent, they break off with even greater ease when you attempt to straighten them. Be certain that you get the legs in the proper sockets, because improper insertion could damage your disk drive when you try to run it later. You may want to buy or borrow an IC Inserter tool, which makes the job safer and easier. Instructions on using the inserter will accompany it.

Press the chip in gently but firmly with one hand, while supporting the PC board from underneath with the other. Too much pressure on an unsupported board will crack it. Now check for bent pins. If everything looks OK, put the drive back together in the reverse order of the way it came apart.

Congratulations! You are now the proud owner of a 1541 disk drive.

—Ted Martino

***Home Computer Magazine* assumes no liability for damage to disk drives or other components as a result of this procedure. Readers who attempt this conversion do so entirely at their own risk and should be aware that tampering in any way with a Commodore product may void any existing warranty.**

The Keyboard Buffer

Deciphering the keyboard buffer may be one of the most difficult parts of learning to use the PCjr for BASIC programming. Many programs, particularly games, require that only the key being pressed be read by the system. The buffer accepts keypresses from the keyboard and stores them until they are read by the program. Up to 15 keypresses can be received and stored in the PCjr's buffer. The IBM's keys are *typematic*. This means that when a key is held down it autorepeats. If it is held down long enough, it fills the buffer. When the buffer is full the program is actually reading the next available key in the buffer instead of the keyboard. If your program reads only one character from the buffer and there are still more characters stored there, then those remaining will stay in the buffer until the program reads them.

If the program later reads the keyboard for a prompt or command, the characters in the buffer will be read first before any new keys are read. This is an easily remedied problem though, because it's possible to clear the keyboard buffer with the following two commands:

```
DEF SEG=0 : POKE 1050,PEEK(1052)
```

If you use these statements just before your program reads the keyboard, the buffer will be cleared of any previous keypresses. Use a little caution though or the keyboard may not be read at all. If your program doesn't continue to read the keyboard until a key is pressed, it may execute the INKEY$ statement and go on without getting the keypress. This is because it executes the statements so quickly that there isn't enough time between clearing the buffer and executing the INKEY$ statement to press a key.

One solution is to place a short delay between the clearing of the keyboard buffer and the reading of the INKEY$ function. Another solution is to have your program check to see if a key has been pressed; if one hasn't, it will keep reading the keyboard until one has. Try using one of these short routines in your program:

```
1000 DEF SEG=0:POKE 1050,PEEK(1052)
1010 A$=INKEY $:IF A$="" THEN 1010
```

or:

```
1000 DEF SEG=0:POKE 1050,PEEK(1052)
1010 FOR DELAY=1 TO 250:NEXT DELAY
1020 A$=INKEY$
```

The first routine will wait until a key has been pressed; the second routine will not wait, but it does give the user time, after clearing the buffer, to press a key. You may want to change the value of the DELAY loop to give yourself more time. Placing code between the INKEY$ function and clearing the keyboard would serve the same purpose. An alternative solution is to set up special keys with the KEY statement so the program will branch to a specific line number when the key is pressed and not send the key to the buffer. Unfortunately, you can only do this with six keys of your choice. Keys 1 through 10 are reserved for the function key, while keys 11, 12, 13 and 14 are used for the arrow keys. This leaves keys 15 through 20 available to be user defined with the KEY statement.

—William K. Balthrop

Generation Gaps. . from p. 83

messages and solid debugging capabilities (e.g., TRACE, PAUSE, CONTINUE), and program development becomes a pleasure. C-64 LOGO even contains a package of procedures that allow one to write assembly language subroutines that can be called from a regular LOGO program. (Unfortunately, these routines have to share the same space as sprites, which will make it difficult, perhaps impossible, to use both capabilities in the same program.) The overall effect is to encourage the development of programming skills to virtually any level you might desire. LOGO may have been designed for children, but it is the best introduction to programming for people of all ages.

If you are already a BASIC programmer, I would recommend LOGO to you as a better example of a good programming environment. Once you've mastered LOGO, I don't think you'll want to go back to BASIC unless you are writing commercial application programs where CBASIC and the like have entrenched themselves. Even if that is the case, you'll find that the programs you write in BASIC will be better for the LOGO experience.

Reconstructive Criticism

With this overall, wildly positive evaluation out of the way, I will proceed to get picky. I have two major criticisms of the C-64 LOGO package. First, only one copy of LOGO is provided, and it does not seem possible to back it up. The replacement cost of $5.00 for a damaged disk is quite reasonable, but the user would be left without a functioning LOGO while waiting for the replacement. I would rather see a backup copy provided with the package, even at a higher cost, with replacements available as needed. This is particularly in order since one is warned not to leave a disk in the drive while powering up or down. It makes sense to anticipate and provide for the possibility of damage with a backup that you can use while your damaged disk is in the mail.

"LOGO may have been designed for children, but it is the best introduction to programming for people of all ages."

My second major criticism is of the turtle display. A triangle to represent the turtle's location and orientation is a LOGO convention. Even with the "bottom" side of the triangle drawn twice as thick as the other two, the orientation of the turtle is clear only when it is facing one of the four cardinal compass points (NSEW). Because of the limited dot resolution of the video display (320 horizontal and 200 vertical), the rotated turtle shape is distorted, and it is often impossible to decide which side is the bottom and which vertex is the "head." I suspect that this problem is not unique to the 64, but will be characteristic of all home computer implementations due to limited dot resolution. I would prefer a shape for the turtle that is less ambiguous in its orientation when rotated.

I used Commodore's 1701 color monitor and the 3-wire cable. Nevertheless, I found that many lines that I drew with the turtle did not have consistent colors. Depending on the background and pen colors chosen, lines can change their colors at different locations and orientations to the extent that one would not be willing to bet what color they were intended to be. For example, a single vertical line can be a completely different color from the one it is supposed to be, depending on where it is drawn. (DOUBLE COLOR mode will, however, help this problem— at the cost of decreased resolution.) I had the most success with yellow or black on a white background and the least with red on a black background.

"Dad! I liked the changing colors!"

"Different strokes for different folks?!"

You will wish to experiment. It is possible to change the defaults for the DRAW and the TEXT modes to your own choices, as well as to change them dynamically. I found the "prismatic" white letters on the black background while in EDIT mode particularly distracting.

A set of routines to support the four-color plotter is supplied on the utility disk. So, if your ultimate desire is hard copy of your pictures, you don't have to worry about the changing colors.

Some minor criticisms: The material on workspace management could be better organized and given more emphasis. It turns out that it is possible to copy a pro-

cedure by editing its name. When you exit from the editor as usual by pressing [RUN/STOP] or [CTRL][C], a copy of the old procedure with a new name is defined. One can even rename procedures by copying as above and then erasing the old version. This is not obvious in the manual.

"Sorry, Dad, but you just lost us!"
"Touché, but I said I was being picky!"

Information on how to "clean up" your workspace via the ERASE command, especially its selective use, is listed under "Defining and Editing Procedures" instead of "Filing and Managing Workspace," a more natural place, to my thinking. I would have preferred more emphasis on saving selected procedures. For example, if you forget to surround the SAVE command (and its list of selected procedures to be saved) with the required parentheses, no error or warning is generated. Instead, the whole workspace is saved. This can lead to mysterious problems in the future if you recall the saved material, thinking it contains only the procedures you meant to save.

Manual Belabored

The manual states that "sprites can do anything that the turtle can do." This is not completely correct. For example, sprite shapes do not rotate in response to RT or LT commands as the turtle does. A related problem involves the behavior of the STAMPCHAR command. Since this was implemented with the 64's built-in screen memory structure (rather than as a simulation of a true bit-mapped graphics screen), the characters or graphics symbols do not change their orientation to match that of the turtle. In addition, they are restricted in their possible locations to one of the 40 by 25 possible character locations in the 64's text screen display.

The turtle or sprites can take on any of the 64 locations within the 8 by 8 dot matrix that forms each character location. As a result, moving the turtle and stamping another character will not always work as expected unless one moves in increments of 8. These are perfectly reasonable limitations and are found in other home computer versions of LOGO; I just wish they had been explicitly and prominently pointed out in the manual. On the other hand, one advantage of the Commodore version of LOGO is that you can write LOGO or assembler routines to get around this if you wish.

Overall, if you are wondering whether you should spend your money on C-64 LOGO, in my opinion, it represents an exceptional value. I can't think of a better one, except perhaps, a microcomputer itself! Anyone who is seeking a friendly and stimulating way to learn or teach programming skills would be hard put to find something better. And the price (including hardware) is very right.

To get down to "BASICs," forget them. Go LOGO!

"Anything more you want to say, Sarah?"

"Yes, I think you made me sound too negative. LOGO's great!" HCM

Files In LOGO . . . from p. 82

```
REWRITE "NAMELIST
WRITE "NAMELIST [[ERNST KIRCH
  NER] [ST. PAUL]]
WRITE "NAMELIST [[MARGARET
  MCCOURTNEY][NORTHFIELD]]
WRITE "NAMELIST [[C.Q. UMBER]
  [SILICON GULCH]]
CLOSEFILE "NAMELIST
```

After executing these commands, we can access this file with a procedure such as the following:

```
TO TESTNAMES "NAMELIST
RESET "NAMELIST
1:
IF EOF "NAMELIST THEN CLOSEFILE
"NAMELIST STOP
MAKE "REC READ "NAMELIST
MAKE "NAME FIRST :REC
MAKE "CITY LAST :REC
PRINT (SE :NAME [LIVES IN] :CITY)
GO "1
END
```

Notice that when we create or re-create a file, we use REWRITE. When an existing file is to be accessed, we use RESET. These commands are like the BASIC commands to OPEN for output and OPEN for update. Suppose other names are to be added to the file. We'll then use the commands:

```
RESET "NAMELIST
SCANTOEND "NAMELIST
WRITE "NAMELIST [[JANE
ELIZABETH][NASHVILLE]]
  etc.
CLOSEFILE "NAMELIST
```

Definitions as Data

The LOGO primitives that make the introduction of files possible are TEXT and DEFINE. These procedures allow procedure definitions to be regarded and manipulated as data. This is an unusual capability for a programming language, a feature LOGO inherited from its precursor, LISP.

Files are implemented as procedure definitions, with each line of a definition containing one record of the file. (After closing a file, inspect its contents by entering TO *filename*.) When a file is rewritten or reset, the procedure definition is transformed into a pair of lists. The first list is empty, and the second contains the records in the file. As data is read from the file, records are transferred from the second list to the first. When the second list is empty, more data can be written into the file. New data is then entered at the end of the first list. The CLOSEFILE procedure combines the records in the two lists and redefines them into a procedure definition with the given file name.

It isn't necessary to know any of these details in order to use the files. The file procedures can be stored as a separate procedure file and given the name FILES. Then when we want file capability, FILES can be merged into the workspace. To save the data file you've created, erase all the procedures in the workspace *except* your data file. Then save it with an appropriate name. Later, you can merge into the workspace when necessary.

Besides allowing file capability, these file procedures are examples of the advanced programming that is possible with LOGO. After all, LOGO is a language for learning, even learning about files.

Muller's challenge and Kirchner's response are addressed to the problem of disk access in TI LOGO. Terrapin LOGO, however, offers a variety of ways to access the disk. The solution presented here—building a pseudo-file in the form of a procedure and then saving that procedure—is only one of the solutions possible in Terrapin LOGO. Another solution might use the DPRINT utility (*Technical Manual*, Section 7.3, pages 80-82), and there may be others as well.

Although Terrapin LOGO does allow procedures to access the disk, it is like TI LOGO in that it doesn't have any facilities for building files as such. For this reason we've adapted the method used in "Files in LOGO" for Terrapin LOGO.

The procedures for the two programs are very similar, and the operations of the programs are nearly identical. (One user-defined procedure in the TI LOGO program—LIST—is a primitive in Terrapin LOGO.) In the Terrapin LOGO version, you can save the procedures used to manipulate your file (along with the file) by saving the entire workspace, just as in the TI LOGO version. In Terrapin LOGO, however, you can save any single procedure (or selection of procedures) from the workspace. In order to save selected procedures, you need to load two files from the Utility Disk: PSAVE and ADDRESSES. (You have to load ADDRESSES because it defines the addresses used in PSAVE's procedures as variable names.) The file PSAVE (described in Section 4.1, page 47 of the manual) contains a procedure named PSAVE; to save a procedure (or a selection of procedures), give PSAVE a file name and a list of procedures as input:

```
PSAVE "DATAFILE [NAMELIST]
```

You can then erase NAMELIST from the workspace and save all the rest of the file manipulation procedures, including PSAVE and ADDRESSES, together. Another option is to save the utilities and the data file together, so that the utilities will always be with the file.

Just as in TI LOGO, the inputs for the data file are lists. A typical session from the command mode might run as follows:

```
REWRITE "NAMELIST
WRITE "NAMELIST [[ERNST
  KIRCHNER][ST. PAUL]]
```

```
WRITE "NAMELIST [[MARGARET MCCOURTNEY][NORTHFIELD]]
WRITE "NAMELIST [[C. Q. UMBER][SILICON GULCH]]
CLOSEFILE "NAMELIST
```

Those commands open a file, put elements in it, and then close it. The following commands add elements to the file:

```
RESET "NAMELIST
SCANTOEND "NAMELIST
WRITE "NAMELIST [[JANE ELIZABETH][NASHVILLE]]
etc.
CLOSEFILE "NAMELIST
```

To check the file, call TESTNAMES with NAMELIST as its input:

```
TESTNAMES "NAMELIST
```

This will print out the names and cities in simple sentences.

C64

The Apple and C-64 versions of Terrapin LOGO are very closely related. In fact, their procedures and program operation are virtually identical with one exception: The C-64 version lacks a utility, such as the Apple's PSAVE, which lets procedures access the disk system directly. In this one respect, Terrapin LOGO for the C-64 is exactly like TI LOGO. It is different from TI LOGO in another respect, however. In TI LOGO, you must erase from the workspace any procedures you don't want to save in a file, because the SAVE command saves *all* the procedures in the workspace. In Terrapin LOGO on the C-64, you can save any single procedure—or any combination of procedures—from the workspace. To save the procedure NAMELIST alone as a data file, for instance, you would type:

```
(SAVE "NAMELIST [NAMELIST])
```

This will save the *procedure* NAMELIST in the *file* NAMELIST on disk. If you wanted to save the file-handling utilities together in a file, you would type the following command:

```
(SAVE "FILEM [REWRITE RESET EOF WRITE FIND SCANTOEND CLOSE FILE])
```

This will save the procedures in a file called FILEM.

LOGO Systems LOGO for the IBM

The procedures we've developed for handling files in LOGO Systems' LOGO for the IBM are very different from those developed for TI LOGO or Terrapin LOGO. This is because IBM LOGO procedures are not only able to access the disk system, but also can create and manipulate true data files, both sequential and relative.

"It is possible to meet this challenge because LOGO is an extensible language."

In this LOGO program, the procedure MAKEFILE opens a sequential data file with the name you give it as input:

```
MAKEFILE "NAMELIST.DAT
```

(If you don't give your file name an extension, you must include the period—signifying the null extension—when you try to use ERASEFILE:

```
ERASEFILE "STUFF.
```

Otherwise, **ERASEFILE** won't recognize it as a valid file and won't erase it.)

The procedure GETINFO then prompts for a name, and follows that with a prompt for a city. With IBM LOGO, it's not necessary to enclose either input in brackets. SAVERECORD gathers the two inputs into one record (actually a list of two lists: [[ERNST KIRCHNER][ST. PAUL]], for instance). Once this is done, WRITERECORD appends it to the data file. Because WRITERECORD *appends* to the data file, you don't have to use procedures like RESET or SCANTOEND when you add to your file. If you want to add something, simply call MAKEFILE with the appropriate file name. When you've finished making entries in your file, you can exit by typing 999 in response to either prompt.

The procedure TESTNAMES, when given the data file name as input, reads data from the file and prints out simple sentences:

```
TESTNAMES "NAMELIST.DAT
```

It uses the procedures PRINTNAMES, READRECORD, and PRINTNAME. PRINTNAMES calls READRECORD, which first tests for the end of file condition using READEOFP. If the end of the file hasn't been reached, it then reads a record. PRINTNAME separates the record into its two component parts and prints them in a sentence.

Continued on p. 96

APPLE LOGO — APPLE II Series
COMMODORE LOGO — C-64

```
TO WRITE :FNAME :X
  TEST EOF :FNAME
  IF PRINT [NOT EOF] ST
OP
  MAKE "X FPUT :X []
  MAKE "F1 FIRST THING :
FNAME
  MAKE "F2 LAST THING :F
NAME
  MAKE "F1 LPUT :X :F1
  MAKE :FNAME LIST :F1 :
F2
END

TO FIND :FNAME
  MAKE "F1 FIRST THING :
FNAME
  MAKE "F2 LAST THING :F
NAME
  TEST EOF :FNAME
  IF PRINT [EOF] STOP
  MAKE "GET FIRST :F2
  MAKE "F2 BF :F2
  MAKE "F1 LPUT :GET :F1
  MAKE :FNAME LIST :F1 :
F2
  OUTPUT FIRST :GET
END
```

Continued on p. 96

Files In LOGO . . . from p. 95

APPLE LOGO — APPLE II Series

COMMODORE LOGO — C-64

```
TO SCANTOEND :FNAME↵
 1:↵
 IF EOF :FNAME THEN STO
P↵
 MAKE "GET FIND :FNAME↵
 GO "1↵
END↵

TO REWRITE :FNAME↵
 DEFINE :FNAME [[ ]]↵
 RESET :FNAME↵
END↵

TO EOF :FNAME↵
 OUTPUT LAST THING :FNA
ME =[ ]↵
END↵

TO RESET :FNAME↵
 MAKE :FNAME LIST [ ] BF
 TEXT :FNAME↵
END↵

TO TESTNAMES :FNAME↵
 RESET :FNAME↵
 1:↵
 IF EOF :FNAME THEN CLO
SEFILE :FNAME STOP↵
 MAKE "REC FIND :FNAME↵
 MAKE "NAME FIRST :REC↵
 MAKE "CITY LAST :REC↵
 PRINT (SE :NAME [LIVE
S IN] :CITY )↵
 GO "1↵
END↵

TO CLOSEFILE :FNAME↵
 MAKE "F1 FIRST THING :
FNAME↵
 MAKE "F2 LAST THING :F
NAME↵
 DEFINE :FNAME FPUT [ ]
SE :F1 :F2↵
 MAKE :FNAME [ ]↵
END↵
```

IBM LOGO — IBM PCjr

```
TO MAKEFILE :FNAME↵
OPEN :FNAME↵
SAVERECORD↵
CLOSE :FNAME↵
END↵

TO PRINTNAME↵
MAKE "F1 FIRST :REC↵
MAKE "F2 LAST :REC↵
PRINT (SE :F1 [LIVES IN
] :F2)↵
END↵

TO READRECORD :FNAME↵
MAKE "REC []↵
SETREAD :FNAME↵
IF READEOFP [SETREAD "C
ON STOP]↵
MAKE "REC READLIST↵
SETREAD "CON↵
END↵

TO PRINTNAMES↵
LABEL "LOOP↵
READRECORD :FNAME↵
IF :REC = [] [STOP]↵
PRINTNAME GO "LOOP↵
END↵

TO TESTNAMES :FNAME↵
OPEN :FNAME↵
PRINTNAMES↵
CLOSE :FNAME↵
END↵

TO MAKERECORD :ALIAS :C
ITY↵
MAKE "RECORD FPUT :CITY
[ ]↵
MAKE "RECORD FPUT :ALIA
S :RECORD↵
OUTPUT :RECORD↵
END↵

TO WRITERECORD :ALIAS :
CITY↵
SETWRITE :FNAME↵
PRINT MAKERECORD :ALIAS
 :CITY↵
SETWRITE "CON↵
END↵

TO FINISH↵
OUTPUT IF (OR :ALIAS =
[999] :CITY = [999]) ["
TRUE] ["FALSE]↵
END↵
```

IBM LOGO — IBM PCjr

```
TO GETINFO↵
MAKE "ALIAS [ ] MAKE "CI
TY [ ]↵
PRINT [TYPE IN NAME:]↵
MAKE "ALIAS READLIST↵
IF FINISH [STOP]↵
PRINT [TYPE IN CITY:]↵
MAKE "CITY READLIST↵
IF FINISH [STOP]↵
END↵

TO SAVERECORD↵
LABEL "LOOP↵
GETINFO↵
IF FINISH [STOP]↵
WRITERECORD :ALIAS :CIT
Y↵
GO "LOOP↵
END↵
```

TI·LOGO — TI·99/4A

```
TO SCANTOEND :FNAME↵
1:↵
IF EOF :FNAME THEN STOP
↵
MAKE "GET FIND :FNAME↵
GO "1↵
END↵

TO REWRITE :FNAME↵
DEFINE :FNAME [[ ]]↵
RESET :FNAME↵
END↵

TO LIST :X :Y↵
IF WORD? :X THEN MAKE "
X SE :X [ ]↵
IF WORD? :Y THEN MAKE "
Y SE :Y [ ]↵
OUTPUT FPUT :X FPUT :Y
[ ]↵
END↵

TO FIND :FNAME↵
MAKE "F1 FIRST THING :F
NAME↵
MAKE "F2 LAST THING :FN
AME↵
TEST EOF :FNAME↵
IFT PRINT [EOF ] STOP↵
MAKE "GET FIRST :F2↵
MAKE "F2 BF :F2↵
MAKE "F1 LPUT :GET :F1↵
MAKE :FNAME LIST :F1 :F
2↵
OUTPUT FIRST :GET↵
END↵

TO CLOSEFILE :FNAME↵
MAKE "F1 FIRST THING :F
NAME↵
MAKE "F2 LAST THING :FN
AME↵
DEFINE :FNAME FPUT [ ]
SE :F1 :F2↵
MAKE :FNAME [ ]↵
END↵

TO EOF :FNAME↵
OUTPUT LAST THING :FNAM
E =[ ]↵
END↵

TO RESET :FNAME↵
MAKE :FNAME LIST [ ] BF
 TEXT :FNAME↵
END↵

TO TESTNAMES :FNAME↵
RESET :FNAME↵
1:↵
IF EOF :FNAME THEN CLOS
EFILE :FNAME STOP↵
MAKE "REC FIND :FNAME↵
MAKE "NAME FIRST :REC↵
MAKE "CITY LAST :REC↵
PRINT (SE :NAME [LIVES
IN] :CITY )↵
GO "1↵
END↵

TO WRITE :FNAME :X↵
TEST EOF :FNAME↵
IFF PRINT [NOT EOF ] ST
OP↵
MAKE "X FPUT :X [ ]↵
MAKE "F1 FIRST THING :F
NAME↵
MAKE "F2 LAST THING :FN
AME↵
MAKE "F1 LPUT :X :F1↵
MAKE :FNAME LIST :F1 :F
2↵
END↵
```

HCM

ANIMATION CREATION

by W. K. Balthrop
HCM Staff

Name:	**Animation Creation**
Program Type:	**Graphics**
Machine:	**IBM PCjr**
Developer:	
Distributor:	**IBM Software** **P.O. Box 1328-S** **Boca Raton, FL 33432**
Price:	**$40.00, diskette**

System Requirements:
Disk Drive and controller of DOS 1.1 or higher, color monitor or television, Cartridge BASIC, 64K Memory Expansion

	poor	fair	good	excellent
Performance	■			
Ease of Use	■	■	■	
Documentation	■	■	■	■

When I was assigned to write the review of *Animation Creation*, I was excited about the opportunity to play around with the machine's graphics. After all, the PCjr is a very powerful graphics computer. But after I tried out the animation software, I was quite disappointed. This program does not take advantage of the PCjr's fine graphics resolution. In fact, it has no graphics except for the 256 predefined graphics characters resident in the character set. The screen is in text mode at all times, which might account for the fact that IBM manages to get eight 40-column screens into memory at the same time.

The program creates animation by changing the screens. Each successive screen is a little bit different from the one before it, so that when shown rapidly, they give the illusion of animation.

When you load the program, you are taken through a short routine that helps you adjust the color on your television or monitor and lets you align the picture on the screen. You are then led to the main menu screen, where you can select one of 4 options. Option 1, Create Existing Screens, will let you enter the screen edit mode and work on screens which have been saved on the disk drive. Options 2 and 3 let you Create New Screens in either 40- or 80- column text mode. When making pictures in 40-column mode you can create up to 8 different screens. In 80-column mode you can make 4 different screens. Option 4 lets you run the animation sequence by flipping through the screens like a deck of cards.

The Screen Editor

When the screen editor first appears for a new screen, the screen is blank except for line 25, which contains information about the 10 function keys. To move around the screen, use the arrow keys. The first time you press an arrow key it will set the direction of movement. Then, when you enter characters via the keyboard, they will repeat in the preset direction.

One of the most peculiar things I found in the program was its use of the [Fn] key in conjunction with the arrow keys. For example, to move the cursor to the *top* of the page, you must press the [Fn] key and the *left* arrow key. This system is confusing at best.

The function keys serve several purposes. You can assign any of the special graphics characters available in the PCjr's character set to function keys 1 through 6. Once a character is assigned to a function key, you can print that character by pressing its assigned function key. You can have up to 6 characters assigned to function keys at one time.

Ease of Use

The program is fairly easy to use once you get the function key sequences figured out. There really isn't much to learn because the program doesn't *do* very much. You type on the screen and use the arrow keys to move the cursor around. The function/arrow key assignments for moving the cursor all the way to the top, bottom, left or right sides of the screen seem backwards at first, but this doesn't take long to figure out. IBM doesn't suggest an age group for the software, but my guess is that it would be appropriate for ages 8 and up. Anyone younger than that would probably have trouble understanding the three key sequences needed for some of the functions.

Documentation

The program's 56-page manual is well-written and easy to understand and includes instructions on using the function keys and on loading and saving screens. The back of the manual includes several appendices that list all of the error messages and give you a list of special characters available. There is also a section entitled "Programming with BASIC." This section doesn't tell you how to interface with the screens in BASIC. It does tell you how to load a program that is written in Advanced BASIC (compatible with Cartridge BASIC) and how to list the program so that you can see how it is done.

A Dust Collector

The *Animation Creation* program was a big disappointment to me from the very beginning. You can create animation by flipping through the screens, but the resolution is on the character level, which is the lowest possible resolution of the computer. You cannot draw lines, plot points, paint the screen, or do any of the fantastic things that the system is capable of accomplishing. The program is poor in general, except as an example of what happens when you swap screens in and out. If you are desperate for entertainment, you might find some enjoyment with this program, but the majority of us will be buying a dust collector.

HCM

Doing Without Extended BASIC

At first glance, the TI-99/4A home computer by itself seems much less powerful than it is when augmented by the Extended BASIC cartridge. However, many of the tasks you thought possible only with Extended BASIC can be accomplished in BASIC with just a little more work. The following five commands found only in Extended BASIC can easily be converted into BASIC subroutines.

PI

The PI function in Extended BASIC is the easiest of the five to convert. All you need to do is add this statement near the beginning of the BASIC program and then use PI just as you would in Extended BASIC:

```
PI = 3.14159265359
```

DISPLAY AT

DISPLAY AT is an Extended BASIC comand that allows you to print information anywhere on the screen without scrolling the screen. It can be simulated (although it will be a bit slower) with the following routine:

```
100 FOR Z=0 TO LEN(A$)-1
110 CALL HCHAR(Y,X+Z,ASC(SEG$(A$,Z+1,1)))
120 NEXT Z
130 RETURN
```

Before branching to the routine with a GOSUB, you need to set up three variables. Set A$ to the string you want printed. The variable Y is the row number, and X is the column number. The length of the string A$ plus the value of X should never be more than 32. This routine cannot handle line wrapping.

RPT$

In Extended BASIC you can repeat a string and place the results in another string using the RPT$ function. You can also do this in BASIC with a simple routine:

```
100 A$=""
110 FOR Z=1 TO R
120 A$=A$&B$
130 NEXT Z
140 RETURN
```

Before using the subroutine you will need to set B$ equal to the string to be repeated and R equal to the number of times to repeat the string. The result will be returned in A$.

MAX and MIN

The last two functions we will cover here allow you to retrieve the lower or higher value from two variables. The MAX and MIN functions do this in Extended BASIC. Two short routines can simulate this in BASIC.

MAX

```
100 IF A<B THEN 130
110 C=A
120 GOTO 140
130 C=B
140 REM CONTINUE PROGRAM
```

MIN

```
100 IF A<B THEN 130
110 C=B
120 GOTO 140
130 C=A
140 REM CONTINUE PROGRAM
```

In these two programs the values in variables A and B are compared. In MAX the higher of the two is assigned to C. In MIN the lower of the two is assigned to C.

—William K. Balthrop

Protecting the Hi-Res Screen

The Apple II graphics system has three modes of operation: *Text* for displaying alphanumeric characters; *Low-Resolution (lo-res) Graphics* for displaying colored graphic blocks on the screen; and *Hi-Resolution (hi-res) Graphics* for displaying graphics that can be controlled point-by-point on the screen. The first two modes use the same memory areas for their displays. As long as your program uses text page one, located at memory locations 1024 to 2047 ($400-$7FF), no difficulties arise. If you use the third (hi-res) mode, however, and your program and variables take up over 6K-bytes of memory, your display is in danger of being destroyed by your program.

Applesoft programs are loaded beginning at 2048 ($800) and build upwards in memory. When a program is run, simple variables and arrays are stored in the memory directly following the program. If you write a reasonably long program (or one with very many variables) that uses hi-res graphics page one (located at 8192 to 16383 [$2000-$3000]), it will both interfere with the graphics and not operate as designed.

There are several approaches to solving this problem. The *Applesoft Programmer's Reference Manual* explains two ways to solve the problem *before* you RUN your program (see page 149). One method sets HIMEM: to 8192, which makes it impossible for the program to store variables on the hi-res page because it cannot access areas above the HIMEM: location. But this is acceptable only with short programs that require relatively small areas for variable storage.

With larger programs the manual suggests POKEing values into the page zero locations (103 and 104 decimal) that point to the beginning of Applesoft's program storage area, so that the program will be loaded above the hi-res screen. This method is perfectly satisfactory, except that each time you turn on the computer you have to do the three POKE statements before you can RUN the program.

Here are two other ways to protect the hi-res screen where the computer does the work when you want to RUN the program. One way uses a command file that you EXECute. The *CREATE LOMEM.EXEC* program in the "3-D IIE" article in this issue creates such a file. When you RUN this program, it creates a text file called LOMEM.EXEC. Once it has been created, you simply type

```
EXEC LOMEM.EXEC
```

When the computer EXECutes the text file, it "types" the POKEs for you and then RUNs the program.

Another way to accomplish these POKES is used in the Apple version of *Cannibals*. Here's line 170 of that program:

```
170 IF PEEK (103) < > 1 OR PEEK (104) < > 64 THEN POKE 103,1 : POKE 104,64 : POKE
    16384,0 : POKE 16385,0 : POKE 16386,0 : PRINT CHR$(13) CHR$(4) "RUN CANNIBAL"
```

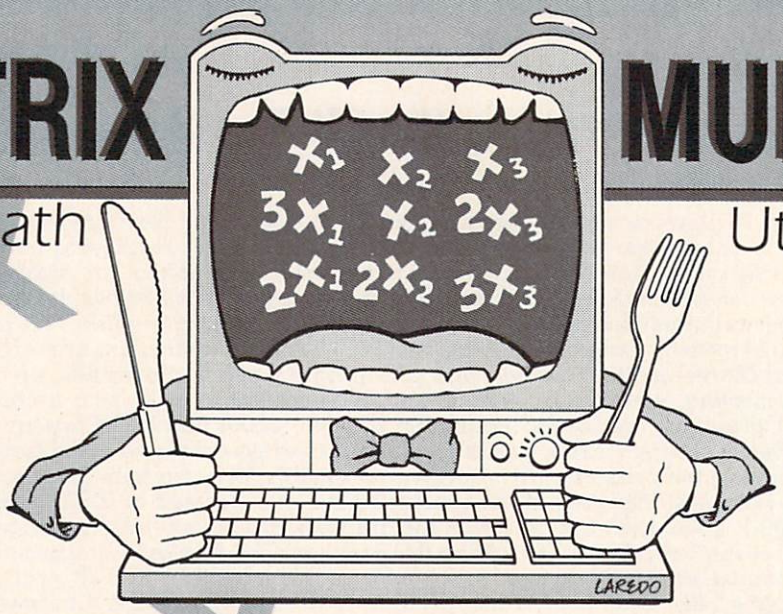
This line checks the page zero locations 103 and 104 to see if this program was loaded above the hi-res screen. If it was, then it simply goes on to line 180 and the rest of the program. If the page zero pointers were not already set to protect the hi-res screen, then the line POKEs the proper values into these locations, zeroes out the new beginning of the program area, and then gets DOS to RUN the program from disk. This means the entire program will be re-loaded above the hi-res screen and its variable storage won't interfere with the hi-res picture. Notice that this is done as the first executable instruction in the program, making this a fast and easy way to use the hi-res page in longer programs.

—Roger Wood

C64 VIC

MATRIX MUNCHER

A Math Utility



by the HCM Staff

In the course of their work, engineers, scientists, technicians, and even high school students often run into math problems involving several unknowns. These unknowns are usually related, and their relationships can be expressed with mathematical equations. When these equations are solved simultaneously, the values of the unknowns are often discovered. The pencil-and-paper method of solving simultaneous equations (usually learned in high school and soon forgotten) is time consuming and error prone.

Matrix Muncher, a short BASIC program for the C-64 and VIC-20, can work through the solution of simultaneous equations for you. It can handle up to nine unknowns (and equations) on the Commodore 64 and up to four unknowns on the VIC-20. All *you* have to do is produce the equations that represent the relationships among the unknowns.

A Simple Example

Let's assume that we have three equations containing three unknowns. The relations among them have been determined for us, and we want to know the values of the three unknowns. It's time for *Matrix Muncher* to go to work.

The three equations are:

$$\begin{aligned} X_1 + X_2 + X_3 &= 9 \\ 3X_1 + X_2 + 2X_3 &= 16 \\ 2X_1 + 2X_2 + 3X_3 &= 21 \end{aligned}$$

The information contained in these equations can also be expressed in matrix form as follows:

$$\begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 2 \\ 2 & 2 & 3 \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 16 \\ 21 \end{bmatrix}$$

coefficients *unknowns* *constants*

In general, for n equations, the matrices can be shown as:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_n \end{bmatrix}$$

This is shown in *matrix notation* as [A] × [X] = [B]. *Matrix Muncher* uses a matrix inversion technique to solve for the unknowns.

After loading the program and typing RUN, you'll see the following screen display:

```
M A T R I X   M U N C H E R

MATRIX INVERSION TECHNIQUE
TO SOLVE [A] × [X] = [B].

ENTER DEGREE OF THE MATRIX,
OR THE NUMBER OF EQUATIONS:
```

For our example, we enter the number 3 and press [RETURN]. The program asks for the coefficients to be entered row by row. After we enter five of the coefficients, the screen display looks like this:

```
THE COEFFICIENTS OF X
ARE IN THE "A" MATRIX

A(1,1),A(1,2),...,A(1,N)
A(2,1),A(2,2),...,A(2,N)
A(N,1),A(N,2),...,A(N,N)

INPUT THE MATRIX VALUES
ROW BY ROW.

A(1,1)
 ? 1
A(1,2)
 ? 1
A(1,3)
 ? 1

A(2,1)
 ? 3
A(2,2)
 ? 1
A(2,3)
 ?
```

After we've entered all values of the coefficients, *Matrix Muncher* requests the values of the constants (Bx).

```
NOW INPUT ELEMENTS OF B

B(1)
 ? 9
B(2)
 ? 16
B(3)
 ?
```

When the last value of the B matrix has been entered, the Magic Math Machine goes to work:

```
M U N C H
M U N C H
M U N C H

SOLUTION VALUES ARE:
X(1) = 2
X(2) = 4
X(3) = 3

** DONE **

READY
```

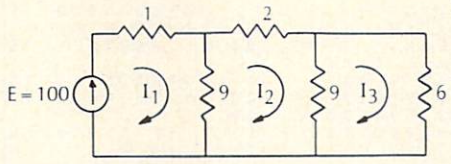
Each major step completed in the program causes the word MUNCH to scroll on the screen. After the unknowns have been found (or *Matrix Muncher* discovers that no unique solution exists), the results are displayed in the format above.

Go ahead—check the answers in the original three equations and see if they work!

Note: TI readers have been with us for a long time will recall the original TI version of Matrix Muncher by Cheryl Whitelaw and the 99'er HCM staff that appeared in the March, 1983 issue.

A Real Life Example

Electrical engineering applications include solving networks for voltages and currents. Here is a simple example using loop equations to solve for currents in a network. The sum of the voltages around a loop must be zero, according to Kirchoff's voltage law.



Given the network above, find the currents. The loop equations are:

$$\text{Loop 1: } -100 + 1I_1 + 9I_1 - 9I_2 = 0$$
$$\text{Loop 2: } -9I_1 + 9I_2 + 2I_2 + 9I_2 - 9I_3 = 0$$
$$\text{Loop 3: } -9I_2 + 9I_3 + 6I_3 = 0$$

Combining terms and rearranging, we get:

$$\begin{aligned} 10I_1 - 9I_2 &= 100 \\ -9I_1 + 20I_2 - 9I_3 &= 0 \\ -9I_2 + 15I_3 &= 0 \end{aligned}$$

In matrix form, this can be expressed:

$$\begin{bmatrix} 10 & -9 & 0 \\ -9 & 20 & -9 \\ 0 & -9 & 15 \end{bmatrix} * \begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix}$$

Once this information has been fed in, *Matrix Muncher* will produce the following values for the unknowns:

$$I1(X_1) = 22.46$$
$$I2(X_2) = 13.85$$
$$I3(X_3) = 8.31$$

You will find that the *Matrix Muncher* is many orders of magnitude faster than the pencil. So save time and have fun munching.

HCM

Matrix Muncher (C-64 and VIC-20) Explanation of the Program

Line Nos.	
100-160	Header.
170-180	Background and variable initialization.
190-250	Get size of equations.
260-370	Input [A] matrix values.
380-420	Input [B] matrix values.
440-610	Calculations to invert matrix.
620-710	Multiply inverse matrix by constant vector to solution vector; print results.
720-860	Subroutine to interchange rows if a diagonal element is zero.
870	End.

COMMODORE 64

```
100 REM  **********************
110 REM  *  MATRIX MUNCHER  *
120 REM  **********************
130 REM  BY
    THE  HCM  STAFF
140 REM  VERSION 4.2.1
150 REM  C64  BASIC
160 DIM A(9,9),X(9),Z(9,9)
170 PRINT "■SHIFT CLR■CTRL BLK■":POKE
    53281,1:POKE 53280,1
180 PRINT"■2CRSRDOWN■7CRSRRIGHT■M A T
    R I X  M U N C H E R"
190 PRINT"■4CRSRDOWN■7CRSRRIGHT■MATRIX
     INVERSION TECHNIQUE"
200 PRINT"■7CRSRRIGHT■TO SOLVE [A] * [X
    ] = [B]."
210 PRINT"■2CRSRDOWN■6CRSRRIGHT■ENTER
    DEGREE OF THE MATRIX,"
220 INPUT"■6CRSRRIGHT■OR THE NUMBER OF
    EQUATIONS:";N
230 IF (N<10)+(N>1)=-2 THEN 250
240 PRINT"■2CRSRDOWN■12CRSRRIGHT■N MUS
    T BE 1<N<10.":GOTO 210
250 PRINT"■6CRSRDOWN■10CRSRRIGHT■THE C
    OEFICIENTS OF X"
260 PRINT"■10CRSRRIGHT■ARE IN THE 'A' M
    ATRIX"
270 PRINT"■2CRSRDOWN■8CRSRRIGHT■A(1,1)
    ,A(1,2),...,A(1,N)"
280 PRINT"■8CRSRRIGHT■A(2,1),A(2,2),...
    ,A(2,N)"
290 PRINT"■8CRSRRIGHT■A(N,1),A(N,2),...
    ,A(N,N)"
300 PRINT"■2CRSRDOWN■8CRSRRIGHT■INPUT
    THE MATRIX VALUES":PRINT"■8CRSRRIGH
    T■ROW BY ROW.■2CRSRDOWN■"
310 FOR I=1TON
320 FOR J=1TON
330 I$=STR$(I):J$=STR$(J):P$="A("+MID$(
    I$,2,1)+","+MID$(J$,2,1)+")"
340 PRINT P$:INPUT A(I,J)
350 Z(I,J)=A(I,J)
360 NEXT J:PRINT" ":NEXT I
370 REM  INPUT MATRIX B
380 PRINT"■CRSRDOWN■NOW INPUT ELEMENTS
    OF B"
390 FOR I=1TON:I$=STR$(I):P$="B("+MID$(
    I$,2,1)+")"
400 PRINT P$:INPUT B(I)
410 NEXT I
420 PRINT"■SHIFT CLR■"
430 REM  INVERT MATRIX A
440 FOR L=1 TO N:PRINT TAB(16)"M U N C
    H"
450 IF Z(L,L)<>0 THEN 470
460 GOSUB 720
470 Z(L,L)=1/Z(L,L)
480 FOR K=1TON
490 IF (K-L)=0 THEN 550
500 Z(K,L)=Z(K,L)*Z(L,L)
510 FOR M=1TON
520 IF (M-L)=0 THEN 540
530 Z(K,M)=Z(K,M)-Z(K,L)*Z(L,M)
540 NEXT M
550 NEXT K
560 FOR M=1TON
570 IF (M-L)=0 THEN 590
580 Z(L,M)=-Z(L,L)*Z(L,M)
590 NEXT M
600 NEXT L
610 PRINT"■2CRSRDOWN■SOLUTION VALUES AR
    E:■CRSRDOWN■"
620 FOR I=1TON
630 X(I)=0
640 FOR J=1TON
650 X(I)=X(I)+Z(I,J)*B(J)
660 NEXT J
670 I$=STR$(I):P$="   X("+MID$(I$,2,1)+"
    )  =":PRINT P$;X(I)
680 NEXT I
690 PRINT"
700 PRINT"***  DONE  ***":END
710 REM  SUB TO SWITCH ROWS
720 FOR LL=L+1 TO N
730 IF Z(LL,L)=0 THEN 830
740 FOR M=1TON
750 DZ=Z(L,M)
760 Z(L,M)=Z(LL,M)
770 Z(LL,M)=DZ
780 NEXT M
790 DB=B(L)
800 B(L)=B(LL)
810 B(LL)=DB
820 RETURN
830 NEXT LL
840 PRINT"■2CRSRDOWN■SORRY, A DETERMINA
    NT = 0."
850 PRINT"THERE IS NO UNIQUE SOLUTION."
860 END
```

HCM

COMMODORE 64

```
530 Z(K,M)=Z(K,M)-Z(K,L)*Z(L,M)
540 NEXT M
550 NEXT K
560 FOR M=1TON
570 IF (M-L)=0 THEN 590
580 Z(L,M)=-Z(L,L)*Z(L,M)
590 NEXT M
600 NEXT L
610 PRINT"■2CRSRDOWN■SOLUTION VALUES AR
    E:■CRSRDOWN■"
620 FOR I=1TON
630 X(I)=0
640 FOR J=1TON
650 X(I)=X(I)+Z(I,J)*B(J)
660 NEXT J
670 I$=STR$(I):P$="   X("+MID$(I$,2,1)+"
    )  =":PRINT P$;X(I)
680 NEXT I
690 PRINT"
700 PRINT"***  DONE  ***":END
710 REM  SUB TO SWITCH ROWS
720 FOR LL=L+1 TO N
730 IF Z(LL,L)=0 THEN 830
740 FOR M=1TON
750 DZ=Z(L,M)
760 Z(L,M)=Z(LL,M)
770 Z(LL,M)=DZ
780 NEXT M
790 DB=B(L)
800 B(L)=B(LL)
810 B(LL)=DB
820 RETURN
830 NEXT LL
840 PRINT"■2CRSRDOWN■SORRY, A DETERMINA
    NT = 0."
850 PRINT"THERE IS NO UNIQUE SOLUTION."
860 END
```

HCM

VIC-20

```
100 REM  **********************
110 REM  *  MATRIX MUNCHER  *
120 REM  **********************
130 REM  BY
    THE  HCM  STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  V20 BASIC
170 DIM A(4,4),X(4),Z(4,4),B(4)
180 PRINT "■SHIFT CLR■CTRL BLK■":POKE
    36879,25
190 PRINT"■2CRSRDOWN■CRSRRIGHT■M A T R
    I X  MUNCHER"
200 PRINT"■2CRSRDOWN■CRSRRIGHT■VIA MAT
    RIX INVERSION"
210 PRINT"■2CRSRDOWN■ TO SOLVE [A]*[X]=
    [B]"
220 PRINT"■2CRSRDOWN■ENTER DEGREE OF":P
    RINT "THE MATRIX, OR THE"
230 INPUT"NUMBER OF EQUATIONS:";N
240 IF (N<5)+(N>1)=-2 THEN 260
250 PRINT"■2CRSRDOWN■N MUST BE 1<N<5.":
    GOTO 220
260 PRINT"■5CRSRDOWN■THE COEFICIENTS OF
     X"
270 PRINT"ARE IN THE 'A' MATRIX"
280 PRINT"■2CRSRDOWN■A(1,1),A(1,2),,A(1
    ,N)"
290 PRINT"A(2,1),A(2,2),,A(2,N)"
300 PRINT"A(N,1),A(N,2),,A(N,N)"
310 PRINT"■2CRSRDOWN■INPUT THE MATRIX:"
    PRINT"VALUES ROW BY ROW.■2CRSRDOWN■"
320 FOR I=1TON
330 FOR J=1TON
340 I$=STR$(I):J$=STR$(J):P$="A("+MID$(
    I$,2,1)+","+MID$(J$,2,1)+")"
350 PRINT P$:INPUT A(I,J)
360 Z(I,J)=A(I,J)
370 NEXT J:PRINT" ":NEXT I
380 REM  INPUT MATRIX B
```

Continued on p. 149

Sea of States . . . from p. 14

TI-99/4A

```
500 IF C<>-1 THEN S(1)=S(1)+A :: S(2)=S
    (2)+B :: GOTO 330 ELSE S(1)=S(1)-A
    :: S(2)=S(2)-B :: IF S(2)<0 THEN S(
    2)=0 :: GOTO 330 ELSE 330
510 A$="A GIANT TALKING OCTOPUS HAS AMB
    USHED YOU!!" :: C=3 :: SH=116 :: GO
    SUB 660
520 IF SB=1.5 THEN DISPLAY AT(15,1):"YO
    U ARE CAUGHT AND GET       CRUSHED B
    Y THE OCTOPUS":"YOU LOSE:" :: GOSUB
     1070 :: C=-1 :: GOTO 480
530 DISPLAY AT(15,1):" HE SAYS ""I WILL
     NOT LET YOUGO EASILY WITHOUT ANSWE
    RING THIS QUESTION:""" :: SEA(S(5),
    S(6),S(7))=1.5
540 S(4)=S(4)+1 :: ST=INT(RND*50)+1 ::
    GOSUB 1070 :: IF RND>.5 THEN 590
550 ST=INT(RND*50)+1 :: DISPLAY AT(15,1
    ):" WHAT IS THE CAPITAL OF:":ST$(ST
    ,1): :: ACCEPT AT(17,1)BEEP:A$
560 IF A$=ST$(ST,2)THEN A$=ST$(ST,1)::
    GOTO 600
570 DISPLAY AT(15,1):" SORRY BUT THAT'S
     WRONG,":ST$(ST,2)&" IS THE CAPITOL
    ":"OF "&ST$(ST,1)&". IT FIGHTS":"YO
    U AND YOU LOSE:"
580 GOSUB 1070 :: C=-1 :: GOTO 480
590 DISPLAY AT(15,1):ST$(ST,2):"IS THE
    CAPITOL OF WHAT STATE" :: ACCEPT AT
    (17,1)BEEP:A$
600 IF A$<>ST$(ST,1)THEN 570 ELSE S(3)=
    S(3)+1
610 CALL MOTION(#1,0,-9):: DISPLAY AT(1
    5,1):" CORRECT, NOT ONLY DOES THE O
    CTOPUS RELEASE YOU BUT IT ALSO GIV
    ES YOU:" :: GOTO 480
620 A$="SHARKS ENCIRCLE YOU, IN AN ATT
    EMPT TO GET AWAY YOU LOSE" :: C=2 :
    : SH=120 :: GOSUB 660 :: C=-1 :: GO
    TO 480
630 A$="YOU HAVE FOUND A WRECK AND:" ::
     C=7 :: SH=124 :: GOSUB 660 :: SEA(
    S(5),S(6),S(7))=5 :: GOTO 480
640 A$="YOU ARE AT A DIVING BELL, A SAF
    E WAY TO GO UP." :: C=7 :: SH=128 :
    : GOSUB 660 :: GOTO 330
650 A$="YOU HAVE FOUND A WRECK. IT HAS
     ALREADY BEEN SALVAGED" :: C=7 :: S
    H=124 :: GOSUB 660 :: GOTO 330
660 CALL DELSPRITE(#1):: DISPLAY AT(15,
    1):A$ :: IF C<>5 THEN CALL SPRITE(#
    1,SH,C,161,200,0,-6):: GOTO 680
670 CALL SPRITE(#1,SH,C,161,200,0,-6,#3
    ,SH,C,161,231,0,-6,#4,SH,C,161,4,0,
    -6)
680 CALL PATTERN(#2,140):: FOR D=1 TO 2
    0 :: NEXT D :: CALL PATTERN(#2,136)
    :: FOR D=1 TO 10 :: NEXT D :: CALL
    POSITION(#1,A,B)
690 IF B>104 THEN 680 ELSE CALL MOTION(
    #1,0,0,#3,0,0,#4,0,0):: RETURN
700 REM ATTEMPT TO GO UP
710 RESTORE 1240 :: READ A,A$ :: CALL C
    HAR(A,A$,142,"0",143,"0"):: IF SB=4
     THEN 790 ELSE CALL PATTERN(#2,140)
    :: CALL MOTION(#2,-4,0)
```

TI-99/4A

```
720 DISPLAY AT(15,1):"YOU'VE ATTEMPTED
    TO GO UP    WITHOUT THE SAFETY OF A
    DIVING BELL AND:"
730 CALL POSITION(#2,A,B):: IF A>55 THE
    N CALL COLOR(#2,RND*10+6):: GOTO 73
    0 ELSE CALL MOTION(#2,0,0)
740 DISPLAY AT(15,1):"THE NITROGEN IN Y
    OUR BLOOD  FORMS BUBBLES AND YOU'RE
     DEAD!": :
750 CALL MOTION(#2,0,0):: GOSUB 1070 ::
     CALL CLEAR :: CALL COLOR(9,8,8)::
    CALL HCHAR(1,1,96,192):: RESTORE 12
    20 :: READ A,A$
760 CALL CHAR(A,A$):: CALL SPRITE(#1,13
    2,12,25,10,0,2,#3,140,15,25,15,0,2,
    #4,120,2,55,200,0,-40)
770 CALL POSITION(#4,A,B):: IF B>110 TH
    EN 770 ELSE CALL MOTION(#4,0,0):: G
    OSUB 1070 :: CALL MOTION(#1,0,0,#3,
    0,0)
780 DISPLAY AT(15,1):"YOUR FRIEND TRIED
     TO RESCUE YOU BUT THE SHARKS GOT T
    O    YOU FIRST:" :: GOSUB 1070 :: GO
    SUB 1070 :: GOTO 960
790 IF S(7)=2 AND S(3)>9 THEN 840 ELSE
    IF S(7)=1 AND S(3)>4 THEN 810 ELSE
    RESTORE 1250 :: READ A$ :: CALL CHA
    R(140,A$)
800 DISPLAY AT(15,1):"WAIT!  YOU HAVEN'
    T ANSWERED":STR$(5*S(7))&" QUES. CO
    RRECTLY ON LEVEL":STR$(S(7)):: GOSU
    B 1070 :: GOTO 330
810 CALL SPRITE(#1,128,C,160,100):: CAL
    L MOTION(#2,-3,0,#1,-3,0):: DISPLAY
     AT(15,1):"YOU GET INTO THE DIVING
    BELLAND GO:" :: S(7)=2
820 FOR A=5 TO 15 STEP 5 :: DISPLAY AT(
    17,A):"UP," :: GOSUB 1070 :: NEXT A
     :: DISPLAY AT(20,1):"YOU'RE AT LEV
    EL 2" :: GOSUB 1070
830 CALL DELSPRITE(#1,#2):: CALL SPRITE
    (#2,136,15,160,100):: RESTORE 1250
    :: READ A$ :: CALL CHAR(140,A$):: G
    OTO 330
840 CALL CLEAR :: CALL HCHAR(1,1,96,192
    ):: CALL SPRITE(#1,128,7,160,100)::
     CALL MOTION(#2,-2,0,#1,-2,0)
850 CALL POSITION(#2,A,B):: IF A>44 THE
    N 850
860 CALL MOTION(#2,0,0):: CALL PATTERN(
    #2,140):: CALL DELSPRITE(#1):: REST
    ORE 1280 :: FOR B=1 TO 9 :: READ A,
    A$ :: CALL CHAR(A,A$):: NEXT B
870 CALL SPRITE(#1,108,2,40,100):: FOR
    A=1 TO 20 :: CALL SOUND(-100,-4,0):
    : CALL COLOR(#1,2):: CALL COLOR(#1,
    16):: NEXT A
880 CALL COLOR(#1,7):: CALL PATTERN(#1,
    112):: CALL SOUND(200,-6,0):: CALL
    SOUND(1,-6,0):: CALL SPRITE(#1,116,
    16,20,200,0,-3)
890 CALL PATTERN(#1,120):: CALL PATTERN
    (#1,116):: CALL POSITION(#1,A,B)::
    IF B>100 THEN 890
900 CALL CHAR(127,"0",131,"0"):: CALL M
    OTION(#1,0,0,#2,-1,0)
```

Sea of States . . . from p. 15

COMMODORE 64

```
600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
610 DATA 0,0,0,0,0,0,0,0,0,0,8,0,0,16,0
    ,0
620 DATA 32,1,128,32,19,194,36,31,224,3
    6,31,224,32,31,226,32
630 DATA 31,224,16,31,225,185,19,192,24
    ,1,128,8,0,8,0,0
640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
650 DATA 0,0,0,8,0,0,24,0,0,60,6,7,255,
    12,15,255
660 DATA 216,29,255,248,7,255,240,31,25
    5,248,0,0,24,0,0,12
670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1
690 DATA 224,0,3,192,0,15,128,0,31,0,0,
    63,192,0,127,192
700 DATA 0,127,255,252,127,192,0,63,192
    ,0,31,0,0,15,128,0
710 DATA 3,192,0,1,224,0,0,0,0,0,0,0,0,
    0,0,0
720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    18
730 DATA 32,0,9,64,0,5,128,0,15,32,0,19
    ,192,160,6,128
740 DATA 64,10,64,160,17,32,0,0,0,20,0,
    0,8,10,64,20
750 DATA 4,128,0,3,64,0,4,128,0,10,64,0
    ,0,0,0,0
760 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
```

COMMODORE 64

```
770 DATA 0,0,0,0,0,0,0,0,0,0,0,0,24,0,0
    ,60
780 DATA 0,0,126,0,0,255,0,1,255,128,3,
    255,192,7,255,224
790 DATA 15,255,240,31,255,248,63,255,2
    52,127,255,254,255,255,255,0
800 DATA 0,0,0,0,0,72,0,0,16,0,0,48,0,4
    ,40,0
810 DATA 8,96,0,16,224,0,33,224,0,67,22
    4,0,199,192,1,63
820 DATA 128,2,63,128,4,127,128,8,255,0
    ,1,254,0,3,252,0
830 DATA 3,248,0,1,240,0,0,224,0,0,0,0,
    0,0,0,0
840 DATA 0,0,0,0,0,16,0,0,80,0,0,60,0,4
    ,32,0
850 DATA 8,96,0,0,224,0,1,224,0,67,224,
    0,199,192,1,63
860 DATA 128,2,63,128,0,127,128,0,255,0
    ,1,254,0,3,252,0
870 DATA 3,248,0,1,240,0,0,224,0,0,0,0,
    0,0,0,0
880 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    255
890 DATA 248,72,2,0,48,7,0,192,31,255,1
    36,63,255,0,63,254
900 DATA 0,31,248,0,15,240,0,0,0,0,0,0,
    0,0,0,0
910 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0
920 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,32
    ,198
930 DATA 64,128,2,0,52,7,0,192,31,255,1
    60,63,255,0,63,254
```

Sea of States

TI-99/4A

```
910 CALL PATTERN(#1,128)::  CALL PATTERN
(#1,124)::  CALL POSITION(#2,A,B)::
IF A>20 THEN 910 ELSE CALL DELSPRIT
E(#2)::  CALL PATTERN(#1,116)
920 CALL MOTION(#1,0,-4)::  CALL PATTERN
(#1,120)::  CALL PATTERN(#1,116)::  C
ALL POSITION(#1,A,B)::  IF B>10 THEN
 920 ELSE CALL DELSPRITE(#1)
930 DISPLAY AT(10,9):"CONGRATULATIONSI"
::  "YOU WENT TO AND RETURNED  UNM
OLESTED FROM:" ::  GOSUB 1070
940 DISPLAY AT(15,10):"THE SEA OF STATE
S" ::  GOTO 960
950 CALL CLEAR ::  DISPLAY AT(15,5):"YOU
 HAVE NO OXYGEN!!"
960 CALL DELSPRITE(ALL)::  CALL SPRITE(#
2,136,15,100,10,0,1)::  GOSUB 1070 :
 CALL CLEAR
970 PRINT "GAME RATING="&STR$(S(1)+S(2)
+S(3)-S(4)-20)::"GOLD COLLECTED="&ST
R$(S(2))::"OXYGEN REMAINING="&STR$(S
(1))
980 IF S(4)<>0 THEN PRINT "YOU GOT "&ST
R$(S(3)/S(4)*100)&"% RIGHT." ELSE P
RINT "(NO QUESTIONS ANSWERED)"
990 PRINT ::  ::  ::  ::  ::"DO YOU CARE TO
PLAY AGAIN,":NAME$;"?"
1000 CALL KEY(3,A,B)::  IF B=0 THEN 1000
ELSE IF A=89 THEN CALL CLEAR ::  RUN
 ELSE IF A<>78 THEN 1000 ELSE END
1010 CALL DELSPRITE(#1,#3,#4)::  ON INT(R
ND*4+1)GOTO 1020,1030,1040,1050
1020 A$="THE CORAL IS TOO HIGH HERE—" ::
 SH=108 ::  C=10 ::  GOTO 1060
1030 A$="A GIANT CLAM BLOCKS YOU—" ::  RE
STORE 1260 ::  READ B$ ::  CALL CHAR(
132,B$)::  SH=132 ::  C=16 ::  GOTO 10
60
1040 A$="YOU LOOK DOWN AND SEE A  BOT
TOMLESS PIT—" ::  CALL CHAR(132,RPT$
(RPT$("0",24)&RPT$("F",8),2))::  C=5
 ::  SH=132 ::  GOTO 1060
1050 A$="STING RAYS!!" ::  C=2 ::  RESTO
RE 1270 ::  READ B$ ::  CALL CHAR(132
,B$)::  SH=132
1060 CALL HCHAR(15,1,32,256)::  DISPLAY A
T(18,1):"BETTER GO ANOTHER WAY!"
::  GOSUB 660 ::  GOTO 350
1070 FOR D=1 TO 500 ::  NEXT D ::  RETURN
1080 DATA ALABAMA,MONTGOMERY,ALASKA,JUNE
AU,ARIZONA,PHOENIX,ARKANSAS,LITTLE
ROCK,CALIFORNIA,SACRAMENTO,COLORADO
,DENVER,CONNECTICUT,HARTFORD
1090 DATA DOVER,DELAWARE,DOVER,FLORIDA,TALLAHA
SSEE,GEORGIA,ATLANTA,HAWAII,HONOLUL
U,IDAHO,BOISE,ILLINOIS,SPRINGFIELD,
INDIANA,INDIANAPOLIS,IOWA
1100 DATA DES MOINES,KANSAS,TOPEKA,KENTU
CKY,FRANKFORT,LOUISIANA,BATON ROUGE
,MAINE,AUGUSTA,MARYLAND,ANNAPOLIS,M
ASSACHUSETTS,BOSTON,MICHIGAN
1110 DATA LANSING,MINNESOTA,ST. PAUL,MIS
SISSIPPI,JACKSON,MISSOURI,JEFFERSON
 CITY,MONTANA,HELENA,NEBRASKA,LINCO
LN,NEVADA,CARSON CITY
```

TI-99/4A

```
1120 DATA NEW HAMPSHIRE,CONCORD,NEW JERS
EY,TRENTON,NEW MEXICO,SANTE FE,NEW
YORK,ALBANY,NORTH CAROLINA,RALEIGH,
NORTH DAKOTA,BISMARK,OHIO
1130 DATA COLUMBUS,OKLAHOMA,OKLAHOMA CIT
Y,OREGON,SALEM,PENNSYLVANIA,HARRISB
URG,RHODE ISLAND,PROVIDENCE,SOUTH C
AROLINA,COLUMBIA,SOUTH DAKOTA
1140 DATA PIERRE,TENNESSEE,NASHVILLE,TEX
AS,AUSTIN,UTAH,SALT LAKE CITY,VERMO
NT,MONTPELIER,VIRGINIA,RICHMOND,WAS
HINGTON,OLYMPIA,WEST VIRGINIA
1150 DATA CHARLESTON,WISCONSIN,MADISON,W
YOMING,CHEYENNE
1160 DATA 96,00,104,000000000001000308020
0000026418284C0000000020C0804040110A0
40A11
1170 DATA 108,0001010303070F0F1F1F3F3F
7F7FFF0080080C0C0E0E0F0F0F8F8FCFCFEF
EFFFF
1180 DATA 112,0,113,00000000020F07C7E0,11
5,0008787060C0FC78,116,0000001070F09
0900D090F1F15754D6825000E0F0F8C8C8E
8C8F8F8382CE4C673
1190 DATA 120,000000001030F1B7F073F010000
0000000000808080E1F2FEFDF0E0
1200 DATA 124,0000020202020202020232FFF7
E3773F8080848484848484E4A4FFFFFFFF
EFC
1210 DATA 128,0F0F1F2D7F7E7C7878C7E7F3F
1F0F0F0F0F0F8B4FE7E3E1E1E3E7EFEFCF8F
0F0
1220 DATA 132,0,133,0000000006073F3F1F,134
,0,135,0000000006FEFCF8
1230 DATA 136,0000000003032E033F2000000
0000000000000002008E0F8F8E01C
1240 DATA 140,00000000000000040E15040A0A
,142,0,143,0
1250 DATA 0002000000003F21030C10180000
0000008000000E0F8F8E01C
1260 DATA 0000C0F0703010000000000FFFF1F
0700000C0F0F8F8783381C0C04FEFEF0C0
1270 DATA 0000000000018273F8F0703000000000
0000000000000C0FFC380
1280 DATA 108,0000000000000101,109,0,110
,0,111,0,112,01C1310D070E1CF81C0E07
0D31C1010000061860C0E0703E70E0C0601
806
1290 DATA 116,01020E1019F1F3F5F5C5E7B01
02000008 3C1828408000000 90E0C080
1300 DATA 120,0000001039 4F9F3F5F5C5E7B01
02000001408142040800000 90E0C08
1310 DATA 124,E3020233E4FFF123F040404040404
0404040C0000010AFC0A1,128,0E020233E4FF
F123F040404040404043C0000404FF0404
```

HCM

Sea of States

COMMODORE 64

```
940 DATA 0,31,248,0,15,240,0,0,0,0,0,0,
0,0,0,0
950 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0
960 DATA ALABAMA,MONTGOMERY,ALASKA,JUNE
AU,ARIZONA,PHOENIX,ARKANSAS,LITTLE
ROCK
970 DATA CALIFORNIA,SACRAMENTO,COLORADO
,DENVER,CONNECTICUT,HARTFORD,DELAWA
RE
980 DATA DOVER,FLORIDA,TALLAHASSEE,GEOR
GIA,ATLANTA,HAWAII,HONOLULU,IDAHO,B
OISE
990 DATA ILLINOIS,SPRINGFIELD,INDIANA,I
NDIANAPOLIS,IOWA,DES MOINES,KANSAS
1000 DATA TOPEKA,KENTUCKY,FRANKFORT,LOUI
SIANA,BATON ROUGE,MAINE,AUGUSTA
1010 DATA MARYLAND,ANNAPOLIS,MASSACHUSET
TS,BOSTON,MICHIGAN,LANSING,MINNESOT
A
1020 DATA ST. PAUL,MISSISSIPPI,JACKSON,M
ISSOURI,JEFFERSON CITY,MONTANA,HELE
NA
1030 DATA NEBRASKA,LINCOLN,NEVADA,CARSON
 CITY,NEW HAMPSHIRE,CONCORD,NEW JER
SEY
1040 DATA TRENTON,NEW MEXICO,SANTA FE,NE
W YORK,ALBANY,NORTH CAROLINA,RALEIG
H
1050 DATA NORTH DAKOTA,BISMARK,OHIO,COLU
MBUS,OKLAHOMA,OKLAHOMA CITY,OREGON
1060 DATA SALEM,PENNSYLVANIA,HARRISBURG,
RHODE ISLAND,PROVIDENCE,SOUTH CAROL
INA
```

COMMODORE 64

```
1070 DATA COLUMBIA,SOUTH DAKOTA,PIERRE,T
ENNESSEE,NASHVILLE,TEXAS,AUSTIN,UTA
H
1080 DATA SALT LAKE CITY,VERMONT,MONTPEL
IER,VIRGINIA,RICHMOND,WASHINGTON
1090 DATA OLYMPIA,WEST VIRGINIA,CHARLEST
ON,WISCONSIN,MADISON,WYOMING,CHEYEN
NE
1100 GOSUB 1400:GOSUB 1440
1111 PRINT "■SHIFT CLR■":POKE 53269,0:AP
=0
1120 INPUT "WHAT IS YOUR NAME";N$
1130 S(1)=36:S(7)=1:SS=21504:S(3)=0:S(2)
=0:S(4)=0
1140 FOR I=1TO2:FOR J=1TO8:FOR X=1TO8
1150 Z=INT(RND(0)*4)+1:ON Z GOSUB 1170,1
180,1180,1180
1160 NEXT X,J,I:GOTO 1230
1170 SEA(X,J,I)=1:RETURN
1180 Z=INT(RND(0)*4)+1:ON Z GOTO 1190,11
70,1200,1200
1190 SEA(X,J,I)=2:RETURN
1200 Z=INT(RND(0)*4)+1:ON Z GOTO 1210,12
20,1220,1220
1210 SEA(X,J,I)=3:RETURN
1220 Z=INT(RND(0)*6)+4:SEA(X,J,I)=Z:RETU
RN
1230 S(6)=INT(RND(0)*8)+1:S(5)=INT(RND(0
)*8)+1
1240 IF AP=2 THEN 1280
1250 PRINT "■SHIFT CLR■■2 1 C R S R D O W N■"
1260 PRINT "■CMDR RED■■4 0 SHIFT A■";"■CTR
L WHT■"
```

Continued on p. 104

Sea of States . . . from p. 103

COMMODORE 64

```
1270 POKE 53250,150:POKE 53251,210:POKE
     53269,2
1280 GOSUB 1640
1290 GOSUB 1570
1300 IF X=36 THEN GOSUB 1710:GOTO 1280
1310 IF X=30 THEN 2170
1320 F2=0:F1=0:IF CL=0 AND RW=0 THEN 129
     0
1330 IF S(5)=8 AND RW=1 THEN RW=0:F1=1
1340 IF S(5)=1 AND RW=-1 THEN RW=0:F1=1
1350 IF S(6)=8 AND CL=1 THEN CL=0:F1=1
1360 IF S(6)=1 AND CL=-1 THEN CL=0:F1=1
1370 S(5)=S(5)+RW:S(6)=S(6)+CL:AP=SEA(S(
     5),S(6),S(7))
1380 ON INT(AP) GOSUB 1790,1950,1970,202
     0,2040,2120,2100,2110,2030
1390 GOTO 1240
1400 PRINT "■SHIFT CLR■■CMDR RED■":POKE
     53269,0
1410 FOR I=1TO5
1420 PRINT "■SHIFT CRSRUP■■4■SHIFT A■":NE
     XT
1430 PRINT "■CTRL WHT■":RETURN
1440 POKE 34809,241:POKE 34810,242:POKE
     53288,1:POKE 53289,7
1450 POKE 53277,7:POKE 53253,75:POKE 532
     52,0:POKE 53269,4
1460 REM HEADER SCREEN
1470 FOR I=1TO64:FOR J=1TO100:NEXT
1480 POKE 53252,I:NEXT
1490 POKE 53250,70:POKE 53251,90:POKE 53
     288,12
1500 PRINT "■5CRSRDOWN■YOU ARE AT THE BO
     TTOM OF THE SEA"
1510 PRINT " IN SEARCH OF GOLD.":PRINT"YO
     U WILL ALSO FIND:"
1520 PRINT "■CRSRDOWN■SHARKS (HATED AND
     FEARED).":PRINT"UNDERWATER WRECKS."
1530 PRINT "VERY SMART OCTOPI.":PRINT"MU
     CH MUCH MORE."
1540 POKE 53269,6:FOR I=90 TO 250:FOR J=
     1TO100:NEXT
1550 POKE 53251,I:IF I/2=INT(I/2)THEN PO
     KE 53250,PEEK(53250)+1
1560 NEXT:RETURN
1570 CL=0:RW=0:F1=0
1580 X=PEEK(197):IF X=64 THEN RETURN
1590 IF X=39 THEN RW=-1
1600 IF X=13 THEN RW=1
1610 IF X=14 THEN CL=1
1620 IF X=9 THEN CL=-1
1630 RETURN
1640 PRINT "■HOME■OXYGEN GOLD #R
     IGHT #QUES."
1650 PRINT "
     "
1660 PRINT "■SHIFT CRSRUP■ ";S(1);SPC(6)
     S(2);SPC(8)S(3);SPC(7)S(4):GOSUB 16
     90
1670 PRINT "■HOME■8CRSRDOWN■WHAT NOW ";N
     $;"?"
1680 PRINT "■2CRSRDOWN■N S E W M U":RET
     URN
1690 PRINT "■HOME■6CRSRDOWN■":FOR I=1TO1
     1
1700 PRINT "
     ":NEXT:RETURN
1710 GOSUB 1690:PRINT"■2HOME■■7CRSRDOWN■
     "TAB(27)"N":PRINT TAB(24)"123456678"
```

COMMODORE 64

```
1720 FOR I=1TO8:POKE (I*40)+34135,I+48:P
     OKE (I*40)+34135+SS,1:NEXT
1730 PA=34135+(40*S(5))+S(6)
1740 POKE PA,64:POKE PA+SS,1
1750 FOR I=1 TO 5000:NEXT
1760 GOSUB 1690:RETURN
1770 GET C$:IF C$<>"" _THEN 1770
1780 RETURN
1790 GOSUB 1690:PRINT "■7SHIFT CRSRUP■YO
     U'RE AMBUSHED BY A GIANT TALKING
     OCTOPUS"
1800 POKE 53287,0:GOSUB 2640:IF AP=1 THE
     N SEA(S(5),S(6),S(7))=1.5:GOTO 1830
1810 IF F1=1 THEN GOSUB 2090
1820 PRINT "HE HAS CRUSHED YOU AND YOU L
     OSE:":F2=1:GOTO 2750
1830 PRINT"HE SAYS,'I MAY LET YOU GO, B
     UT IT WILL"
1840 PRINT"NOT BE EASY FOR YOU. CAN YOU
     ":PRINT"ANSWER THIS QUESTION?'"
1850 S(4)=S(4)+1:S1=INT(RND(0)*50)+1:ON
     INT(RND(0)*2)+1 GOTO 1860,1910
1860 GOSUB 1770:PRINT ST$(S1,2);" IS THE
     CAPITAL OF";:INPUT"WHAT STATE";S$
1870 IF S$=ST$(S1,1) THEN 1930
1880 GOSUB 1690:PRINT"■7SHIFT CRSRUP■YOU
     ARE WRONG."
1890 PRINT ST$(S1,2);" IS THE CAPITAL OF
     ";ST$(S1,1):PRINT"THE GIANT OCTOPU
     S"
1900 PRINT"REACHES OUT AND GRABS YOU AND
     YOU LOSE:":F2=1:GOTO 2750
1910 GOSUB 1770
1920 PRINT"WHAT IS THE CAPITAL OF ";ST$(
     S1,1):INPUT S$:IFS$<>ST$(S1,2)THEN
     1880
1930 S(3)=S(3)+1:GOSUB 1690:PRINT"■7SHIF
     T CRSRUP■THAT IS CORRECT."
1940 PRINT"THE OCTOPUS RELEASES YOU AND
     ":PRINT"GIVES YOU:":F2=1:GOTO 2830
1950 GOSUB1690:PRINT"■7SHIFT CRSRUP■YOU
     HAVE FOUND A DIVING BELL.":IFF1=1TH
     ENGOSUB2090
1960 POKE 53287,2:GOSUB 2640:RETURN
1970 GOSUB 1690:PRINT"■7SHIFT CRSRUP■YOU
     HAVE FOUND A WRECK":POKE 53287,0:G
     OSUB 2640
1980 IF AP=3 THEN SEA(S(5),S(6),S(7))=3.
     5:GOTO 2010
1990 PRINT"BUT■CRSRRIGHT■THIS WRECK HAS
     BEEN SALVAGED.":IF F1=1 THEN GOSUB
     2090:RETURN
2000 S(1)=S(1)-1:FOR I=1 TO 2500:NEXT:RE
     TURN
2010 PRINT "YOU SALVAGE IT AND GAIN:":F2
     =0:GOTO 2830
2020 C$="YOU HAVE COME ACROSS A GIANT CL
     AM.":POKE 53287,1:GOTO 2050
2030 C$="THE CORAL PYRAMID IS NICE BUT "
     :POKE 53287,7:GOTO 2050
2040 C$="THERE IS NOTHING HERE BUT SAND
     & SHELLS.":POKE 53287,1
2050 GOSUB 1690:PRINT"■7SHIFT CRSRUP■";C
     $:GOSUB 2640
2060 PRINT "SORRY NO TREASURE HERE."
2070 GOSUB 2730:IF F1<>0 THEN 2090
2080 FOR I=1TO2500:NEXT:RETURN
2090 PRINT"BETTER GO ANOTHER WAY.":FOR I
     =1TO2500:NEXT:RETURN
```

Sea of States

COMMODORE 64

```
2100 C$="STINGRAYS ABOUND.":POKE 53287,0
     :GOTO 2050
2110 C$="NOTHING HERE BUT STARFISH":POKE
     53287,1:GOTO 2050
2120 GOSUB 1690
2130 PRINT"■6SHIFT CRSRUP■YOU ARE SURROU
     NDED BY SHARKS.":POKE 53287,0:GOSUB
     2640
2140 IF F1=1 THEN PRINT"NEXT TIME GO ANO
     THER WAY."
2150 PRINT "IN AN ATTEMPT TO GET AWAY YO
     U LOSE:"
2160 FOR I=1TO2500:NEXT:F2=2:GOTO 2760
2170 GOSUB 1690:GOSUB 1400:IF AP=2 THEN
     2220
2180 GOSUB 1690:PRINT"■7SHIFT CRSRUP■YOU
     SHOULD NOT HAVE TRIED TO GO UP"
2190 PRINT"WITHOUT A DIVING BELL. YOU N
     OW HAVE":PRINT"THE BENDS."
2200 PRINT"SORRY, BUT YOU LOSE CONSIOUSN
     ESS AND":PRINT "DROWN."
2210 GOTO 2860
2220 IF S(3)>4*S(7) THEN 2260
2230 N=S(7)*5:S(1)=S(1)-1
2240 PRINT"YOU CANNOT GO UP UNTIL YOU HA
     VE":PRINT"ANSWERED";N;"QUESTIONS."
2250 FOR I=1TO2500:NEXT:GOTO 1250
2260 POKE 53252,0:POKE 53253,75:POKE 532
     69,5
2270 FOR I=1TO145:FORJ=1TO50:NEXT:POKE 5
     3252,I:NEXT
2280 FOR I=1TO16:POKE(40*I)+33930,66:POK
     E(40*I)+33930+SS,1:NEXT
2290 CT=17:FOR I=210 TO 96 STEP-1:FOR J=
     1TO10:NEXT
2300 IF INT(I/8)=I/8 THEN GOSUB 2620
2310 POKE 53249,I:NEXT:FOR I=1TO2:GOSUB
     2620:NEXT:POKE 53269,4
2320 IF S(7)=2 THEN 2350
2330 GOSUB 1690:PRINT"YOU HAVE GRADUATED
     FROM LEVEL 1":PRINT"TO LEVEL 2."
2340 S(7)=2:FOR I=1TO 5000:NEXT:GOTO 125
     0
2350 GOSUB 1690:PRINT "■9SHIFT CRSRUP■CO
     NGRATULATIONS! YOU WENT TO AND"
2360 PRINT"RETURNED, UNMOLESTED, FROM TH
     E":PRINT"SEA OF STATES."
2370 PRINT" ":GOSUB 2630
2380 PRINT "YOUR FINAL SCORE IS";FS
2390 POKE 1,252:POKE 53254,100:POKE 532
     55,60:POKE 53264,8:POKE 53277,15
2400 POKE 53290,7:POKE 53269,12
2410 FOR I=100 TO 1 STEP -1:FOR J=1TO10:
     NEXT
2420 IF PEEK(B1)=252 THEN POKE B1,253:GO
     TO 2440
2430 POKE B1,252
2440 POKE 53254,I:NEXT
2450 POKE 53264,0:POKE 53254,255
2460 FOR I=255 TO 145 STEP-1:FOR J=1TO10
     :NEXT
2470 IF PEEK(B1)=252 THEN POKE B1,253:GO
     TO 2490
2480 POKE B1,252
2490 POKE 53254,I:NEXT
2500 FOR I=1TO200:FOR J=1TO10:NEXT
2510 IF PEEK(B1)=254 THEN POKE B1,255:GO
     TO 2530
2520 POKE B1,254
```

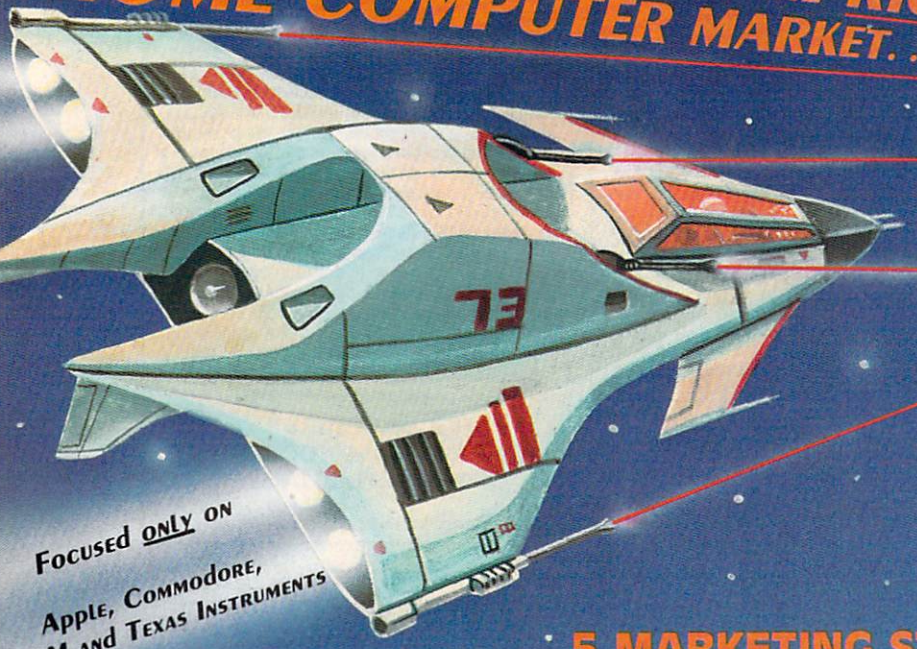
COMMODORE 64

```
2530 NEXT
2540 FOR I=145 TO 0 STEP-1:FOR J=1TO10:N
     EXT
2550 IF PEEK(B1)=252 THEN POKE B1,253:GO
     TO 2570
2560 POKE B1,252
2570 POKE 53254,I:NEXT:POKE 53269,4
2580 GOSUB 1770
2590 PRINT" ":PRINT"WOULD YOU LIKE TO PL
     AY AGAIN?":INPUT"ENTER Y OR N.";P$
2600 IF P$="Y" THEN GOTO 1110
2610 END
2620 POKE (CT*40)+33930,32:CT=CT-1:RETUR
     N
2630 FS=(S(1)+S(2))*(S(3)/(S(4)+1))*S(7)
     :RETURN
2640 POKE 34808,242+INT(AP):POKE 53264,1
     :POKE 53248,100:POKE 53249,210
2650 POKE 53269,3
2660 FOR I=100 TO 1 STEP-1:FOR J=1TO20:N
     EXT
2670 IF INT(I/2)=I/2 THEN GOSUB 2710
2680 POKE 53248,I:NEXT:POKE 53248,255:PO
     KE 53264,0
2690 FOR I=254 TO 150 STEP -1:FOR J=1TO5
     0:NEXT:IF INT(I/2)=I/2 THEN GOSUB 2
     710
2700 POKE 53248,I:NEXT:RETURN
2710 IF PEEK(B2)=241 THEN POKE B2,240:RE
     TURN
2720 POKE B2,241:RETURN
2730 S(1)=S(1)-1:IF S(1)<1 THEN 2840
2740 RETURN
2750 FOR I=1TO5000:NEXT:GOSUB 2810
2760 S(1)=S(1)-LO:IF S(1)<1 THEN S(1)=0:
     GOTO 2840
2770 S(2)=S(2)-LG:IF S(2)<1 THEN S(2)=0:
2780 GOSUB 1690:LO=LO:PRINT"■5SHIFT CRSR
     UP■";LO;"UNITS OF OXYGEN"
2790 PRINT LG;"GOLD PIECES"
2800 FOR I=1 TO 5000:NEXT:RETURN
2810 FOR I=1TO1000:NEXT
2820 LG=INT(RND(0)*6)+F2:LO=INT(RND(0)*6
     )+F2:RETURN
2830 GOSUB 2810:S(1)=S(1)+LO:S(2)=S(2)+L
     G:GOTO 2780
2840 GOSUB 1690:PRINT"■7SHIFT CRSRUP■OH
     NO! YOU HAVE RUN OUT OF OXYGEN!"
2850 PRINT"SO SORRY, BUT YOU LOSE.":PRIN
     T:PRINT:GOSUB 1770
2860 GOSUB 2630:PRINT "YOUR FINAL SCORE
     IS";FS:PRINT:PRINT
2870 POKE 53269,0:GOSUB 1770
2880 PRINT "WOULD YOU LIKE TO PLAY AGAIN
     ?":INPUT "ENTER Y OR N:";G$
2890 IF G$="Y" THEN 1100
2900 END
```

HCM

PLEASE DON'T FORGET TO RETURN THE QUESTIONNAIRE ON THE FRONT BIND-IN CARD.

LASER-TARGETED TO THE RIGHT HOME COMPUTER MARKET...



Focused ONLY on
Apple, Commodore,
IBM and Texas Instruments

5 MARKETING STRATEGIES FOR MAXIMUM REACH, IMPACT, AND RESPONSE

1 Pinpoint The High-Growth Segment Of The Home Market.
The home computer expansion of the 80's will be dominated by 3 brands: Apple, Commodore, and IBM. Advertisers who focus on this group will realize the largest gains.

2 Avoid Too Broad Or Too Narrow An Advertising Medium.
Indiscriminate advertising in both "general interest" and "machine-specific" publications is an inefficient strategy that quickly depletes your advertising budget.

3 Place Your Ad In The Strongest Editorial Environment.
Commercial messages stand out best when surrounded by facts—not fluff! Ads that keep company with the right balance of useful, stimulating articles will be noticed repeatedly.

4 Capture An Overlooked Universe Of Proven Computer Enthusiasts.
The over two million TI Home Computer owners represent an explosive buying potential. These proven enthusiasts will continue to shop for new products and many will migrate to larger, expandable home computer systems.

5 Simplify Your Media Buy With *Home Computer Magazine*.
No other magazine gives you more for your advertising dollar than *Home Computer Magazine*. It gives you instant access to the Apple, Commodore, IBM, and TI home markets without having to pay for readers you don't need or ad duplication you don't want. Our popular programs, colorful graphics, valuable reference articles, and in-depth product reviews provide readers with a magazine that is eagerly read, carefully saved, and constantly referred to. We invite you to examine a copy of *Home Computer Magazine*, and measure it against other print buys.

ONCE YOU COMPARE — THERE'S NO COMPARISON.™

HOME COMPUTER™ magazine

Contact our Ad Department: 1500 Valley River Drive, Suite 250 • Eugene, OR 97401 • Tel. (503) 341-1050

HOME COMPUTER™ product news

Each month we publish items of interest and news of recently or soon-to-be released computer products. Our publication of information from manufacturers of computers, peripherals, software, and accessories is not to be construed as product endorsement. Prices quoted are the manufacturers' suggested retail prices and are subject to change.

Send press releases to:

Product News Editor
Home Computer Magazine
1500 Valley River Dr., Suite 250
Eugene, OR 97401

TECH-SKETCH Light Pens

Tech Sketch, Inc., announces TECH-SKETCH Light Pens for Commodore, Apple, and Atari computers. These devices allow the user to select from a menu or control the cursor without typing commands on the keyboard. The user touches the screen with the light pen, which works directly on the CRT to access programs and manipulate data. The pen plugs into the standard RS232 joystick port.

Two light pen models are available: LP10-S controls the cursor via screen contact, and LP15-S is a high resolution model that controls the cursor up to six inches from the screen. Both pens have a built-in switch and are equally usable by left- and right-handed people. All TECH-SKETCH Light Pens are priced from $39.95 and include a free *Paint-N-Sketch I* program, available in cassette, disk, and cartridge form.

Tech-Sketch, Inc.
26 Just Rd.
Fairfield, NJ 07006
(800) 526-2154

CARTRIDGE PROGRAMMING AT HOME

Navarone Industries has introduced its Cartridge Programming System for the TI-99/4A and IBM PCjr. An add-on device, the system will transfer programs to ROMOX(tm) ECPC(tm) reprogrammable cartridges. Users who want to change a program simply erase it and program it again. The Cartridge Programming System, including hardware, diskette, and manual, retails for $299.95 for the TI-99/4A and for $499.95 for the IBM PCjr. Navarone has also announced a licensing agreement with ROMOX Software Publishing to publish and distribute their entire line of cartridge software for the TI-99/4A and Commodore 64, including *Anteater*, *Topper*, and several more unreleased titles. Six titles already released for the TI-99/4A retail for $29.95 each, and retail prices for the C-64 titles released so far range from $29.95 to $39.95.

Navarone has also announced an updated version of its Widget, the New Cartridge Expander that inserts directly into the game slot on the TI-99/4A and allows three Command Cartridges to be plugged in at one time. With three sockets and a selector switch, the user can choose among cartridges without plugging and unplugging to save wear on the console's cartridge slot. A built-in reset button allows the computer to be reset without turning the power off and on. The New Cartridge Expander retails for $39.95.

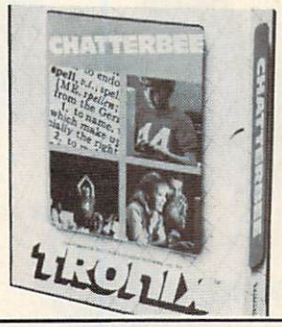
Navarone Industries, Inc.
510 Lawrence Expressway, #800
Sunnyvale, CA 94086
(408) 866-8579

SPEECH SYNTHESIZER AND SPELLING SOFTWARE

Tronix Publishing, Inc., has introduced a C-64 version of its S.A.M., Software Automatic Mouth. The speech synthesizing product is already available for the Atari and Apple computers. Four BASIC demo programs are included with S.A.M.: *SAYIT*, *DEMO*, *SPEECHES*, and *GUESSNUM*. Users can input words using either RECITER, the English text-to-speech converter on S.A.M. that uses 400 rules for combining letters, or phonetic spelling. A dictionary of 1,500 words with phonetic spellings is included. The Atari and C-64 versions of S.A.M. retail for $59.95, and the Apple version retails for $99.95.

Also available for use with S.A.M. is *Chatterbee*, an interactive spelling program for synthesized speech. Available for the C-64 and Atari, *Chatterbee* features a dictionary of 2,500 words and automatic placing of the user at either grades 1-12 or college levels of study. Requiring a disk drive and 40K RAM, *Chatterbee* retails for $39.95.

Tronix Publishing, Inc.
8295 S. La Cienega Blvd.
Inglewood, CA 90301
(213) 215-0529



S.A.T. CONTEST

Krell Software Corp. has announced The Great American S.A.T. (Scholastic Aptitude Test) Contest in two categories: Highest S.A.T. Score and Most Improved S.A.T. Score. First prize in each category will be $1,500 worth of personal computer equipment; second prize will be $750 worth of personal computer equipment. Fifteen runner-up prizes in each category will be $500 worth of Krell software. Each of the 34 winners will be given $1,000 worth of Krell Educational Software to donate to a school of his or her choice. All completed entries must be received by August 31, 1984, and must include copies of official ETS (Educational Testing Service) transcripts and proof of purchase of Krell's College Board S.A.T. Preparation Series. If Krell's S.A.T. Preparation Series is used in a school setting, then proof of purchase by the school is required. Winners will be asked to certify their use of Krell's College Board S.A.T. Preparation Series, and Krell reserves the right to use names and pictures of winners for promotional purposes. The College Board S.A.T. Exam Preparation Series retails for $299.95 and is available for Acorn, Apple, Atari, Commodore 64, PET, IBM, and Radio Shack computers.

Krell Software Corp.
1320 Stony Brook Rd.
Stony Brook, NY 11790
(800) 24-KRELL

HOME COMPUTER product news

HOME PRODUCTIVITY ON THE C-64

Creative Software has announced *The People's Choice*, a series of three integrated home productivity programs for the Commodore 64, with IBM PC and PCjr versions due early second quarter. *Joe's Writer* (word processing), *Fred's Filer*, and *Jack's Calc* (spreadsheet) will be sold separately at $49.95 each.

Creative Software has also introduced *Bumblebee*, an educational software program teaching programming basics in a game-like format, for children ages 6 and up. *Bumblebee* will be available for the Commodore 64 for $34.95.

Three other new titles for the C-64 are also available from Creative Software: *Crisis Mountain*, a game licensed from Synergistic Software; and two educational programs, *In the Chips* (already available for the VIC-20), and *I Am the C-64* in two parts—Introductory and Advanced. *In the Chips* and *Crisis Mountain* are available on cartridge for $34.95, and Part I and Part II of *I Am the C-64* are available on disk for $34.95 each.

Creative Software
230 East Caribbean Drive
Sunnyvale, CA 94089
(408) 745-1655

UNIVERSAL PRINTER

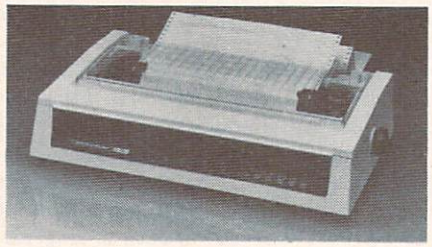
Alphacom has introduced a "universal" printer that is compatible with most popular home computers including the Commodore 64, Apple, TI-99/4A, and Atari, as well as with computers having an RS232 or Centronics interface. The Alphacom 81 has a column width of 80 characters per line. It is priced at $169.95, not including the required interface cables, which range in price from $44.95 to $59.95.

Alphacom Inc.
2323 So. Bascom Ave.
Campbell, CA 95008
(408) 559-8000

DOT MATRIX DEBUTANTS

Smith-Corona announces the debut of their line of dot matrix printers with three new models. The Smith-Corona D-100 prints a column width of 80 characters per line at 100 cps. It offers six different pitches, emphasized or elongated print, proportional spacing, and italics. The D-100 also has bit image graphics and "true" descenders. The character matrix size is 9H x 8V, and the horizontal resolution is 60 and 120 dots per inch. The printer offers a full-line buffer (up to 132 characters) and a 96-character ASCII character set. The D-100 retails for $395. Smith Corona's D-200 has a column width of 80 characters per line and prints at 120 cps. It has a 2K buffer. The character matrix size is 9H x 8V (standard) to 17H x 16V (NLQ), and it offers horizontal resolutions of 60, 72, and 120 dots per inch. The D-200 retails for $595. The Smith Corona D-300 is the top-of-the-line model, which retails for $795. It has a column width of 132 characters per line and prints at a speed of 140 cps.

Smith-Corona
65 Locust Ave
New Canaan, CT 06840



LARRYVISION LOOKS GOOD

LarryVision announces two Extended BASIC games for the TI-99/4A. *QUACKERS* is a duck hunting game that plays like a classic animated cartoon, complete with Elmer Fudd-ish theme music. *PHOTO SAFARI* puts the player on assignment in Africa where he or she must capture 12 jungle animals on film. The photographing of endangered species is a socially responsible outlet for those who wish to avoid violent and destructive videogames.

Both games begin with a 195-color "rainbow" logo and feature color graphics developed for equal contrast and playability on a color or black-and-white monitor. The games were designed for instant accessiblity for very young children and sustained challenge for seasoned arcade players. The price per game is $12.95, disk; and $9.95, cassette.

LarryVision
112 South Third St.
Lebanon, PA 17042

HOME AND SMALL BUSINESS MANAGEMENT SOFTWARE

Timeworks, Inc., has announced four home and small business management programs (including one game) for the IBM PC and PC-compatible personal computers. Versions for the PCjr are expected. *Money Manager*, retailing for $59.95, is a home and small business budget and cash flow system that features bar chart graphics and interfaces with the *Electronic Checkbook* program, which records, sorts, and balances. Timeworks' X-SEARCH(tm) feature permits cross-searches of the *Electronic Checkbook*'s sort categories. The *Electronic Checkbook* retails for $69.95. *Data Manager 2* stores information, including name and address lists and research data, and calculates numerical data. X-SEARCH(tm), X-SORT(tm), and X-CHART(tm) features permit cross-search. *Data Manager 2* will also produce the sum, average, and standard deviation of statistical data along with frequency charts. It retails for $89.95.

Wall Street is a competitive game for one to four players based on realistic financial models. With $1 million each, players speculate in stocks, real estate, precious metals, minerals, and high risk/high return ventures. *Wall Street* retails for $39.95.

Timeworks, Inc.
P.O. Box 321
Deerfield, IL 60015
(312) 291-9200

TOUCH ME FIRST

3M Static Control Systems has introduced a desktop computer protection pad designed to protect equipment from the harmful effects of operator-induced static electricity. The "First Touch" Series 9200 Static Control Computer Pad is a semi-conductive vinyl overlay and a highly-conductive "scrim" layer adhered to a non-skid, noise-reducing foam backing. When grounded, it protects the computer's microelectric circuitry by draining static charge. The operator touches the pad before activating the computer, then periodically thereafter. Static charge is drained to protect against video interference, memory loss, garbled or lost data, mechanical malfunction, and computer downtime. The pad is priced at $69.95.

3M
P.O. Box 33600
St. Paul, MN 55133

TURBO PASCAL COMPILER FOR THE IBM PCjr

Borland International has introduced the only Pascal language compiler to date that will run on the IBM PCjr. The combination compiler/editor occupies 33 Kbytes of memory, compared to 129-300 Kbytes for other Pascal compilers. A single-pass native-code compiler, *Turbo Pascal* produces Pascal for microcomputers based on the Z-80 and 8088/8086 microprocessors. In addition the compiler offers extensions of standard Pascal that include bit-byte manipulation, direct access to CPU memory, dynamic strings, and include files. It will compile more than 2,000 lines of code per minute. Borland will pre-install *Turbo Pascal* on customers' PCjr systems at no charge, eliminating the need for a terminal installation program. *Turbo Pascal* on diskette retails for $49.95.

Borland International
4113 Scotts Valley Dr.
Scotts Valley, CA 95066
(408) 438-8400

TUTORIALS FOR IBM COMPUTERS

Individual Software has announced three interactive tutorial programs for the IBM PCjr, PC, and PC-XT computers. *Typing INSTRUCTOR* features traditional drill and practice, a chase game, and a link to word processing concepts. *Typing INSTRUCTOR* retails for $49.95. *Professor Pixel* includes interactive training in generating sound, graphics, and animation in BASIC, and retails for $59.95. *Professor DOS* is a PC-DOS and MS-DOS training package retailing for $59.95. Each training program includes one or two interactive diskettes and a user guide. System requirements vary slightly, but most programs require a minimum 64K memory, any version of DOS, a monochrome or color display, and at least one double-sided disk drive.

Individual Software
1163-I Chess Drive
Foster City, CA 94404
(415) 341-6116

SNEAKY SNAKE

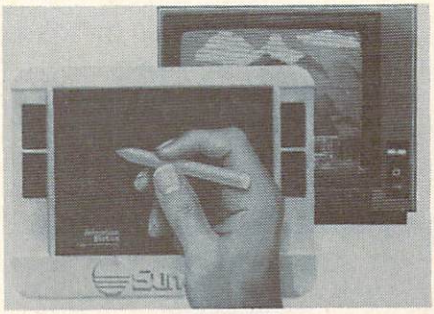
Kean Computing Inc., introduces an action game written in BASIC for the TI-99/4A. *Sneaky Snake* features colorful graphics, quick keyboard response, and multiple screens for a continuous challenge. *Sneaky Snake* sells for $14.95. In addition, Kean Computing offers three TI-99/4A programs in Extended BASIC: *Heist*, an arcade/action game; *Void*, a graphics adventure game; and *Sprite Maker*, a programming and learning tool.

Kean Computing Inc.
Box 571
Kingston, NJ 08528
(201) 329-2438

GRAPHICS TABLET CURSOR CONTROLLER

Suncom, Inc., has introduced *Animation Station*(tm), a touch-sensitive graphics tablet computer cursor controller. *Animation Station* features side mounted dual left or right hand function buttons. Its touch sensor surface is built to the rectangular proportions of a home television set to make the on-screen graphics fit the borders of the TV screen. *Animation Station* will be packed with its own graphics utility program and a line of educational, word processing, and graphics applications is planned. Versions of *Animation Station* for the Commodore 64 and Atari retail for $79.95; the Apple version retails for $114.95; a version for the IBM PCjr is planned.

Suncom, Inc.
650E Anthony Trail
Northbrook, IL 60062
(312) 291-9780



PERIPHERALS FOR THE TI-99/4A

Mikel Laboratories, Inc., has announced a new RS232 Interface system compatible with the TI-99/4A. The system is a free standing unit allowing the use of a printer and modem without the peripheral expansion box. The system retails for $149.95. Mikel also offers cassette interface systems for $49.95, TI cassette cables for $11.95, and printers and monitors.

Mikel Laboratories, Inc.
3341 W. El Segundo Blvd.
Hawthorne, Ca 90250
(213) 679-2542

A RUN FOR YOUR MONEY

Scarborough Systems, Inc., announces a unique arcade-action business game designed to encourage the development of sound entrepreneurial skills while providing family fun. *Run For The Money* takes place on the hypothetical planet Simian, where the user's spaceship has suffered a forced landing. The players must develop a business and raise money for the needed repairs. Scarborough claims that the action is based on accurate economic models and requires the players to quickly make important decisions about raw materials, marketing, prices, and profits. Increasingly challenging "what if's" occur at regular intervals. *Run For The Money* is available for the IBM PC for $49.95, and versions for the Atari, Apple II+ and Apple IIe are in the works.

Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591
(914) 332-4545

MULTIPLAN FOR THE COMMODORE

Through a licensing agreement with Microsoft Corp., Human Engineered Software will distribute Microsoft's *Multiplan* (tm) spreadsheet program under the HesWare (tm) label for the Commodore 64. The program, which Microsoft has recreated for the C-64, contains built-in arithmetic, financial, and trigonometric functions, screen windows, variable column widths, alphabetic and numeric sorting, the ability to link worksheets, and flexible formatting for screen displays and reports. Human Engineered Software's *Multiplan* package comes with a disk, manual, and keyboard overlay, and retails for $99.95. Files in *Multiplan* for the C-64 can be transferred to *Omniwriter*, Human Engineered Software's word processor for the C-64 that includes *Omnispeller*, a 30,000 word spelling checker. *Omniwriter* retails for $69.95.

Human Engineered Software
Brisbane, CA 94005
(415) 468-4111

NEW INTERFACE CABLE ALLOWS OKIDATA/TI HOOK-UP

Innovative Electronics and Computing has announced an interface that permits an Okidata printer to be connected to the PIO (parallel) interface port on the TI RS232 card. The interface circuitry fits into one end of a standard parallel cable. The interface changes the signal from the PIO port to make it acceptable to the Okidata parallel interface port. The interface is priced at $29.95.

Innovative Electronics and Computing
4150 Fox St., A-5
Denver, CO 80216
(303) 458-5600.

CAGEY CATERPILLARS FOR HOME GAMESMEN

Castle Software, Inc., has announced the release of *Killerpiller*, available on cassette and diskette for the Commodore 64 computer. The game features the player as a pest eradicator against caterpillars that threaten an orchard. The caterpillars grow as they eat trees, multiply as they split when hit, and are protected by attacking moths. *Killerpiller* has seven screen levels and two skill settings. Joystick is required. The game is available at $9.99 for cassette and $12.75 for diskette.

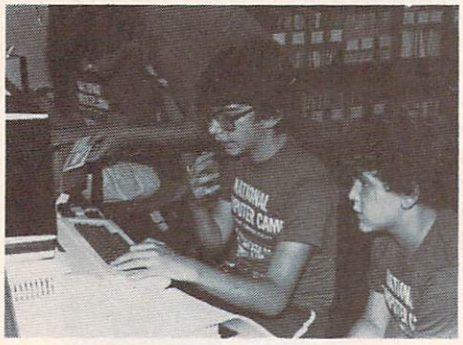
Castle Software, Inc.
P.O. Box 350
New Castle, DE 19720
(302) 429-8565.

DUAL DISK DRIVE FOR APPLE

Apple Computer announces a new floppy disk drive unit which contains two half-high, 140 kilobyte drives side-by-side in a single case. Duodisk offers full compatibility with Apple II software and is operationally identical to the Disk II floppy disk drive. Technical improvements include a new disk eject mechanism and an advanced head positioning mechanism, which gives a more precise reading of half-tracks.

The Duodisk comes with a controller card that connects it to any Apple II, Apple II+ or Apple IIe. It fits between the computer and monitor in a case designed to create an integrated system look. The suggested retail price is $795.

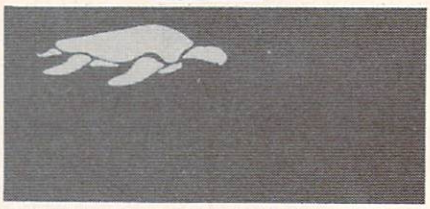
Apple Computer, Inc.
10260 Bandley Dr.
Cupertino, CA 95014
(408) 996-1010



COMPUTER CAMPS

National Computer Camps has announced 5 locations for 1984, its 7th season. Campers ages 9 to 18 may sign up for one or more weeks during June, July, and August. The coeducational camps with a ratio of one teacher and assistant per twelve campers will feature 5 hours "hands-on" instruction per day on TRS-80 and Apple computers, two campers per computer, with up to 5 additional hours of optional computer time each day. Campers will be grouped by background and age into 11 different levels of study. The program of study will include BASIC, machine language for Apple and Radio Shack computers, graphics and animation, word processing, Pascal for beginners, computer game design, the social implications of computers, and computer careers and literacy. Outdoor recreation—tennis, swimming, aerobic dancing, weight-lifting, soccer, basketball, softball, and volleyball—will be featured as well. Camps will be located at prep schools, colleges, and universities in Simsbury, CT; Atlanta, GA; St. Louis, MO; Cleveland, OH; and McMinnville, OR.

National Computer Camps
Box 585
Orange, CT 06477
(203) 795-9667



ENHANCED LOGO FOR THE APPLE

Terrapin, Inc., has announced Terrapin LOGO, an enhanced MIT LOGO for the Apple II or II+ (with 64K memory) and the Apple IIe. With assembly language interface capability and the ability to save pictures on disk, Terrapin LOGO features more than 120 primitives, including program-tracing capability, music procedures, and *Instant*, a program to draw turtle graphics with single-letter commands. The LOGO package, including a LOGO language disk, a utilities disk, 229-page LOGO Tutorial, and 86-page technical manual, retails for $149.95.

Terrapin, Inc.
380 Green St.
Cambridge, MA 02139
(617) 492-8816.

PORTABLE PRACTICE KEYBOARD

Computer Practice Keyboard Co. has introduced a portable keyboard for practice away from the computer. Each plastic-laminated keyboard, printed on 8 1/2" x 11" stock to fit standard size binders, explains special key functions for the Apple, Atari, TRS-80, TI-99/4A, IBM, Commodore, Wang, Xerox, Osborne, Timex, and Advantage computers. A brief dictionary of computer words is printed on the back of the keyboard, which retails for $3.95.

Computer Practice Keyboard Co.
616 9th St.
Union City, NJ 07087
(201) 863-0999

NEW ARCADE POWER STRUGGLE

Gadget Software has announced *Power Failure*, a multi-screened arcade game for the TI-99/4A. The game features ten different playing screens, keyboard or joystick control, and one- or two-player capability. *Power Failure* is available in both BASIC and Extended BASIC for $16.99 U.S./$19.99 Can.

NU-WEST Marketing
10013 Densmore Ave.
North Seattle, WA 98133

THREE FOR TI FROM TOMPUTER

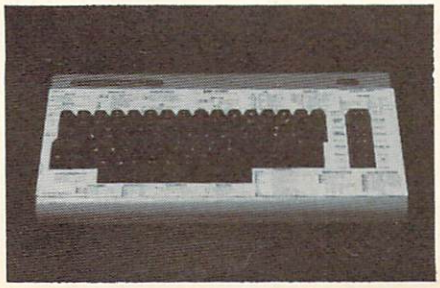
Tomputer Software announces three new programs for the TI-99/4A. *Extended Addition and Subtraction* is designed to improve the math skills of grade school children. It is priced at $14.99 and requires Extended BASIC and a cassette recorder. *The Castle of Death* is a graphics adventure game featuring a 100-room castle complete with bats, spiders, mice, bombs, skulls, mazes, and more. In *The Quest for Hitler's Diary* players are confronted with a subterranean secret room, flying swastikas, and Hitler's ghost. Both of the games require Extended BASIC, a joystick, and a cassette recorder, and are priced at $17.99 each.

Tomputer Software
1550 Montgomery Dr.
Deerfield, IL 60015
(312) 945-9677

CHEATSHEET OVERLAYS

Cheatsheet Products(tm) announces help for Commodore 64 and VIC-20 users in the form of Leroy's Cheatsheets. These durable plastic-coated templates lie on the keyboard to identify specialized commands and functions. Overlays are available for many popular Commodore and third-party programs, including *Basic*, *Quick Brown Fox*, *Easy Script*, and *Graphic Printer*. Each Cheatsheet comes with starting instructions, marked function keys, and additional tables, charts, or drawings particular to the program for which it was designed. All Cheatsheets are $3.95 apiece.

Cheatsheet Products
P.O. Box 8299
Pittsburgh, PA 15218
(412) 456-7420



TWO NEW APPLE MODEMS

Apple Computer has announced two modems, a 300-baud model and a 1200-baud model, each fully compatible with all Apple II, Apple III, Macintosh, and Lisa personal computers. The modems permit the transmission and reception of data files, electronic mail, and computer programs to and from other personal computers, minicomputers, and mainframes. Home users can perform home banking transactions and tie into information services such as CompuServe, Dow Jones News/Retrieval, and The Source.The Modem 1200 can operate at both 300-baud and 1200-baud rates. The Modem 300 works at both 300-baud and 110-baud rates.

The modems use an RS-232C serial interface with a 9-pin connector. The Apple II requires a serial interface card to connect with either modem. The modems come with an accessory kit containing installation manual, user's manual, and computer-specific connector cable. An optional accessory kit for Apple II users includes a serial interface card. Modem accessory kits for Apple II and Apple III computers include a terminal program that allows users to dial into information services such as CompuServe and The Source. The Apple Modem 300 retails for $225.00; with accessory kit including the serial interface card, for $299.00. The Modem 1200 retails for $495.00; with the optional accessory kit containing the card, for $570.00.

Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, CA 95014
(408) 996-1010

DISK DRIVE DIAGNOSIS

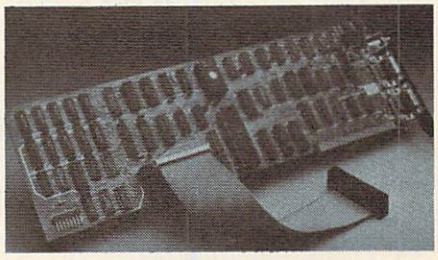
Data Encore has announced the Datalife Disk Drive Analyzer(tm) on 5 1/4" minidisk for IBM PC and some compatible systems. Also available for Apple II, II+, IIe, and III computer systems, the analyzer will run four tests (for head alignment, disk clamping, write/read accuracy, and disk speed) and display an evaluation on-screen to indicate whether tested areas will require adjustment and/or repair. The IBM version and a new Apple version of the Disk Drive Analyzer retail for $39.95 each.

Data Encore
585 N. Mary Ave
Sunnyvale, CA 94086
(408) 720-7400

IBM-APPLE LINK

Quadram Corporation has introduced Quadlink, an expansion board that allows IBM PC users to run Apple software. When installed in slot #4, the board makes the IBM PC or XT functionally equivalent to an Apple 64K computer. There is no need to convert or reformat any disks, nor to plug or unplug monitor cables. The user can switch back and forth between Apple mode and IBM mode with just a couple of keypresses. The Quadlink board works with most installed I/O devices such as parallel and serial ports for connecting printers and other accessories. It allows the use of all IBM enhancements such as printer buffers, etc., while running Apple software. The Quadlink system comes standard with 64K of memory, a game port that is both IBM and Apple compatible, and a display adapter that offers five display modes, including high-resolution color graphics. The price is $680.

Quadram Corporation
4355 International Blvd.
Norcross, GA 30093
(404) 923-6666

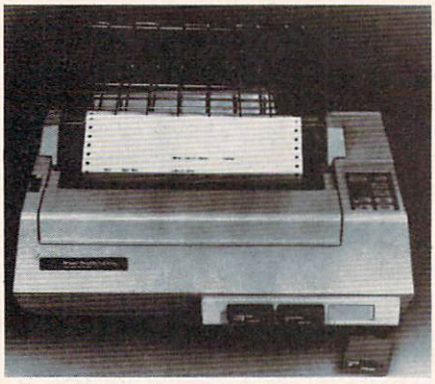


COMPUTER VACUUM CLEANER

The Pine Cone has announced the MINI-VAC, a 6-ounce, 5" x 1 1/4" x 1 1/4" vacuum cleaner or blower designed to remove minute particles of dust and debris from computers, cameras, and other dust-sensitive gear. MINI-VAC comes with lens and all-purpose brushes, two interchangeable directional wands, two fine-bristle brushes, and a cloth vacuum bag. The motor is AC or DC powered with optional adapter (9-volt battery not included). The MINI-VAC retails for $29.95.

The Pine Cone
Blake Building, Dept. ER-284
P.O. Box 1378
Gilroy, CA 95021

. . .from p. 143



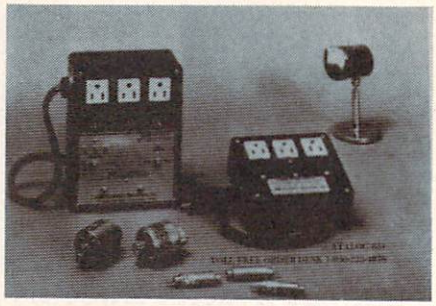
DUAL-MODE TI MICROPRINTER

Texas Instruments has announced a "two-in-one" microprinter, the OMNI 800(tm) Model 855. The dual-mode dot matrix printer provides both 35 cps letter-quality printing for word processing and a high-speed 150 cps draft mode for data processing printing. The printer is designed to operate with the TI Professional Computer and all other major personal and professional computers and with existing third-party word processing and data processing software.

Font modules (up to three at a time) plug into the front of the printer to allow the user to select from a number of fonts and character sets without stopping. Each font module provides both draft and letter-quality character sets. The letter-quality mode uses a 32 × 18 dot matrix format, and the draft mode is in a 9 × 9 matrix format.

The Model 855 will print an original and two copies on paper three to eleven inches wide. TI's patented metering tractor on the input side of the friction feed roller is designed to minimize form and paper waste on the output side by allowing about an inch tear-off from the last printed line on a document. A future option for the Model 855 will be a bin feeder to ease continuous printing on letterhead stationery. The friction-feed model retails for $935, and the tractor-feed model is priced at $995.

Texas Instruments
Data Systems Group
P.O. Box 402430, H-669
Dallas, TX 75240
(800) 527-3500.



CATALOG OF PROTECTION DEVICES

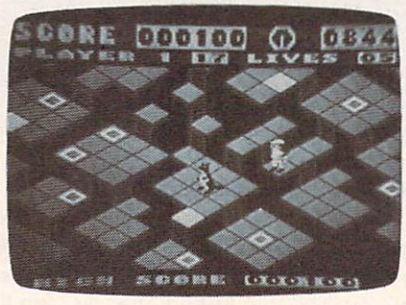
Electronic Specialists, Inc. has just announced their new *Interference Control & Electronic Products Catalog*, a 40-page catalog of products designed to eliminate software performance problems. Protective devices include equipment isolators, AC power line filter/suppressors, AC line voltage regulators, and modem protectors. Descriptive sections outline particular software problems and suggested solutions. Typical applications are highlighted.

Electronic Specialists, Inc.
171 South Main Street
Natick, MA 01760
(800) 225-4876

MULTI-LEVEL & 3D GAMES FROM FIRST STAR

First Star Software, Inc., has announced two games for the Commodore 64 and Atari 400, 800, 1200 XL, and 5200 computers. *Bristles*(tm), authored by Fernando Herrera (Commodore version by Adam Bellin), is a four-player game featuring the painting of 8 different dwellings and including 48 levels. *Flip and Flop*(tm) is a three-dimensional one- or two-player game. The player can be a kangaroo on a multi-screened, scrolling, stepped-platform maze with ladders, or a monkey hanging from the underside of the maze, swinging from square to square. At higher levels, a flying net chases the monkey. *Flip and Flop* features 6 animations, thirty-five levels of play, and a musical soundtrack for the titles portion of the program. Available on cassette, diskette, and cartridge, each game retails at $29.95 for cassette and diskette, $39.95 for the cartridge.

First Star Software, Inc.
22 East 41st St.
New York, NY 10017
(212) 532-4666.



WWII PINBALL COMES HOME

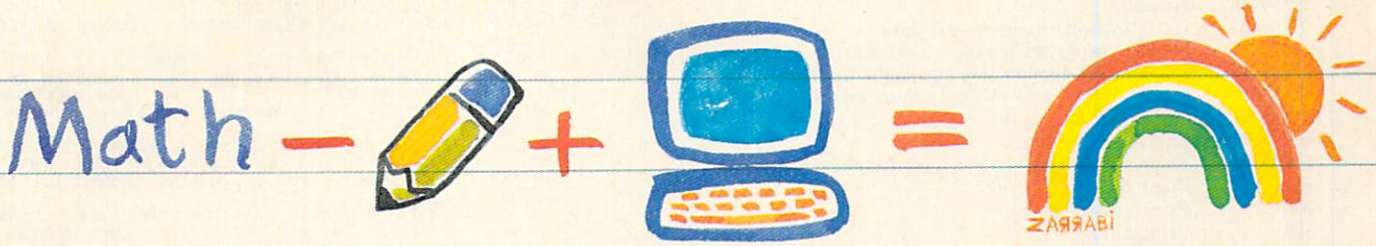
SubLOGIC has announced *Night Mission Pinball* on cassette or diskette for the Commodore 64 and the Apple II. The one- to four-player game's theme is based on a WWII B-17 Flying Fortress night bombing run and features five bumpers, seven standup targets, nine rollovers, and two spinners. Options include ten modes of play—from Competition to Cosmic—and forty user-adjustable parameters. Joysticks are recommended. *Night Mission Pinball* is available for the C-64 and for the Apple II at $34.95.

SubLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
(217)359-8482
Telex 206995.

PROFESSIONAL LEVEL HOME PRODUCTIVITY

Western Properties Investment Company has announced for the TI-99/4A, a professional level accounting spreadsheet, the *Income and Expense Spreadsheet IVp*, at $43.95. The spreadsheet is composed of two income and 50 expense categories. Also offered is *File-Book III* for up to 100 records, 6 items per record, at $39.95. Western Properties is also announcing the availability of *Printer Book*, a word processing program for letter writing or up to two pages of text. The *Printer Book* is priced at $35.95. *Income and Expense Spreadsheet IVp* and *File Book III* require only the Extended BASIC cartridge and cassette recorder and cable to run. Printer is optional. *Printer Book* requires the Extended BASIC cartridge, cassette recorder and cable, and printer. No memory expansion is required for any of the programs.

Western Properties Investment Co.
Software Division
P.O. Box 9602
Marina Del Rey, CA 90295.



Elementary Addition and Subtraction

By Mark Dewese
and the HCM staff

The old saw, "a picture's worth a thousand words," is especially true when you're dealing with very young children. *Basic Addition and Subtraction* came out of that realization. I wanted to write a program that would teach my four-year-old son to add and subtract while he played with the computer. I asked my wife, who is a public school teacher, what she thought would best teach the concepts of addition and subtraction to a preschooler who already knew the numbers zero through ten. As she began to explain methods that have worked in the school system, my thoughts raced to programming structure and design.

I've designed *Basic Addition and Subtraction* to be a user-friendly program, bearing in mind that what may seem user-friendly to an adult may not necessarily be so for a four-year-old. But although little adult instruction is necessary, parents will enjoy helping their young children learn basic addition and subtraction with this easy-to-use program. The problems generated by the computer are basic and will have answers that range from zero to nine.

Working With the Menu

Once the program begins to execute, the computer will play a musical scale as the monitor displays the program title and instructs the user to press any key to begin. Having pressed a key, the child will see the first menu, instructing him to press the number 1 for addition or the number 2 for subtraction. The monitor will then display a second menu to allow the child to select one of three skill levels: EASY, KIND OF HARD, and FOR THE CHAMP.

Easy

The EASY level is for children who are learning how to count from one to ten and getting their first exposure to addition and subtraction. This level is also helpful for children learning to associate numbers with actual quantities. Here, two identical problems are displayed on the monitor. One is in the form of a regular addition or subtraction problem, with each number displayed in large, colorful block characters. Above the number problem is the same problem, displayed in color graphics instead of numbers. On this level the answer is given in the graphics problem but not in the number problem.

Kind of Hard

As in level one, two identical problems are displayed, but here the graphics answer is not given. When the child presses a number key, that corresponding number is placed on the screen along with the corresponding number of graphics characters associated with the pressed number key.

For the Champ

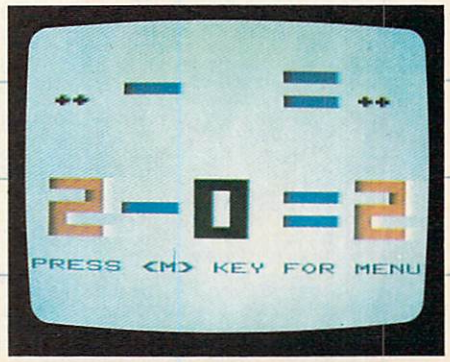
This screen does not display a graphics problem and so is the most difficult of the three levels. Once the child masters this level, he will have successfully fulfilled the educational intentions of the program.

Using the Program

The parent helps the child select the mode he wants by pressing the corresponding keys when the menu screens are displayed. To enter an answer, the child may press any of the number keys as many times as he likes. When he is satisfied that the answer is right, he should press the [RETURN] key. Then the computer will evaluate the answer and provide an appropriate response. If the answer was correct, an upbeat musical progression will be played and the number answer will flash different colors. After a slight delay, the screen will clear and another problem will be displayed. If the answer was incorrect, a slow, low musical progression will be played, the answer will be erased, and the child may try again. Either way the child will be rewarded and will be likely to try again. To change modes, he simply presses the M key, and the program will go back to the first menu.

Continued on next page

". . . although little adult instruction is necessary, parents will enjoy helping their young children learn basic addition and subtraction with this easy-to-use program."



Add & Subtract . . . from p. 113

BASIC ADDITION AND SUBTRACTION (VIC-20)
Explanation of the Program

Line Nos.	
100-160	Header.
170	Initialization. Read sound data.
190-310	Display program prompts and messages.
320-450	Derive and display left and right numbers.
460-620	Input answer and evaluate and reward.
630-650	Get keyboard input.
660-680	Sound subroutine.
690-870	Print subroutines to position numbers on screen.
880	Print plus sign.
890	Print equal sign.
900	Print minus sign.
910	ON GOTO statement for numbers.
920-960	Position cursor subroutines.
970-1050	POKE graphics character subroutines.
1060-1070	Clear portions of screen subroutines.
1080-1090	Routine to change color of answer number.

VIC-20

```
100 REM  ********************
110 REM  *  ADD SUBTRACT PROGRAM  *
120 REM  ********************
130 REM  BY MARK DEWESE AND THE HCM STA
    FF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  V20 BASIC
170 POKE36879,25:S=30720:D=100:V=36878:
    FORI=0TO6:READS2(I):NEXT
180 DATA 135,163,175,135,175,135,195,19
    1,183,175,163,159,147,135
190 PRINT"■SHIFT CLR■CTRL BLU■6CRSRRI
    GHT■B A S I C":PRINT"■3CRSRDOWN■6C
    RSRRIGHT■ ADDITION"
200 PRINT"■CRSRDOWN■"TAB(8)"A N D":PRIN
    T TAB(5)"■CRSRDOWN■SUBTRACTION"
210 PRINT"■4CRSRDOWN■5CRSRRIGHT■2CRSR
    DOWN■PRESS ANY KEY":PRINT"■5CRSRRIG
    HT■TO CONTINUE"
220 FORI=1TO7:READS1(I):F1=S1(I):GOSUB6
    60:NEXT
230 IFPEEK(197)=64THEN230
240 PRINT"■SHIFT CLR■4CRSRDOWN■ENTER":
    PRINT"1 FOR ADDITION":PRINT"2 FOR S
    UBTRACTION"
250 GOSUB 630:IF A<>49 AND A<>50 THEN
    250
260 OV=A-48
270 PRINT"■SHIFT CLR■CRSRDOWN■ENTER:":
    PRINT"■CRSRDOWN■1 FOR EASY."
280 PRINT"■CRSRDOWN■2 FOR KIND OF HARD.
    "
290 PRINT"■CRSRDOWN■3 FOR THE CHAMP."
300 GOSUB630:IFA<49ORA>51THEN300
310 GV=A-48
320 PRINT"■SHIFT CLR■":N=INT(RND(0)*9)+
    1:IFOV=1THEN360
330 LN=INT(RND(0)*N)+1:IF LN>=N/2 THEN
    RN=N-LN:GOTO 350
340 RN=LN:LN=N-RN
350 N=LN-RN:GOTO 370
360 LN=INT(RND(0)*N)+1:RN=N-LN
370 PN=LN:TB=1:GOSUB 920:GOSUB 910:IFGV
    >2THEN390
380 NC=LN:AD=7815:GOSUB 970
390 IF OV=1 THEN GOSUB 930:GOSUB 880:
    GOTO 410
400 GOSUB 950:GOSUB 900:IFGV<3THENGOSUB
    960:GOSUB 900:GOTO420
410 IF GV<3 THEN GOSUB 940:GOSUB 880
420 PN=RN:TB=9:GOSUB 920:GOSUB 910:IFGV
    >2THEN440
430 AD=7823:NC=RN:GOSUB 1030
440 GOSUB 930:GOSUB 890:IFGV<3THENGOSUB
    940:GOSUB 890
450 IFGV=1THENSN=N:GOTO590
460 PRINT"■HOME■19CRSRDOWN■PRESS <M> K
    EY FOR MENU":GOSUB630:IFA<>13THEN55
    0
470 IF SN<>N THEN 530
480 FOR J=1TO7:F1=S1(J):D=200
490 IFJ/2=INT(J/2)THENAD=7940:GOSUB1080
    :GOTO510
500 TB=18:GOSUB 920:PN=SN:GOSUB 910
510 GOSUB 660:NEXT
520 FORI=1TO 5000:NEXT:GOTO 320
530 FORJ=1TO6:D=500:F1=S2(J):GOSUB670:N
    EXT
540 GOSUB1070:GOTO460
550 IFA<48ORA>57THEN610
560 IFGV<>2THENGOSUB1070:GOTO580
570 GOSUB1060
580 SN=A-48:TB=18:PN=SN:GOSUB 920:GOSUB
    910:IFGV<>2THEN600
590 AD=7832:NC=SN:GOSUB1030
600 GOTO460
610 IF A=77 THEN 240
620 GOTO 460
```

VIC-20

```
630 GET A$:IF A$<>"" THEN 630
640 GET A$:IF A$="" THEN 640
650 A=ASC(A$):RETURN
660 POKEV-4,0:POKEV-2,F1:GOTO680
670 POKEV-2,0:POKEV-4,F1
680 POKEV,15:FORZX=1TOD:NEXT:POKEV,0:RE
    TURN
690 PRINT TAB(TB)"■CTRL BLK■CTRL RVSON
    ■ ":PRINT TAB(TB)"■CTRL RVSON■■C
    TRL RVSOFF■■CTRL RVSON■ ":PRINT TA
    B(TB)"■CTRL RVSON■ ■CTRL RVSOFF■■C
    TRL RVSON■ "
700 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL RVSON■ ■CTRL RVSOFF■CTRL BLU
    ■":RETURN
710 FOR I=1TO5:PRINT TAB(TB)"■CTRL RED■
    ■CTRL RVSON■ ■CTRL RVSOFF■":NEXT:
    PRINT"■CTRL BLU■":RETURN
720 PRINT TAB(TB)"■CTRL PUR■■CTRL RVSON
    ■ ":PRINT TAB(TB)" ■CTRL RVSON■
    ":PRINT TAB(TB)"■CTRL RVSON■ "
730 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■ ":PRINT TAB(TB)"■CTRL RVSON■
    ■CTRL BLU■":RETURN
740 PRINT TAB(TB)"■CTRL CYN■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINTTAB(TB)" ■
    CTRL RVSON■ ":PRINT TAB(TB)"■CTRL R
    VSON■ "
750 PRINT TAB(TB)" ■CTRL RVSON■ ":PRIN
    T TAB(TB)"■CTRL RVSON■ ■CTRL BLU■
    ":RETURN
760 PRINT TAB(TB)"■CTRL PUR■■CTRL RVSON
    ■ ■CTRL RVSOFF■ ":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL RVSO
    N■ ■CTRL BLU■":RETURN
770 PRINT TAB(TB)" ■CTRL RVSON■ ■CTRL
    RVSOFF■":PRINT TAB(TB)" ■CTRL RVSO
    N■ ■CTRL BLU■":RETURN
780 PRINT TAB(TB)"■CTRL GRN■■CTRL RVSON
    ■ ":PRINT TAB(TB)"■CTRL RVSON■ ■C
    TRL RVSOFF■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
790 PRINT TAB(TB)" ■CTRL RVSON■ ":PRIN
    T TAB(TB)"■CTRL RVSON■ ■CTRL RVSO
    FF■CTRL BLU■":RETURN
800 PRINT TAB(TB)"■CTRL BLU■■CTRL RVSON
    ■ ":PRINT TAB(TB)"■CTRL RVSON■ ■C
    TRL RVSOFF■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
810 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL RVSON■ ":RETURN
820 PRINT TAB(TB)"■CTRL RED■■CTRL RVSON
    ■ ■CTRL RVSOFF■":FOR I=1TO4:PRINT
    TAB(TB)" ■CTRL RVSON■ ":NEXT:PRIN
    T"■CTRL BLU■"
830 RETURN
840 PRINT TAB(TB)"■CTRL RED■■CTRL RVSON
    ■ ":PRINT TAB(TB)"■CTRL RVSON■ ■C
    TRL RVSOFF■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
850 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL RVSON■ ■CTRL BLU■":RETURN
860 PRINT TAB(TB)"■CTRL GRN■■CTRL RVSON
    ■ ":PRINT TAB(TB)"■CTRL RVSON■ ■C
    TRL RVSOFF■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
870 PRINT TAB(TB)" ■CTRL RVSON■ ■CTRL
    RVSOFF■":PRINT TAB(TB)" ■CTRL RVSO
    N■ ■CTRL RVSOFF■CTRL BLU■":RETURN
880 PRINTTAB(6)"■CTRL RVSON■ ":PRINTTAB
    (5)"■CTRL RVSON■ ":PRINTTAB(6)"■C
    TRL RVSON■ ":RETURN
890 PRINTTAB(14)"■CTRL RVSON■ ":PRINT
    TAB(14)"■CRSRDOWN■CTRL RVSON■ ":
    RETURN
900 PRINTTAB(5)"■CTRL RVSON■ ":RETURN
910 ON PN+1 GOTO 690,710,720,740,760
    0,780,800,820,840,860
920 PRINT"■HOME■11CRSRDOWN■":RETURN
930 PRINT"■HOME■12CRSRDOWN■":RETURN
940 PRINT"■HOME■2CRSRDOWN■":RETURN
950 PRINT"■HOME■13CRSRDOWN■":RETURN
960 PRINT"■HOME■3CRSRDOWN■":RETURN
970 CL=((INT(RND(0)*4)+1)*2:CH=INT(RND(0
    )*4)+1
980 ON CH GOTO 990,1000,1010,1020
990 CH=83:GOTO 1030
1000 CH=88:GOTO 1030
1010 CH=90:GOTO 1030
1020 CH=65
1030 IF NC=0 THEN RETURN
1040 FORI=0TONC-1:IFI/3=INT(I/3)THEN AD=
     AD-25
1050 POKEAD+1,CH:POKES+AD+1,CL:AD=AD+1:N
     EXT:RETURN
1060 PRINT"■HOME■":FORI=1TO18:PRINTTAB(1
     8)" ":NEXT:RETURN
1070 PRINT"■HOME■8CRSRDOWN■":FORI=1TO9:
     PRINTTAB(18)" ":NEXT:RETURN
1080 FORX=1TO3:POKEAD+S+X-1,J:NEXT:IFAD<
     >8050THENAD=AD+22:GOTO1080
1090 RETURN
```

HCM

Adventures in Math

a review

by Sharyn Lyon
HCM Staff

Practice, practice, practice...who says it has to be dull? Certainly not IBM! They have put the fun back into the fundamentals of math with their *Adventures in Math* software for the PC and the PCjr. Parents, educators, and elementary school students will be hard pressed to find a more entertaining and educationally sound way of practicing math facts and processes. In this game, knowing math skills literally opens the door to one adventure after another.

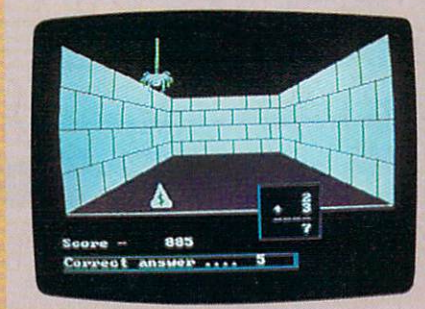
Name:	**Adventures In Math**
Program Type:	**Educational Game**
Machine:	**IBM PC and PCjr**
Distributor:	**IBM Corp.**
	P.O. Box 1328-W
	Boca Raton, FL 33432
Price:	**$35.00, diskette**

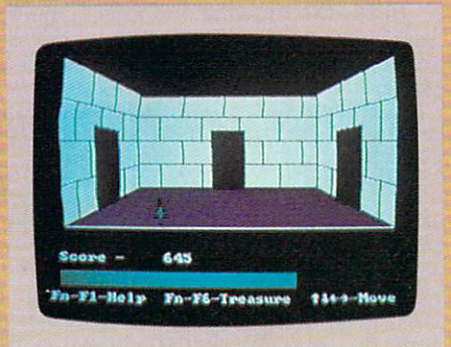
System Requirements:
IBM PC with 64K or PCjr with 128K, Disk Drive, Color Monitor with IBM Color-Graphics Adapter (for PC) or Color TV (for PCjr).

	poor	fair	good	excellent
Documentation				
Independence				
Graphics				
Rewards				
Concept Presentation				

Learning Is An Adventure

Adventures in Math is an educational game for one or two players. It follows the format of a typical adventure game—you try to find your way through a castle while amassing riches and treasure but without getting zapped. There is an educational twist in *Adventures in Math*—you not only have to figure out where the exit is, but you must unlock the doors and gather treasure by correctly completing the math problems that appear on the castle floor. If you do not try to open the doors, you will find yourself hopelessly wandering about in circles. If you ignore the treasure, you will be scorched by a dragon. A flashy reward is in store for earnest adventurers who do the math, gather the treasures, and find their way out of the castle. Superstars will even get to see their names and scores on the Ten Highest Scores list.



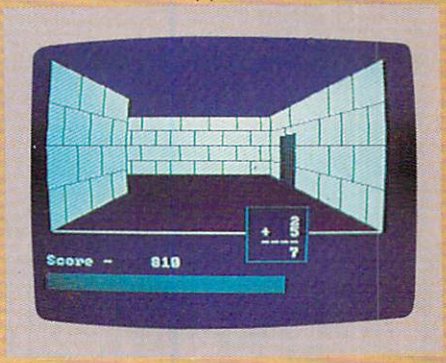


To play *Adventures in Math* you will need a good understanding of at least one of the four basic math operations. Although no age-range is specified in the documentation, children who have not been exposed to re-grouping (carrying in addition, borrowing in subtraction) will have difficulty with this game. The software could be used to good advantage, however, in a multi-age tutoring situation in which a younger child works with an older one. Using either the one- or two-player option, students can cooperate to solve the problems and get out of the castle.

Double Your Fun

Adventures in Math subtly achieves a dual purpose, providing both practice and entertainment. As you move through the castle answering problems correctly, the level of difficulty automatically increases. But the challenge doesn't stop there. Simply answering a given number of problems correctly will not get you out of the castle. You really do have to *find* your way out of this mathematical maze! Clues help: the more problems/treasures you encounter, for example, the more likely you are to be on the path out of the castle. We felt it was clever of the software designers to associate math problems with treasure and riches. You will undoubtedly find yourself actually looking forward to the next opportunity to do a math problem.

Parents will appreciate this game

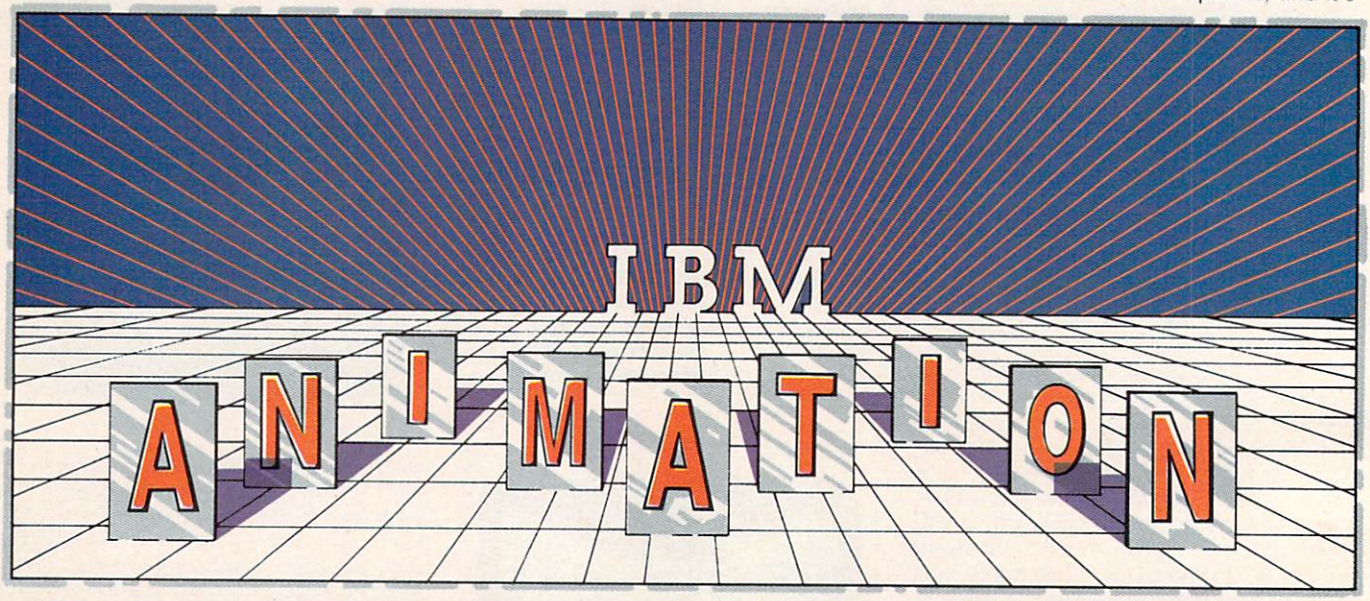


because it helps children refine skills in other areas. While they are practicing what they know about math to collect points, they are also applying their deductive powers, mapping skills, and understanding of spatial relationships to escape from the castle. With three castle sizes and five skill levels to intrigue them, youngsters will be fascinated by *Adventures in Math* for a long time; it grows with them. At first, children might be tempted to get the answers wrong just to see how the creatures zap them. Chances are, however, that once they get into the game they will lose interest in seeing "what-ifs" because they'll be so involved in seeing how many points they can get and how quickly they can find the castle exit.

Graphics Galore

Adventures in Math is a graphics adventure, but the chairs, rugs, and jars that appear in the various rooms do not operate as they usually do in adventure games. They are not collectibles or things to look behind. They are "props" that help you orient yourself in the castle. If, as in any good adventure game, you make a map noting where these set decorations are, you can save yourself moves. The graphics are appropriately encouraging, and help keep you on the right track. The creatures that appear when you ignore a treasure or answer a problem incorrectly are not nearly so spectacular as the

Continued on p. 146



Controlling the Palette on the PCjr

by W. K. Balthrop
HCM Staff

Graphically speaking, the IBM PCjr can draw circles around any other home computer, including its brother, the IBM PC. Four new graphics modes, available only on the junior, give you increased screen resolution and expanded color choice. Our program, *Animation*, is written in Mode 5, which allows you to use the machine's 16 colors with a screen resolution of 320x200. In Mode 5 you can actually change colors within the program by using the PALETTE command. Modes 5 and 6 both require 32K of memory for the screen graphics alone. You will need to have the 64K Memory and Display/Expansion Option card installed and use the CLEAR command in line 180 to reserve 32K of video memory.

Pallette Table Available Colors

(holds color choice)	color #	color
0	0	Black
1	1	Blue
2	2	Green
3	3	Cyan
4	4	Red
5	5	Magenta
6	6	Brown
7	7	White
8	8	Grey
9	9	Light Blue
10	10	Light Green
11	11	Light Cyan
12	12	Light Red
13	13	Light Magenta
14	14	Yellow
15	15	High Intensity White

You're probably wondering what the PALETTE command has to do with creating animation. PALETTE allows you to alter the assigned color of any of the color numbers used in the program. For example, you might want to transform black (color #0) into high intensity white (color #15). To do this you would use the PALETTE command as follows:

PALETTE 0,15

This command assigns the color of the second number to the first number. The second number in the sequence represents a selection from the list of 16 available colors. The colors assigned to these numbers never change. The first number, however, is changeable: You can think of it as a paint bucket. At the start of the program each of the 16 buckets is filled with the color that corresponds to its number on the list. Bucket #1 has color #1 on the list, bucket #2 has color #2, and so on.

The PALETTE command simply takes a color from the list and pours it into a different bucket. In our example above, white paint is poured into the bucket that once held black. Now all graphics painted from bucket #0 will be white instead of black. In addition, any graphics already on the screen that were painted from bucket #1 will automatically change to the new color. It's important to remember that the numbers on the list never change their colors: The paint colors stay the same no matter what bucket they're put in.

The PALETTE command is used in *Animation* to create an illusion of movement. The program starts out with all of the colors set to their default colors. By changing all of the colors (buckets) to black and then making one color at a time white in the proper sequence, the program creates animation that rivals the best graphics of any computer in the PCjr's price range.

The Program

The program is surprisingly short considering the amount of graphics displayed and the breathtaking result. I chose Mode 5 because it was the only mode that offered both 16 colors and high resolution. At first I just wanted to create a grid landscape that would move toward the bottom of the screen and look as if you're flying over it. I've tried this on several other computers, and it's never been easy. The PALETTE command suggested a different approach to the problem.

Instead of drawing and erasing lines, which is very time consuming, I decided to place all of the lines on the screen beforehand and set their colors to black. The only exception was color #1, which I set to white and used to create the vertical lines for the grid. Color #0 stays black for the background. The algorithm in lines 220 and 230 draws horizontal lines across the screen, spacing them by color so that as the lines move down the screen, the distance before any color gets repeated becomes progressively longer. The vertical lines made with color #1 are then placed over the horizontal lines. When the program starts cycling the colors with white, it creates the illusion that the horizontal grid lines are moving towards the viewer, so the ground appears to be moving.

This wasn't enough for me: The top half of the display was too empty. I decided to use the same effect to create

a star field flying off the top of the screen. The final effect was dazzling! In order to move the stars through the sky with the horizontal lines on the ground, I had to draw my own lines radiating out from the center of the screen. These couldn't be ordinary lines created with the LINE command. I had to write an algorithm to draw the lines and make every pixel in the sequence a different color. The changing color is necessary for the illusion of motion. This is the slowest part of the program, as any programer who has tried to do plots the hard way, in BASIC, knows. After the system has drawn 30 of these lines on the screen, the program is ready to give you a show.

The last part of the program (lines 270 to 300) is where the excitement begins. The variables X and Y are indexes to the color numbers. The value in X represents the color number that will be set to white, and Y represents the color number that is set back to black. When either of the numbers reaches 15, it is automatically set back to 2. This way Y, with a value of 15, comes just after X, with a value of 2. When the color number in Y is turned off, we are turning off the color number that was set to white in the previous pass. This prevents the screen from filling up with white.

Line 280 is responsible for all of the changes you see on the screen. Just two PALETTE commands in BASIC cause all that commotion. Can you imagine how many lines you would have to draw, how many calculations you would have to make, and how many points you would need to plot to create all of the motion you see? Nearly 150 stars move through the heavens, all in different—apparently random—directions, with high resolution for smooth movement. And all the while, the ground appears to be moving towards you.

You may want to experiment with different values for varied effects. Here are some key areas to try and the approximate results you can expect.

Speeding Stars

To change the speed of the stars, adjust the three values in line 250 that calculate the stars' positions and spacing. Line 250 contains a PSET command in which Z is multiplied by 2 twice.

250 FOR X = 1 TO 30:Y = RND*3 + 3.21: C = INT(RND*14 + 2):FOR Z = 1 TO 50:PSET(COS(Y)*(Z*2) + COS(Y)* 50 + 160, SIN(Y)*(Z*2) + 69), C:C = C + 1:IF C > 15 THEN C = 2

If you increase the numbers by which Z is multiplied, the stars will be spaced farther apart and will appear to move faster. A smaller number will compact the stars and result in slower motion. The two numbers you multiply by Z should be fairly close in value to maintain symmetry on the screen. If you change these values very much, you may want to adjust the value 50 in the loop FOR Z = 1 TO 50. If you make the value in the PSET command larger, then you may want to make the value 50 smaller. In this loop, Z will not exceed 50. The largest value of Z multiplied by the value in the PSET command shouldn't exceed 100. If it does, the program may take longer to set up. If the value is less than 100, then the stars may disappear before they reach the edge of the screen. These equations should help you pick values.

To find the maximum value for Z:

Z = 100/X

(where X represents the value by which Z is multiplied in the PSET command)

To find the value to multiply Z by in the PSET command:

X = 100/Z

Streaking Stars

You might like to try transforming the stars into streaks of light. The stars in the *Animation* listings are one pixel in size. They can't be made any smaller, but they can be enlarged. This can be done simply by changing the value assigned to Y in line 270. If you make Y equal to 14, then the stars will be two pixels long; Y = 13 will make them 3 pixels long, and so on, until you reach Y = 2, which will produce a solid line with no movement except a black dot that flashes briefly and moves up the line.

Altered Grid

Two equations calculate the positions of the lines in the grid pattern. In line 220, the equation X = X + SQR(X)/4 determines the spacing of the horizontal lines that appear to move towards you. Changing the value 4 alters the spacing, and consequently the speed of the grid's motion. If the value is made smaller, the grid will be spread out: The horizontal lines will be farther apart, and the grid will appear to move faster. There is a logical limit to this because once you reach a certain point, you won't have enough horizontal lines remaining to give the proper effect. To slow the grid down, use a value greater than 4. Here again, if you make the value too large, the grid will become too compact and the effect will be distorted, if not destroyed.

To change the spacing of the vertical lines, you can adjust the second equation in line 240: X = X + SIN(X)/5. If the value 5 is made smaller, the spacing will be wider, and if the value is made larger, the spacing will be narrower. The limit here is that too large a value will paint the screen almost solidly with vertical lines and obliterate the scrolling horizontal lines.

Color Changes

Here are some adjustments to change the color of the graphics. Remember that the best effect is obtained by using only one color for the screen. As color #1 must be used on the vertical grid, you must first assign a new color to it. You can do this by adding this line:

185 PALETTE 1,X
(where X is the new color)

You must also change the color value in line 280. Change the value 15 in the command PALETTE X,15 to the number of the color you select. You could also change the color of the background with the command PALETTE 0,X (where X is the number of the color you choose for the background.) HCM

ANIMATION
Explanation of the Program

Line Nos.	
100-170	Program header.
180	Turn off function key display; clear screen; reserve 32K for screen display; select graphics mode 5.
190	Seed random number generator with the rightmost two digits from TIME$. (TIME$ contains the hours, minutes, and seconds.)
200	Make all colors black (except #1).
210-230	Draw horizontal lines.
240	Draw vertical lines.
250-260	Plot lines for star field; assign colors for scrolling effect.
270-300	Cycle colors between 2 and 15. Only one color will be turned on at a time, and will be set to white. All other colors are set to black.

IBM PCjr

```
100 REM ***********************
110 REM *  PCjr  ANIMATION  *
120 REM ***********************
130 REM BY WILLIAM K. BALTHROP
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM IBM PCjr
170 REM
180 KEY OFF:CLS:CLEAR ,,,32768:SCREEN
    5
190 RANDOMIZE VAL(RIGHT$(TIME$,2))
200 PALETTE 1,15
210 LINE (0,70)-(319,70),1:COL=2:X=.01
220 X=X+(SQR(X)/4):LINE (0,X+71)-(319,X
    +71),COL:COL=COL+1:IF COL=15 THEN C
    OL=2
230 IF X<129 THEN GOTO 220
240 FOR X=0 TO 3.14 STEP .05:LINE (175-
    X*10,71)-(COS(X)*400+160,SIN(X)*200
    +71),1:X=X+SIN(X)/5:NEXT X
250 FOR X=1 TO 30:Y=RND*3+3.21:C=INT(RN
    D*14+2):FOR Z=1 TO 50:PSET (COS(Y)*
    (Z*2)+COS(Y)*50+160,SIN(Y)*(Z*2)+69
    ),C:C=C+1:IF C>15 THEN C=2
260 NEXT Z:NEXT X
270 X=2:Y=15
280 PALETTE Y,0:PALETTE X,15:X=X+1:Y=Y+
    1
290 IF X>15 THEN X=2:GOTO 280
300 IF Y>15 THEN Y=2:GOTO 280 ELSE GOTO
    280
```

HCM

***Tablut* . . . from p. 60**

TABLUT (Apple)
Explanation of the Program

Line Nos.	
100-160	Program header.
170-250	Display title screen; initialize game.
260-390	Display playing screen.
400-460	Main control loop.
470-510	Keep track of players' turns.
520-570	White player inputs move.
580-620	Red player inputs move.
630-740	Check for legal moves.
750-810	Move the player.
820-850	Adjust player's location array.
860-900	Check for king at border.
910-1040	Check for captures.
1050-1120	Check for capture of king.
1130-1210	End of the game; option to play again.
1220-1500	Draw row and column labels at edge of board.
1510-1550	Subroutine to input one character between A and I.
1560-1570	Subroutine to input one character between 1 and 9.
1580-1610	Illegal move message.
1620-1650	DATA statement containing coordinates for the screen position of each board location.
1660-1690	DATA statement containing information for the TB( ) array to maintain location of pieces.
1700-1830	DATA statements contain shape table for the players' pieces and king's crown.
1840	End of the program.

APPLE II Series

```
100 REM  *********
110 REM  *  TABLUT  *
120 REM  *********
130 REM  BY  JAMES  J.  MULLIGAN
140 REM  AND  THE  HCM  STAFF
150 REM  HOME  COMPUTER  MAGAZINE
160 REM  VERSION  4.2.1
170 REM  APPLE  II  SERIES  APPLESOFT
180 REM
190 REM  TITLE,  AND  INITIALIZATION
200 REM
210 HOME : VTAB 12: HTAB 17: PRINT "TAB
    LUT": FOR X = 1 TO 1000: NEXT X: HO
    ME : GAME = 0
220 DIM TB(9,9),S(2,9)
230 KING = 0:KC = 5:KR = 5:PN = 1:FG =
    0
240 FOR X = 1 TO 2: FOR Y = 1 TO 9: REA
    D S(X,Y): NEXT Y: NEXT X
250 FOR X = 1 TO 9: FOR Y = 1 TO 9: REA
    D TB(X,Y): NEXT Y: NEXT X
260 FOR X = 24576 TO 24668: READ A: POK
    E X,A: NEXT X: POKE 232,0: POKE 233
    ,96
270 REM
280 REM  DISPLAY  PLAYING  SCREEN
290 REM
300 HGR : SCALE= 1: ROT= 0
310 HCOLOR= 3
320 HPLOT 41,8 TO 239,8 TO 239,152 TO 4
    1,152 TO 41,8
330 FOR X = 63 TO 217 STEP 22: HPLOT X,
    8 TO X,152: NEXT X: FOR X = 24 TO 1
    36 STEP 16: HPLOT 41,X TO 239,X: NE
    XT X
340 GOSUB 1250
350 FOR X = 1 TO 9: FOR Y = 1 TO 9
360 IF TB(X,Y) = 0 THEN 400
370 IF TB(X,Y) = 1 THEN HCOLOR= 3: DRA
    W 2 AT S(1,X),S(2,Y) - 2: GOTO 400
380 IF TB(X,Y) = 2 THEN HCOLOR= 3: DRA
    W 1 AT S(1,X),S(2,Y): GOTO 400
390 IF TB(X,Y) = 3 THEN HCOLOR= 5: DRA
    W 1 AT S(1,X),S(2,Y)
400 NEXT Y: NEXT X
410 REM
420 REM  ***  CONTROL  LOOP  ***
430 REM  PLAY  IS  CONTROLED  FROM  THIS  LO
    OP
440 REM
450 GOSUB 510: GOSUB 790: GOSUB 900: GO
    SUB 1090: IF KING = 1 THEN GOTO 11
    70
460 IF KING = 2 THEN GOTO 1190
470 GOTO 450
480 REM
490 REM  ADJUST  PLAYERS  TURN  NUMBER
500 REM
510 PN = ABS (PN - 1)
520 IF PN = 1 THEN 620
530 REM
540 REM  WHITE  PLAYER  INPUTS  MOVE
550 REM
560 HOME
```

Tablut

APPLE II Series

```
570 VTAB 21: HTAB 1: PRINT "WHITE MOVES
     FROM:";: GOSUB 1540: PRINT A$;:F1
    = T1: GOSUB 1560: PRINT T2;:F2 = T2
    : PRINT " TO:";: GOSUB 1540: PRINT
     A$;: GOSUB 1560: PRINT T2
580 GOTO 670
590 REM
600 REM RED PLAYER INPUTS MOVE
610 REM
620 HOME
630 VTAB 23: HTAB 1: PRINT "RED MOVES F
    ROM:";: GOSUB 1540: PRINT A$;:F1 =
    T1: GOSUB 1560: PRINT T2;:F2 = T2:
    PRINT " TO:";: GOSUB 1540: PRINT A
    $;: GOSUB 1560: PRINT T2;
640 REM
650 REM CHECK FOR LEGAL MOVES
660 REM
670 IF (F1 < > T1 AND F2 < > T2) OR (
    F1 = T1 AND F2 = T2) THEN 1610
680 IF (TB(F1,F2) = 1 AND PN = 1) OR (T
    B(F1,F2) = 2 AND PN = 1) OR (TB(F1,
    F2) = 3 AND PN = 0) OR TB(F1,F2) =
    0 OR TB(T1,T2) < > 0 THEN GOTO 16
    10
690 IF F1 = T1 THEN GOTO 720
700 FOR CM = F1 + SGN (T1 - F1) TO T1
    STEP SGN (T1 - F1): IF (TB(CM,F2)
    > 0) OR ((CM = 5 AND F2 = 5) AND (T
    B(F1,F2) < > 1 OR PN = 1)) THEN G
    OTO 1610
710 NEXT CM: GOTO 740
720 FOR CM = F2 + SGN (T2 - F2) TO T2
    STEP SGN (T2 - F2): IF (TB(F1,CM)
    > 0) OR ((CM = 5 AND F1 = 5) AND (T
    B(F1,F2) < > 1 OR PN = 1)) THEN G
    OTO 1610
730 NEXT CM
740 IF TB(F1,F2) = 1 THEN TKC = KC:TKR
    = KR:KC = T1:KR = T2: GOSUB 1090: I
    F KING = 2 THEN KING = 0:KC = TKC:K
    R = TKR: GOTO 1610
750 RETURN
760 REM
770 REM MOVE THE PIECES
780 REM
790 IF TB(F1,F2) = 1 THEN HCOLOR= 0: D
    RAW 2 AT S(1,F1),S(2,F2) - 2: HCOLO
    R= 3: DRAW 2 AT S(1,T1),S(2,T2) - 2
    :KC = T1:KR = T2: GOTO 860
800 IF TB(F1,F2) = 2 THEN HCOLOR= 0: D
    RAW 1 AT S(1,F1),S(2,F2): HCOLOR= 3
    : DRAW 1 AT S(1,T1),S(2,T2): GOTO 8
    60
810 IF TB(F1,F2) = 3 THEN HCOLOR= 0: D
    RAW 1 AT S(1,F1),S(2,F2): HCOLOR= 5
    : DRAW 1 AT S(1,T1),S(2,T2): GOTO 8
    60
820 GOTO 1610
830 REM
840 REM ADJUST PIECE LOCATION IN ARRAY
850 REM
860 TB(T1,T2) = TB(F1,F2):TB(F1,F2) = 0
    : RETURN
870 REM
880 REM CHECK FOR KINGS SUCCESS
890 REM
900 IF TB(T1,T2) = 1 AND (T1 = 9 OR T1
    = 1 OR T2 = 9 OR T2 = 1) THEN KING
    = 1: RETURN
910 IF TB(T1,T2) = 1 THEN RETURN
920 REM
930 REM CHECK FOR CAPTURES
940 REM
950 IF PN = 1 THEN MP = 3:CP = 2: GOTO
    970
960 MP = 2:CP = 3
970 IF T1 < = 2 THEN GOTO 990
980 IF TB(T1 - 1,T2) = CP AND TB(T1 - 2
    ,T2) = MP THEN TB(T1 - 1,T2) = 0: H
    COLOR= 0: DRAW 1 AT S(1,T1 - 1),S(2
    ,T2)
990 IF T1 > = 8 THEN GOTO 1010
1000 IF TB(T1 + 1,T2) = CP AND TB(T1 + 2
    ,T2) = MP THEN TB(T1 + 1,T2) = 0: H
    COLOR= 0: DRAW 1 AT S(1,T1 + 1),S(2
    ,T2)
1010 IF T2 < = 2 THEN GOTO 1030
1020 IF TB(T1,T2 - 1) = CP AND TB(T1,T2
    - 2) = MP THEN TB(T1,T2 - 1) = 0: H
    COLOR= 0: DRAW 1 AT S(1,T1),S(2,T2
    - 1)
1030 IF T2 > = 8 THEN RETURN
1040 IF TB(T1,T2 + 1) = CP AND TB(T1,T2
    + 2) = MP THEN TB(T1,T2 + 1) = 0: H
    COLOR= 0: DRAW 1 AT S(1,T1),S(2,T2
    + 1)
1050 RETURN
1060 REM
1070 REM CHECK FOR KING CAPTURED
1080 REM
1090 IF (TB(KC - 1,KR) < > 3) AND (KC -
     1 < > 5 OR KR < > 5) THEN RETUR
    N
```

Continued on p. 120

Tablut . . . from p. 119

APPLE II Series

```
1100 IF (TB(KC+1,KR)<>3)AND(KC+1<>5)OR KR<>5)THEN RETURN
1110 IF (TB(KC,KR-1)<>3)AND(KC<>5)OR KR-1<>5)THEN RETURN
1120 IF (TB(KC,KR+1)<>3)AND(KC<>5)OR KR+1<>5)THEN RETURN
1130 KING=2:RETURN
1140 REM
1150 REM END OF GAME, AND PLAY OPTION
1160 REM
1170 HOME:VTAB 21:PRINT "THE KING HAS ARRIVED AT THE EDGE OF THE BOARD. WHITE WINS THE GAME."
1180 GOTO 12000
1190 HOME:VTAB 21:PRINT "THE KING HAS BEEN CAPTURED BY THE RED ARMY. RED WINS THE GAME."
1200 VTAB 24:HTAB 1:PRINT "PLAY AGAIN (Y/N):";:GET A$:IF A$<>"N" AND A$<>"Y" THEN GOTO 1200
1210 IF A$="Y" THEN RUN
1220 GOTO 18400
1230 REM SUBROUTINE TO DRAW LABELS FOR THE BOARD
1240 REM
1250 HCOLOR=7:HPLOT 52,7 TO 52,3 TO 54,1 TO 55,1 TO 57,3 TO 57,7:HPLOT 52,4 TO 55,4
1260 HPLOT 74,1 TO 77,1 TO 78,2 TO 78,3 TO 77,4 TO 78,5 TO 78,6 TO 77,7 TO 74,7 TO 74,1:HPLOT 75,4 TO 76,4
1270 HPLOT 99,1 TO 96,1 TO 96,7 TO 99,7
1280 HPLOT 118,1 TO 121,1 TO 122,2 TO 122,6 TO 121,7 TO 118,7 TO 118,1
1290 HPLOT 143,1 TO 140,1 TO 140,7 TO 143,7:HPLOT 141,4 TO 142,4
1300 HPLOT 165,1 TO 162,1 TO 162,7:HPLOT 163,4 TO 164,4
1310 HPLOT 188,1 TO 184,1 TO 184,7 TO 188,7:HPLOT 187,4
1320 HPLOT 204,1 TO 204,7:HPLOT 208,1 TO 208,7:HPLOT 205,4 TO 207,4
1330 HPLOT 228,1 TO 228,7
1340 HPLOT 255,11 TO 255,22:HPLOT 255,27 TO 255,38:HPLOT 260,27 TO 260,38
1350 HPLOT 253,11 TO 257,11:HPLOT 253,22 TO 257,22:HPLOT 253,27 TO 262,27:HPLOT 253,38 TO 262,38
1360 HPLOT 255,43 TO 255,54:HPLOT 260,43 TO 260,54
1370 HPLOT 253,43 TO 267,43:HPLOT 253,54 TO 267,54
1380 HPLOT 255,59 TO 255,70:HPLOT 260,59 TO 263,70 TO 266,59
1390 HPLOT 253,59 TO 268,59:HPLOT 253,70 TO 268,70
1400 HPLOT 255,75 TO 258,86 TO 261,75
1410 HPLOT 253,75 TO 263,75:HPLOT 253,86 TO 263,86
1420 HPLOT 255,91 TO 258,102 TO 261,91:HPLOT 265,91 TO 265,102
1430 HPLOT 253,91 TO 267,91:HPLOT 253,102 TO 267,102
1440 HPLOT 255,107 TO 258,118 TO 261,107:HPLOT 265,107 TO 265,118:HPLOT 253,107 TO 270,118
1450 HPLOT 253,107 TO 272,107:HPLOT 253,118 TO 272,118
1460 HPLOT 255,123 TO 258,134 TO 261,123:HPLOT 265,123 TO 265,134:HPLOT 253,123 TO 270,134:HPLOT 275,123 TO 275,134
1470 HPLOT 253,123 TO 277,123:HPLOT 253,134 TO 277,134
1480 HPLOT 255,139 TO 255,150:HPLOT 260,139 TO 266,150:HPLOT 260,150 TO 266,139
1490 HPLOT 253,139 TO 268,139:HPLOT 253,150 TO 268,150
1500 RETURN
1510 REM
1520 REM INPUT SUBROUTINES
1530 REM
1540 GET A$:IF A$<"A" OR A$>"I" THEN PRINT CHR$(7);:GOTO 1540
1550 T1=ASC(A$)-64:PRINT CHR$(7);:RETURN
1560 GET A$:IF A$<"1" OR A$>"9" THEN PRINT CHR$(7);:GOTO 1560
1570 T2=ASC(A$)-48:PRINT CHR$(7);:RETURN
1580 REM
1590 REM ILLEGAL MOVE MESSAGE
1600 REM
1610 HOME:VTAB 21:PRINT "ILLEGAL MOVE, TRY AGAIN.";:PRINT CHR$(7);CHR$(7);:GOTO 520
1620 REM
1630 REM BOARD COORDINATES
1640 REM
```

Continued on next page

Tablut

APPLE II Series

```
1650 DATA 60,82,104,126,148,170,192,21
     4,236,14,30,46,62,78,94,110,126,142
1660 REM
1670 REM  PIECE POSITIONS
1680 REM
1690 DATA 0,0,0,3,3,3,0,0,0,0,0,3,0,0,0,3,0
     ,0,0,0,0,0,2,0,0,0,0,0,0,0,0,2,2,
     0,0,0,0,3,3,3,2,0,1,2,0,2,3,0,3,0,0,0,0,2,2,
     ,0,0,0,0,3,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,
     3,0,0,0,0,0,0,0,0,0,3,3,3,0,0,0,0
1700 REM
1710 REM  **** SHAPE TABLE ****
1720 REM
1730 REM  TABLE INDEX
1740 REM
1750 DATA 2,0,6,0,56,0
1760 REM
1770 REM  PLAYERS SOLDIER
1780 REM
1790 DATA 63,63,63,55,54,46,53,46,53,45,5
     4,39,60,39,60,63,54,63,54,62,39,36,36,45,45
     36,45,36,36,63,63,54,63,54,62,36,36,45,45
     5,45,45,37,60,36,53,37,53,62,54,45,54
1800 REM
1810 REM  KINGS SHAPE
1820 REM
1830 DATA 54,62,54,55,62,54,55,63,63,
     63,63,60,36,39,60,36,39,36,52,53,53,45,54
     ,45,44,44,52,53,45,44,52,53,53,45,44
     4,44,52,54,0
1840 END
```

HCM

Tablut . . . from p. 60

TABLUT (TI-99/4A)
Explanation of the Program

Line Nos.

100-170	Program header.
180-220	Define graphics characters and colors.
230	Display title screen.
240-280	Initialize game; set up board.
290-300	Main control loop.
310-410	Initialize the board array.
420-500	Display the board.
510-650	Input players' moves; check entries.
660-680	Illegal move message.
690-800	Check for legal moves.
810-850	Move player; update arrays.
860-990	Check for captures.
1000-1050	Check for capture of king.
1060-1120	End of game message; option to play again.
1130	Input column letter.
1140	Input row number.
1150	Delete captured player.
1160	Clear left side of display.
1170	End of program.

TI-99/4A

```
 100 REM  **********
 110 REM  *  TABLUT  *
 120 REM  **********
 130 REM  BY JAMES J. MULLIGAN
 140 REM  HOME COMPUTER MAGAZINE
 150 REM  VERSION 4.2.1
 160 REM  TI EXTENDED BASIC
 170 REM
 180 CALL CLEAR :: GAME=0
 190 CALL SCREEN(4)
 200 CALL CHAR(120,"FF8080808080808080FF00
     0000000000808080808080808000")
 210 SOL1$="0101010707070707C4C4C4F4F4
     FCF4070703030303030700E4E4646464647440
     " :: CALL CHAR(128,SOL1$,136,SOL1$
     )
 220 CALL CHAR(140,"0001010D0D6F6F7F7F0000E0
     80B0F6F6FE7F7F3F1F0000FEFE
     EFCF8") :: CALL COLOR(12,2,12,13,16,
     2,14,7,16)
 230 DISPLAY AT(10,11):"TABLUT" :: DISPL
     AY AT(12,7):"BY J. MULLIGAN" :: FOR
     Z=1 TO 15 :: CALL SOUND(-10,110*Z,
     0) :: NEXT Z
 240 REM **INITIALIZE***
 250 DIM TB(9,9)
 260 KING=0 :: KC=5 :: KR=5 :: PN=1 :: F
     G=0
 270 GOSUB 310 :: CALL CLEAR :: CALL SCR
     EEN(8)
 280 GOSUB 420
 290 GOSUB 510 :: GOSUB 810 :: GOSUB 870
     :: GOSUB 1010
 300 IF KING=1 THEN GOTO 1070 ELSE IF KI
     NG=2 THEN GOTO 1090 ELSE GOTO 290
```

TI-99/4A

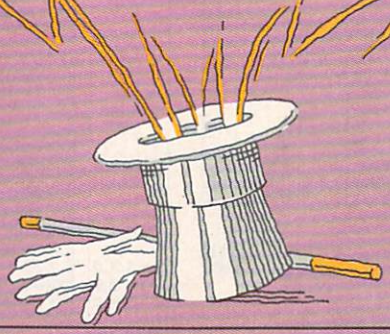
```
 310 REM  **SET UP BOARD**
 320 REST ORE 330 :: FOR R=1 TO 9 :: FOR
     C=1 TO 9 :: READ TB(R,C) :: NEXT C :
     : NEXT R
 330 DATA 0,0,3,3,3,0,0,0,0
 340 DATA 0,0,0,0,0,0,0,0,3
 350 DATA 0,0,0,0,0,0,0,0,3
 360 DATA 0,0,3,3,2,0,0,0,3
 370 DATA 0,0,3,2,1,2,0,0,3
 380 DATA 3,3,0,0,2,0,0,3,3
 390 DATA 0,0,0,0,2,0,0,0,0
 400 DATA 0,0,0,0,3,3,0,0,0
 410 DATA 0,0,0,3,3,0,0,0,0
 420 REM  ***PRINT BOARD***
 430 DISPLAY AT(1,9):"A B C D E F G H I"
 440 REST ORE 470 :: FOR R=2 TO 19 :: REA
     D A$ :: FOR C=1 TO 18 :: CALL HCHAR
     (R,C+10,ASC(SEG$(A$,C,1))+55) :: NEX
     T C
 450 IF INT(R/2)=R/2 THEN CALL HCHAR(R,3
     0,INT(R/2)+48)
 460 NEXT R :: RETURN
 470 DATA ABABABABIJIJIJABABABAB,CDCDCDKLKLKD
     LCDCDCDCDCD,ABABABABIJABABABABAB,CDCDCDCDCD
     LCDCDCDCDCD,ABABABABQRABABABAB
     DATA CDCDCDSTCDCDCDCDCD,IJABABABQRAR
     BVQRQRIJIJIJ,KLCKLKLSTSTWXSTSTKLKL
     DDATA IJABABABQRABABABIJ,KLCDCDCDCDSTC
     DCDCDCDKL,ABABABABABAB,CDCDCDCDCDCD
     DSTCDCDCDCDCD,ABABABABIJABABABABAB,CDCDCDC
     DDATA ABABIJIJIJABABABAB,CDCDCDKLKLK
     LCDCDCDCD
 510 REM  **PLAYER INPUT***
 520 PN=ABS(PN-1)
 530 IF PN=1 THEN GOTO 600 ELSE GOSUB 11
     60
 540 DISPLAY AT(5,1)SIZE(4):"RED" :: DIS
     PLAY AT(6,1)SIZE(5):"MOVES" :: DISP
     LAY AT(7,1)SIZE(8):"FROM: _"
 550 GOSUB 1130 :: CALL HCHAR(7,8,K1) ::
     CALL HCHAR(7,10,95)::: GOSUB 1140 :::
     CALL HCHAR(7,10,K2)::: F2=K1-64 :::
     F1=K2-48
 560 IF PN=0 AND(TB(F1,F2)<>1 AND TB(F1,
     F2)<>2)THEN GOTO 670
 570 DISPLAY AT(8,1)SIZE(8):"TO: _"
 580 GOSUB 1130 :: CALL HCHAR(8,8,K1) ::
     CALL HCHAR(8,10,95)::: GOSUB 1140 :::
     CALL HCHAR(8,10,K2)::: T2=K1-64 :::
     T1=K2-48
 590 GOTO 660
 600 GOSUB 1160
 610 DISPLAY AT(10,1)SIZE(5):"WHITE" :::
     DISPLAY AT(11,1)SIZE(5):"MOVES" :::
     DISPLAY AT(12,1)SIZE(8):"FROM: _"
 620 GOSUB 1130 :: CALL HCHAR(12,8,K1)::
     CALL HCHAR(12,10,95)::: GOSUB 1140 :::
     CALL HCHAR(12,10,K2)::: F2=K1-64 :::
     F1=K2-48
 630 IF PN=1 AND TB(F1,F2)<>3 THEN GOTO
     670
 640 DISPLAY AT(13,1)SIZE(8):"TO: _"
 650 GOSUB 1130 :: CALL HCHAR(13,8,K1)::
     CALL HCHAR(13,10,95)::: GOSUB 1140 :::
     CALL HCHAR(13,10,K2)::: T2=K1-64 :::
     T1=K2-48
 660 IF F1=T1 OR F2=T2 THEN GOTO 690
 670 DISPLAY AT(21,3):"ILLEGAL MOVE, TRY
     AGAIN." :: CALL SOUND(50,110,0,220
     ,0,330,0)
 680 FOR DELAY=1 TO 260 :: NEXT DELAY ::
     CALL HCHAR(21,1,32,32)::: GOTO 530
 690 IF T1<>F1 AND T2<>F2 THEN GOTO 670
 700 IF T1=F1 THEN GOTO 740
 710 FOR CM=F1+SGN(T1-F1)TO T1 STEP SGN(
     T1-F1)::: IF TB(CM,F2)>0 THEN GOTO 6
     70
 720 IF (CM=5 AND F2=5)AND(TB(F1,F2)<>1
     OR PN=1)THEN GOTO 670
 730 NEXT CM :: GOTO 790
 740 IF F2=T2 THEN GOTO 670
 750 FOR CM=F2+SGN(T2-F2)TO T2 STEP SGN(
     T2-F2)::: IF TB(F1,CM)>0 THEN GOTO 6
     70
 760 IF (CM=5 AND F1=5)AND(TB(F1,F2)<>1
     OR PN=1)THEN GOTO 670
 770 IF TB(F1,F2)=1 THEN KCT=KC :: KRT=K
     R :: KC=T1 :: KR=T2 :: GOSUB 1010 ::
     IF KING=2 THEN KING=0 :: KC=KCT ::
     KR=KRT :: GOTO 670
 780 NEXT CM
 790 IF TB(F1,F2)=1 THEN KCT=KC :: KRT=K
     R :: KC=T1 :: KR=T2 :: GOSUB 1010 ::
     IF KING=2 THEN KING=0 :: KC=KCT ::
     KR=KRT :: GOTO 670
 800
 810 RETURN
 820 CALL HCHAR(F1*2,F2*2+9,120)::: CALL
     HCHAR(F1*2,F2*2+10,121)::: CALL HCHA
     R(F1*2+1,F2*2+9,122)::: CALL HCHAR(F
     1*2+1,F2*2+10,123)
 830 ZZ=TB(F1,F2):: IF ZZ=1 THEN CH=140
     ELSE IF ZZ=2 THEN CH=136 ELSE IF ZZ
     =3 THEN CH=128
```

Continued on p. 151

SPELLAKAZAM

by Sharyn Lyon
HCM Staff

Many software developers and major educational publishers are presently working together for the purpose of producing educationally sound software. One such pairing—between DesignWare and Silver Burdett Company—has resulted in *Spellakazam*, an educational game for ages 7 to 14.



Name:	Spellakazam
Program Type:	Educational
Machine:	Apple II+, IIe, or Commodore 64 (available for IBM PC & PCjr in March, 1984)
Author:	Marguerite Frick
Distributor:	DesignWare 185 Berry Street Building Three, Suite 158 San Francisco, CA 94107
Price:	$39.95, disk
System Requirements:	48K memory; 1 disk drive

	poor	fair	good	excellent
Documentation				
Independence				
Graphics				
Rewards				
Concept Presentation				

It's A Game!

Spellakazam is no mere purple and white ditto rescued from extinction by the new technology. It is a real computer game. You are in competition with the Magician (a.k.a. the computer) in a race through an alphabetical maze of color and letters. Whoever spells the magic word and gets to the magician's hat first gets the most points for that round. If you spell the word correctly, a creature pops out of the hat. If your spelling is incorrect, the magic hat disappears, the correct spelling appears on the screen, and you lose some of your points.

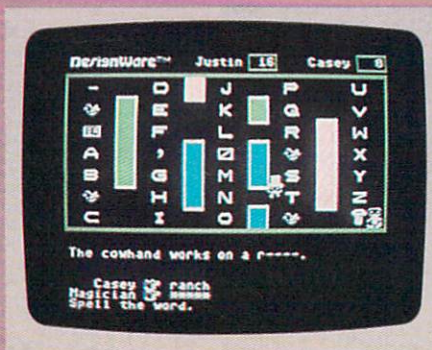
The instructions on the screen and in the extensive documentation are presented at an appropriate reading level, but they aren't really necessary. If you don't press [RETURN] or [f1] when the program begins, it automatically takes you through a demonstration of Game I and Game II.

Spellakazam is as much a learning tool as it is an educational game. The first page of the documentation is "A Note to Parents" that explains why the words appear in sentences, how practice and the results of educational research have been built into the game, and why particular words were chosen. Parents are shown how they can help their children get the most out of the software.

Magic Words

Students from first through eighth grade will be challenged by the 20-word lists built into *Spellakazam*. Because users can set their own skill levels, word list lengths, word list topics, and can even make up their own word lists, children beyond the suggested grade levels can benefit from playing this spelling game. The word lists on the elementary school level give an excellent indication of the concepts in this package. The elementary word lists can be seven to 30 words long and can span concepts from the short vowel sounds on the second grade level, through vowels and consonant clusters on the third and fourth grade levels, to short vowels for fifth graders. After you make your list and game selections, the spelling rule around which the list was built will appear on the screen like this:

The short vowels /a/,/e/,/i/,/o/,/u/ are usually spelled with single letters.



Next, the actual list of words that will be used in the game appears with a prompt that asks if you want to use it, modify it, or pick another list.

Parents and teachers who create their own student-specific lists will be pleased to find that they can also customize all of this accompanying material. Parents could, for example, create a list of words from their child's favorite book, help the child look over the words to see if they have any spelling element in common, and then help the child actually compose the rule for that list.

Performance

Spellakazam on the Apple is much faster and more responsive than it is on the Commodore 64. Setting up the Commodore to play the game takes almost 10 minutes, and there is a 30-second wait each time you change modes. Although a colorful little disk flashes reassuringly in the screen's lower right corner, it doesn't shorten the wait. Not only is the process lengthy, but the loading information in the documentation is incomplete. To load the program, we had to type in this load message: LOAD"**",8,1. The Apple version, on the other hand, boots automatically and moves swiftly from one screen to the other. Once you get into the game, however, Apple's version responds to a keypress only slightly faster than Commodore's.

Joystick fans will be more pleased with the game's performance. In fact, a Commodore devotee who doesn't own a joystick might consider investing in one; it takes a lot of playing to get good enough on the C-64 keys to make a real race out of beating the Magician to the correct letters and then to the magic hat.

Now You See Him...

The Magician can be an adversary or an ally. First-time players might have more fun sending him back to the beginning of the maze than they will trying to spell the word. Actually, sending the Magician back is a good strategy to use to buy time to spell a difficult word. This technique is most useful when playing Game One. In Game One you must spell the word before going to the hat, but the Magician can go directly to the hat. In Game Two he too has to spell the word before going to the hat. In Game Two, then, you can

Continued on p. 146

THE BEST OF 99'er™

Pick up a copy at your favorite bookstore or dealer. To order direct, use form on page 165.

A Giant Home Computer Compendium™ for the Texas Instruments 99/4A

VOLUME 1

The largest, most comprehensive collection of programs and articles ever assembled for the TI Home Computer

- Over 200 thoroughly tested key-in-and-RUN programs and sub-programs typeset in a grid format for maximum clarity.
- Programming instruction in 4 languages—learn to use BASIC, Extended BASIC, LOGO and Assembly Language—for everything from record keeping and money management to arcade-quality action games.
- A selection of sensational game software featuring full-color graphics, animation, and sound effects.
- Beyond the owner's manual—tips and techniques for getting the most out of your computer system.
- Computer-Assisted Instruction—The home computer becomes your private tutor.
- Page after page of innovative applications—transforming your computer into a home productivity center.

People Who Know the Home Computer Best™

HOME COMPUTER magazine™

Emerald Valley Publishing Co.
P.O. Box 5537, Eugene, OR 97405

AN EMERALD VALLEY BOOK

TEXAS INSTRUMENTS 99/4A PRODUCTS

LIMITED SUPPLY

Back Issues of 99'er HOME COMPUTER magazine™

Each Only $3.95 postpaid

Canada $4.50, Foreign Surface $5.50, Foreign Air $7.50

ISSUE #1 ISSUE #3 ISSUE #5
ISSUE #2 ISSUE #4

OUT OF PRINT

ISSUE #6 (Partial Contents)
• How To Produce Sound Effects • Debugging a Game Program • How to Start a User's Group • Verbose: A speech Vocabulary Expansion Aid • Color Mapping • Dynamic Manipulation of Screen Character Graphics • The Beginner's Guide to Cassette Operation With the Home Computer • Pre-School Block Letters and Data Compaction • Picking the Ponies in TI BASIC • Battle Star Space Game • 3-D Animation on the Home Computer • Programming Tips • Who is LOGO for? • Tower of Hanoi in TI LOGO • A Review of the TI Lesson-Development Software • An Interview with a Game Designer • Learning Assembly Language with a Magic Crayon • and much, much more.

NOVEMBER 1982 (Partial Contents)
• Chatting with Your Micro: Languages for the Home Computer • A Review of the Smith Corona TP-1 Daisy Wheel Printer • The Micro Jaws Arcade Game • A Knight's Tour in TI BASIC • LOGO Has Style • ASPIC: A Language for Children • A p-System Beginners' Tutorial • An Interview with a p-System Pioneer • A Mini-Memory Screen Dump to the Home Computer Printer • Up Scope!—An exciting Undersea Combat Game • Strategy for Munch Man • A Brief Encounter with a TI Hand-Held Computer • 99'er Shopping Bus • A Pocket Battleship • Sub-Programs in Extended BASIC • Arcade & Adventure Game Reviews • and much, much more.

DECEMBER 1982 (Partial Contents)
• Tex-Scribe: A Text Editor for the Home Computer • A Christmas Computer Carol • Managing a Mailing List the Futura Way • Parsec: The Arcade Game • Plotting With the Home Computer—Pixel by Pixel • Preventing the Situation—Oh No! Memory Full • A Colorful Tour of TI-Fest: The Home Computer Show • Santa's Workshop: The Making of a Home Computer • The Turtle Arcade: Movies & Video Games in LOGO • Controlling a BASIC Termite • The 99'er Gold Rush—An Arcade/Adventure in the Home • 99'er Digest of News & Happenings in the TI World • Plus Games, Reviews, and much, much more.

JANUARY 1983 (Partial Contents)
• Computer Assisted Instruction for the Handicapped • p-System Basics • Debugging in LOGO • The Dow-4 Gazelle Flight Simulator • Note Whiz and Pitch Master Musical Game Reviews • Learning With the PLATO Computer Library • Strategies for Adventure Gaming • Death Drones • Using the Line-By-Line Assembler • Close Encounters of the Simon Kind • Electrical Engineering Education Program • Interview With an Arcade Game Designer • TI Invaders • Programming With Pascal • Cyber-Dice • News and Happenings in the Home Computer World • Arcade Game Reviews • The Thief Adventure Game • Programming Tips • and much, much more.

FEBRUARY 1983 (Partial Contents)
• Texas Instruments at the Winter Consumer Electronics Show • Home Computer Printers on Review • How to Create Math Daisies in LOGO • Vectors in LOGO • ASPIC: A Language for Teachers • The Joys of Adventuring—Part 2 • Programming Pointers with Chuck-A-Luck—Part 4 • Interview With the Voice of Parsec • Why You Need a Printer for Your Home Computer • Lifeline to Titan Space Game • Night Blockade Battleship Game • Tower of Hanoi Pocket Program • Computer Gaming Software Reviews • News of Late Developments in the World of Home Computers • and much, much, more.

MARCH 1983 (Partial Contents)
• An Introduction to the TI-99/2 Basic Computer • The Hex-bus and the 4/A Connection • Making Your Own Say and Spell Game • Disabled Children Learn and Grow • Super Cataloger—A Review of a Disk Library Utility Program • TI's New CC-40 Compact Computer • Robots and Their Social Impact • Twenty Questions With Robot Redford • The Gravity of LOGO • Joystick Jockey—An Overview of Remote Controllers • Parsec Strategy • Converting Extended BASIC to Assembly Language • Matrix Muncher • Mini Memory Disassembler Utility • Pulling the Shade on Sprites • Letters on LOGO • Tiny Tutorials • Games programs, reviews, and much, much more.

APRIL 1983 (Partial Contents)
• Computer Assisted Savings Planning to Build Your Nest Egg • Tex-Cipher Writes and Decodes Secret Messages • Crossbytes—Computer Vocabulary Crossword Puzzle • Cutting Corners On Your Food Budget Using Coupons • Introducing Financial Planning with Multiplan • The Design Philosophy of the Compact Computer • LOGO Takes On the Popular Fifteen Puzzle • Super Language—Programming Sprites in Mini Memory • Colorful Word World—Reading Readiness for Pre-schoolers • Gameware Buffet's A-Maze-ing Boa Alley Game • Giant and Dwarfs Entrapment Game • Game Reviews • Programming Tips • Money Saving Hints • and much, much more.

MAY 1983 (Partial Contents)
• A Consumer's Guide to Word Processing • Word Processing Market Basket • A Generalized Filing Program for VIPs • The Multiplan Medium Balances Your Checkbook and Budget • Activity Accountant Helps School Secretaries with Extracurricular Activities • Maximizing Your Mini Memory's 4K of RAM • Exploring Enhanced BASIC on the Compact Computer • The LOGO Tortoise Debates the BASIC Hare • A Pocket Program to Organize Data with Linked Lists • Mentally Handicapped Learners Team Up with the TI-99/4A • The Wonders of Diskette Storage • Beeline—a Multi-Screen Strategy Game • Lost Ruins—an Archeological Dig Game • 3-D Illusions with Sprites in Depth • Game reviews, Group Grapevine, and much, much more.

JUNE 1983 (Partial Contents)
• Children and Computers Make the 99/4H Connection • Tune Your Guitar with Our TI Tuning Fork • Talk to Your Computer—Voice Technology Is Here • Gameware Buffet's Eat or Be Eaten Aardvark Game • Protect Your Station in the Space Zapper Game • What Multiplan Can and Can't Do • Understanding Inputs and Outputs in Drive For Diskettes—Part 2 • Calculate Loan Schedules on the CC-40 • Go on a LOGO Vacation • Letters on LOGO • A Review of Upper Room Software's Programs for Special Learners • Construct an RS232/Joystick Interface • Group Grapevine • Shopping Bus • A Natural Language Interface for the Professional • Game Reviews • and much, much more.

JULY 1983 (Partial Contents)
• The Evolution of Home Computer Graphics Comes Alive in Graphics Grows Up • Five Data Organizers in Never Out of Sorts • TI & 99'er at the Consumer Electronics Show • WarGames: The Movie and the Book • Editing with Multiplan • The LOGO Logician Presents To Model Is to Learn • LOGO Mosaic Designs Fill the Screen • Your Speech Synthesizer as a Spelling and Foreign Language Teacher • Software for Your Low-cost Printer Port • Gameware Buffet's Treasure Island and the Colorful Switch-A-Row • A Book Review of Learn BASIC for CC-40 Users • 3-D Animation with the TMS 9918A Video Chip • Games Reviews • Group Grapevine • and much, much more.

AUGUST, 1983 (Partial Contents)
• The Home Computer Goes To Work • Bit One, Purl Two at the Fashion Factory • Better Business Bar Graphs in Graphic Persuasion • An Ensemble of Assemblers • Cashflow Helps Money Management • Keystrokes for Thrifty Folks—a Review of Typwriter • Game Reviews of Cavern Quest and Starprobe '99 • Counting Fun for Preschoolers • Peripheral Vision 99 • Mean Machines and Small Potatoes • Multiplan Medium Groups Cells into Rectangles • Turtle Text, a LOGO Word Processor • Group Grapevine • Hello, Little Brother—CC-40 Speaks to 99/4A • Plato's Progress Looks at Student Assessment • Pocket Sunrise • Public Investigator • Gameware Buffet's Jungle Jim, Success Formula, and much, much more!

SEPTEMBER, 1983 (Partial Contents)
• Adding on to Your Home Computer • 99'er Directory of Commercially Available Software, Accessories and Peripherals • Peripheral Vision 99 Hardware Reviews • Pocket Sunrise, Part Two in Extended BASIC • Byte Lightning—Mini Memory Plays a Game • TI-WRITER Tutorial • LOGO's Functions, Sets and Turtles • The CC-40 and 99/4A Take the Data, and RUN • PLATO's Progress Looks at Geometry Courseware and the Shape of Things to Come • Gameware Buffet's Challenge of Camelot and The Fly • Game Reviews of Nit Wit and Crime and Punishment • Group Grapevine • 99'er Hall of Fame • 99'er Digest Update on New Products, and much, much more!

OCTOBER, 1983 (Partial Contents)
• Adventures and Fantasy with Your Home Computer • Do-It-Yourself Adventure • Picoprocessor Emulator, a 4-Bit Microprocessor • Have No Fear: Assembly Language Won't Byte, Part 1 • Make Your Mark • Les Izmore and Debug Computing Cartoon • Once Upon a Tortoise Shell—A LOGO Adventure • Turtle Strut • Number Nibbler for Children • Lots of Plots on Your CC-40 and HX1000 Four-Color Printer/Plotter • Multiplan Medium—Bartender • A Grisly Adventure Bear Hunt • Escape From Wizard's Keep in Extended BASIC • Game Reviews • Shopping Bus • and much, much more.

NOVEMBER, 1983 (Partial Contents)
• Education with your Home Computer • Five Creative Learning Activities for Children • Let's Build America • Have No Fear: Assembly Language Won't Byte, Part 2 • Squeezing the Most Out of TI BASIC • TI-WRITER Tutorial • LOGO Lexicon • Interview with Dale Osborn • TI-WRITER: At Home In the Office • PLATO's Progress • Les Izmore and Debug • The Multiplan Medium • Computer Arrested Instruction—99'er Interviews the Kids • Gameware Buffet: Taco Man and Robo Chase • Game Reviews of Jail Break and Arithmetex • Hall of Fame • 99'er Digest • and much, much more.

Keep Your Back Issues & Tapes Together With a FINDER-99'er BINDER™

Only $11.95

- Big enough to hold 6 magazines and 12 tapes
- Uses wire straps to hold magazines so that no hole punching is necessary
- Attractive and Durable

$16.95 with 6 Blank Tapes

$21.95 with 12 Blank Tapes

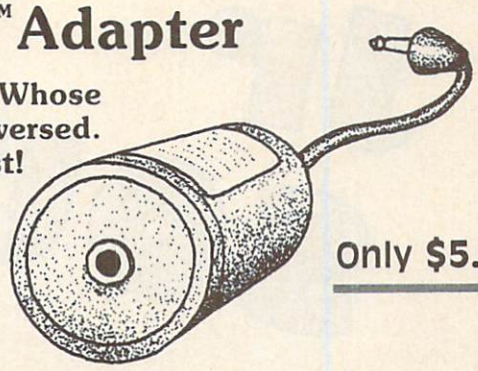
12 High-Quality 99'er-WARE C-10 Digital Computer Cassettes Available Separately for $14.00

99'er.WARE™
P.O. Box 5537
Eugene, Oregon 97405
Tel. (503) 485-8796

The TEX-SETTE™ Adapter

For Those Cassette Recorders Whose Remote Jack Wiring is Reversed. Cassette Compatibility At Last!

If the TI-99/4A will not control your cassette recorder's motor (automatic start/stop), we have the solution for you.



Only $5.95

DUST COVERS

Features:

- Equipment Protection
- Handsome Appearance
- Custom-Fit
- Antistatic Treated
- Quality Construction

A	10″ Color Monitor Cover	$10.95
	13″ Color Monitor Cover (now shown)	$12.95
B	Peripheral Expansion Box Cover	$12.95
C	TI's 99/4 Matrix Printer Cover (same as Epson MX-80 Cover)	$ 9.95
D	Cassette Recorder Cover (1 size fits up to 10″ x 6″)	$ 4.95
E	TI-99/4(A) Console Cover	$ 8.95
F	Speech Synthesizer Cover	$ 3.95
G	Peripheral Box Cover (Specify: 32K Memory Expansion RS232 Interface, or Disk Controller)	$ 5.95
H	Thermal Printer Cover	$ 8.95
I	Disk Memory Drive Cover	$ 5.95

Dealer Inquiries Invited

NAME _____ PLEASE
ADDRESS _____ PRINT
CITY _____ STATE _____ ZIP _____

Check enclosed ☐ MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my: ☐ VISA ☐ Master Card ► _____ Expiration Date

Account No.

Tel. No. _____ Signature _____

QTY	ITEM(S)	PRICE	SHIPPING	AMOUNT
	Best of 99'er	$19.95	$1.50 ea.☆	
	Best of 99'er On Tape	$35.00	$1.00 ea.☆	
	Book and Tape Package	$49.95	$2.50 ea.☆	
	Finder Binder (no tapes)	$11.95	$1.50 ea.☆	
	Finder – Binder w/6 blank tapes	$16.95	$1.50 ea.☆	
	Finder-Binder w/12 blank tapes	$21.95	$2.50 ea.☆	
	12 Blank Tapes	$14.00	$2.00 ea.	
	Texsette Adapter	$5.95 ea.	.50 ea.	
			Subtotal	

Qty	ITEM(S)	PRICE	SHIPPING	AMOUNT
	Back Issues (Circle Issues Desired) Volume 1 No. 6, Nov '82, Dec '82, Jan '83, Feb '83, Mar '83, Apr '83, May '83, Jun '83, Jul '83, Aug '83, Sept '83, Oct '83, Nov '83 (Handling incl.)	$3.95 ea.	Foreign Surface add $2.00 ea. Foreign Air add $4.00 ea.	
	Dust Covers A B C D E F G H I		.50¢ ea. cover	
	99'er Directory (Handling incl.)	$5.00		

☆ Foreign surface add $3.00 for shipping.

Foreign Air Inquire

Subtotal
Total Order
Handling $1.50
TOTAL PAYMENT ENCLOSED

Enclose check, money order or credit card information to:
99'er ware, Dept M1
P.O. Box 5537
Eugene, OR 97405
or
Use VISA/MasterCard and Order Toll-Free
1-800-828-2212
In Oregon, Alaska, and Hawaii telephone
(503) 485-8796

Please allow 10 days for your order to be processed.

Jr SOUNDS OFF

Using the PCjr Sound Chip

by W.K. Balthrop
HCM Staff

The IBM PCjr's many new additions to the standard capabilities of the IBM PC make the PCjr an attractive and salable computer. Unfortunately, in order to make it compatibile with its big brother, IBM equipped the PCjr with Cassette BASIC as its primary programming language. If you are familiar with the PC version of Cassette BASIC, then you already know PCjr's Cassette BASIC because they are identical.

The major drawback to Cassette BASIC on the PCjr is that it does not allow you to access the new enhancements incorporated into the PCjr with special commands such as SOUND and PLAY. These enhancements include the improved sound capabilities added with the SN76496N complex sound generator. This is the same sound chip found in the TI-99/4A Home Computer. It allows you to produce up to three tones and one noise at the same time, and they are much better quality than the single tone generated with the 8253 programmable timer. This timer is also built into the PCjr, and is Cassette BASIC's only means of sound generation with the SOUND statement.

Don't get too discouraged, though. You can access the SN76496 complex sound generator from Cassette BASIC using the OUT command. The OUT command allows you to output data to the system's many output ports. With the OUT command, you have more direct control over the sound chip than with the SOUND and PLAY commands of Cartridge BASIC.

In this article we will cover one aspect of the OUT command's capabilities in producing sound effects with the SN76496N complex sound generator. In future articles we will cover further aspects of this command, as well as the INP command, which reads a port.

The OUT command allows you to directly output data to the system ports. Many of the system's features are accessible through these ports, including the entire sound subsystem, cassette port, graphics subsystem, disk drive, serial port, modem, and keyboard.

The Sound Subsystem

In order to access the sound subsystem, you must have a general knowledge of how it operates. The PCjr can send sound out to the speaker of your television, monitor, or remote amplifier and speaker in four ways. The first way is via the 8253 programmable timer. This chip, the same as that in the IBM PC, is capable of producing one frequency at a time. The second method is to output the sound that comes from the cassette recorder interface. The third method is the I/O channel. The I/O channel is the port at the side of the main unit that attaches to the parallel interface. In the future, you may be able to attach a speech synthesizer to the I/O channel port and have the speech output to your television's speaker. The fourth mode of generating sound, and the one we will cover in more detail in a moment, is the SN76496N complex sound generator.

All four of these sound sources are fed into an analog multiplexer (MPX)—a device that can take several inputs and select one of them for an output. By setting the proper select lines to the MPX, you can choose any one of the four sound sources. The MPX output is fed into a low pass filter and an amplifier, then split off to the three different outputs—the direct drive monitor audio, R.F. modulator audio, and the external audio jack. The signal is presented to all three outputs at once, so that if more than one output were in use, each of them could be heard.

One other route for the audio is accessible only from the 8253 programmable timer and from none of the other sound sources. The audio can go through the internal speaker, whose access can also be controlled with the OUT command. Access to both the MPX and the internal speaker is through the same port. The port address used to access them is 97. Its command would look like this:

OUT 97,x where x is the output value.

Each of the eight output bits in the port has a special function, and they should all be noted. The following is a list of their functions:

Functions for Output Port Address 97.
This port is part of the PPI 8255 device.

BIT #	FUNCTION
0	This bit controls the gate input of the 8253 timer. When this bit is set low (0), the input is disabled and the counter is halted. When set high (1), this bit enables the counter.
1	When set to 1, this bit enables the 8253 timer's output. If set to 0, then the output will be forced to zero. This bit can be used to modify the output of the timer.
2	This bit is used to transfer data from memory into the video gate array. This bit should be set to 1 when you use any of the alpha modes. When you use the graphics modes, this bit should be set to 0.

BIT #		FUNCTION
3		This bit is used with bit 4 to control the cassette motor. When this bit is 1, the cassette motor will be off. When this bit and bit 4 are zero, then the cassette motor will be on.
4		This bit controls the internal speaker and the cassette motor. If this bit is set high (1), the internal speaker and the cassette motor are disabled. When this bit is low (0), then the internal speaker is active. The cassette motor becomes active when both bits 4 and 3 are set low (0). If you wish to disable the speaker and have the cassette motor enabled at the same time, you should set bit 1 to 0, thus disabling the 8253's output.
6	5	Sound source at the MPX
0	0	8253-5 Timer 2
0	1	Cassette Audio Input
1	0	I/O Channel Audio In
1	1	SN76496N Complex Sound Chip
7		Reserved for future use.

In the program example listed here, we set this port location up in line 200. Line 200 outputs decimal 124 to port 97. If you convert decimal 124 to binary form, you will find that bits 2, 3, 4, 5, and 6 are on (set to 1), and that bits 0, 1, and 7 are off (set to 0). If you check this with the table above, you will find that the 8253 timer is off, the internal speaker is disabled, and the SN76496N sound chip has been selected. This is equivalent to using the following BASIC commands in Cartridge BASIC:

Beep Off: Sound On

When outputting data to this port, use caution, and be sure that the gate array bit (bit 2) is set according to the current screen mode. If this bit is not set correctly, the result may be a screen full of garbage, or possibly worse. If this happens to you, halt the program and type SCREEN 0. This will usually get you out of trouble. If not, you may need to re-boot your system. If you've been making changes though, always save your program before attempting the SCREEN 0 command.

So far, we've shown you how to direct the sound from the complex sound chip through the multiplexer, and out to the system's audio output. Now we will directly control the sound chip itself to generate some interesting sound effects.

With direct control over the sound chip you can create sound effects that just aren't possible with the SOUND or PLAY commands. However, when you use a SOUND or PLAY command, the computer must call an assembly language routine to interpret the command, gather and store the variables being passed, and call a subroutine which actually executes the command. Certain restrictions are built into these commands since there isn't enough memory in the entire world to allow them unlimited flexibility. These commands must execute within certain rules, which must be checked. All of this takes time, but it does make programming easier.

With the OUT command, there are no limit checks to see if the output is within a certain range, and there are no data conversions from an ASCII character representing a note to the special code required for the note to be produced. All of this has to be done by the programmer, making the programmer's task a little harder, but the end result is worth it.

Port addresses 192 and 193 access the sound chip's registers, which control the chip's output. The following table lists the values that control the complex sound chip:

PORT ADDRESS: 192 193

1st byte 192	2nd byte 193	VOICE #	ACTION
128-143	0-63	1	FREQUENCY
144-159		1	VOLUME
160-175	0-63	2	FREQUENCY
176-191		2	VOLUME
192-207	0-63	3	FREQUENCY
208-223		3	VOLUME
224-231		NOISE	SOURCE
240-255		NOISE	VOLUME

The following is a breakdown of the bit functions for port address 192:

PORT 192

BIT #	FUNCTION
0	This bit must always be set to 1.
1,2,3	Register addresses of the 76496 sound chip. Selects the function being accessed.
0 0 0	Tone 1 frequency
0 0 1	Tone 1 volume
0 1 0	Tone 2 frequency
0 1 1	Tone 2 volume
1 0 0	Tone 3 frequency
1 0 1	Tone 3 volume
1 1 0	Noise source
1 1 1	Noise volume
4-7	These bits are used differently depending on the register selected with bits 1 through 3. When you set the frequency for one of the three tones, these four bits act as the four low-order bits used to select the frequency. Six bits from the byte sent to port address 193 contain the high-order bits to complete the value of the frequency. When used as the volume control for any of the three tones or the noise, these four bits directly represent the volume range of 0 to 15. When setting volume it is not necessary to output anything to port address 193. When setting the source for noise, the four bits are used slightly differently:

BIT	FUNCTION
4	Not used
5	0 = Periodic, 1 = White
6, 7	
0	N/512
0	N/1024
1	N/2048
1	From tone #3 frequency

When you set up a frequency, the second output goes to port 193. (Actually, any port address from 192 to 199 can be used for the second byte sent. When the computer reads the first byte as a frequency it waits and reads the next byte as the second part of the frequency.) This second output contains the six higher order bits of the frequency value.

PORT 193

BIT #	FUNCTION
0	This bit must always be set to zero (0).
1	Bit 1 is not used. It doesn't matter what the bit is set to.
2-7	These bits are the high-order part of the whole frequency value. There are 10 bits which make up the frequency value—six from this byte and four from the first byte.

The frequency value is set up with the following bits:

FREQ. BIT	LOCATION
F0	Port 193, bit 2
F1	Port 193, bit 3
F2	Port 193, bit 4
F3	Port 193, bit 5
F4	Port 193, bit 6
F5	Port 193, bit 7
F6	Port 192, bit 4
F7	Port 192, bit 5
F8	Port 192, bit 6
F9	Port 192, bit 7

The Program

The program below uses the OUT command to set up the MPX to accept sound from the SN76496N complex sound generator and then creates some interesting sound effects. The loop between lines 250 and 320 controls the first siren type of sound. Each of the three voices is accessed in lines 260, 280, and 300. When the program outputs to port 192, only the bits which access the registers are set. The low-order bits for the frequency were not needed for the ramping effect on the routine. Now that an output to port 192 has

Continued on p. 148

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Disk Drives

Starting at **$~~169.00~~!** NEW LOW PRICE

Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor

40 or 80 Tracks — Single or Dual Head — New 3½'' Drivette™

Our Disk Drives are Capable of Single **and** Dual Density Operation

The **NEWEST** Technology Capable of Operating on Most Popular Computers

Drive a Hard Bargain!!™ For your TRS/80, Color Computer, IBM, Apple, Franklin

5 M.B.-20 M.B. Complete Systems.....from $999.95

Diskette Breakthrough — 10 Pack in Library Case — $~~18.95~~ NEW LOW PRICE

SAVE!! PLEASE CALL FOR OUR MOST CURRENT PRICE REDUCTIONS.

TOLL FREE ORDERING
1-800-343-8841

GENERAL AND TECHNICAL
1-617-872-9090

Disk Drives (0123) TRS/80-IBM-Apple - TI Franklin-Max/80-LNW .....◄
Model I/III/IV Upgrade (Disk Drives - Memory).....◄
Printers — Daisywheel/Dot Matrix .....◄
Percom Double Density Controller (Model I) .....◄
Color Computer Printer Interfaces .....◄
Disk Drive Operating Systems .....◄
Repair Services Now Offered — FAST Turn-a-Round.....◄
Apple/Franklin Compatible Add-On Drives with Case & Cable.....◄
Diskettes in Library Cases .....◄

CALL TOLL FREE FOR NEW PRICES

DISK DRIVE CASES AND POWER SUPPLIES ..... starting at $59.95
Printer **Buffers** 8K to 512K ..... starting at $143.95
Holmes Model I/III Speed-up Mod ..... starting at $90.00
Cables — Printer/Disk Drive ..... starting at $23.00

Warranty on Disk Drives — 6 Months to 1 Year

SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090

Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

DEALER INQUIRIES INVITED.

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.

CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular **Tel. (514) 845-1534**
Canadian Toll Free 800-361-5155

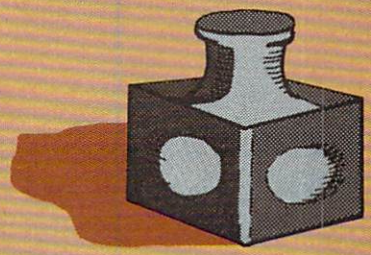
Service! Service!

All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. School and D&B corporate P.O.s accepted.

TRS/80 Registered Trademark Tandy Corp. IBM-PC Registered IBM Corp. Apple Registered Trademark Apple Computer Corp.
Franklin Registered Trademark Franklin Corp. Max/80 Registered Trademark Lobo Int.

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

Let's Get Graphic



Reviews of *Graphics Designer* and *Graphics Code Generator*

by Peter Bloch

HCM Staff

I can recall my excitement the first time I got my hands on a TI-99/4A. With one thought in mind, I immediately started to pore over the manual. Finally I found it: the CALL CHAR command. "Now," I thought, "I can design custom high-resolution graphics." After a few attempts, I discovered the correct syntax for the command. Then it appeared. My first custom character jumped up before my expectant eyes. . . "Is that what it's supposed to look like?" I wondered. Perhaps the computer made a mistake. What I'd produced on the screen was quite different from what I'd anticipated. After some code checking, I was forced to conclude that the fault did not lie with the computer. I realized that full use of the 99/4A's graphics capabilities would require deeper understanding.

Fortunately, there are several tools—in the form of programs—available for realizing the burgeoning programmer's visions. The format for these programs usually consists of an enlarged plotting area, a sample of the actual character, and the hexadecimal codes for that character. If this is all you need to carry out your graphics designs, you can get by with the *Character Definition* program in the *TI User's Reference Guide* (pages III-26 and III-27). But if you want to work with different colors or sprites, or if you'd like more than one character on the screen at one time, you may be interested in either *Graphics Designer* from Scott Compware or *Graphics Code Generator* from Software Carousel. These programs not only compute the hexadecimal codes for custom characters, but also allow almost instant editing of these characters. With the aid of these tools, the monitor and keyboard/joysticks become, in effect, your canvas and paintbrush for graphics design.

Using *Graphics Designer*

Graphics Designer from Scott Compware loads from cassette or diskette and runs under Extended BASIC. You use the program by "painting" in sections of an 8x8 Design Grid. This grid represents the standard 8x8 character definition block enlarged 64 times. You paint an area by pressing Y when the cross-shaped cursor is in the correct position and move the cursor using [FCTN] and the arrow keys. If you are familiar with hexadecimal codes, you can enter a hex digit to selectively deal with four areas at a time. Immediately to the right of the Design Grid, you'll find the hexadecimal

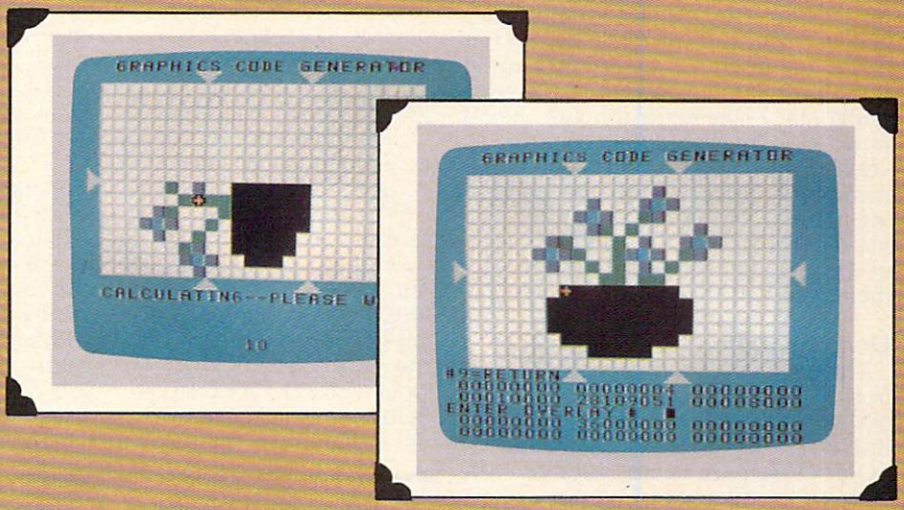
Continued on next page

Name:	Graphics Designer
Program Type:	Graphics Utility
Machine:	TI-99/4A
Distributor:	Scott Compware 5710 Lee Highway 18 Chattanooga, TN 37421
Price:	$16.95, diskette $14.95, cassette

	poor	fair	good	excellent
Performance			■	
Ease of Use				■
Documentation			■	

Name:	Graphics Code Generator
Program Type:	Graphics Utility
Machine:	TI-99/4A
Distributor:	Software Carousel P.O. Box 55561 Valencia, CA 01355
Price:	$19.95, diskette $16.95 cassette
System Requirements:	Joysticks

	poor	fair	good	excellent
Performance			■	
Ease of Use				■
Documentation			■	



code for each row of the Design Grid. Below these codes, you'll see a sample of your character scaled down to its correct size. Beneath the sample is a Plot Grid. Here you can plot your characters to see how several of them look together. You can also change the foreground and background colors of your characters on the Plot Grid. And if you're working with sprites, you can take advantage of the Magnify option by pressing [CTRL][S].

Any previously defined character can be recalled for editing via the Insert command. This command moves the selected character back onto the Design Grid. You edit with Y, N, [FCTN] and the arrow keys. Since you are changing the character definition when you edit, all the characters (even magnified sprites) with that character code on the Plot Grid will also change.

"These programs not only compute the hexadecimal codes for custom characters, but also allow almost instant editing of these characters."

Two of the most useful features of this program are the Flip and Rotate commands. With the Rotate command you can rotate a character 90 degrees clockwise. As the command is completed, you are given the new hexadecimal codes for the rotated figure. At the same time, the program will rotate the figure—in the sample area as well as on the Plot Grid—to conform to your editing changes. If you rotate twice, your figure will of course be pointed in the opposite direction, and unless it is a horizontally symmetrical figure, it will now be upside-down. To avoid this inversion, you can invoke the Flip command to flip a figure so that it is pointing in the opposite direction but is not upside-down. This is very useful to those of us who like to make sprites change their shapes as they change directions.

Documentation

The uncluttered format of the instruction manual for *Graphics Designer* makes it easy to use. Instructions are illustrated. At the end of the manual a "Quick Reference Guide" reviews each instruction. After spending just a short time with the documentation, you can start making your own designs.

Ease of Use

Graphics Designer's commands are easy to find and use. If you happen to forget a command, you simply go to the menu displayed at the bottom of the screen. You can quickly correct errors made while entering command variables by typing [FCTN] [9]. The cursor will then return to its previous position, where you'll be able to enter a new response. When I responded to command prompts with an inappropriate answer, the program's error-handling routines accurately identified the error and listed the range of correct values.

With its quick editing, *Graphics Designer* fulfills many of the graphics programmer's needs. If hexadecimal codes are keeping you from using your TI's graphics capabilities, this program will provide the assistance you need to get you designing your own characters. One feature of *Graphics Designer* I would like to see changed, however, is the size of the Plot Grid. Why couldn't the PLOT command place the characters or sprites exactly where they will be on an entire screen reserved for this option? Then we could really see how our designs will appear in our programs. For that matter, how about letting us see motion?

Graphics Code Generator

Software Carousel's *Graphics Code Generator* presents another approach to the difficulties of graphics. This Extended BASIC program, available on cassette and diskette, does a lot more than its name implies. In addition to providing hexadecimal codes, it allows you to design up to six characters at a time. This is possible because *Graphics Code Generator*'s design area is six times larger than that of either the program in the *TI Users Reference Guide* or the *Graphics Designer*.

Joysticks control cursor movement on this 24x16 grid. To paint in an area, press the fire button. Central to this program's operation is the idea of *overlays*. After filling the design area with up to six characters, you can select another overlay which can be placed over the original. Since all but the first overlay are considered to be sprites, this process is limited by the number of sprites TI Extended BASIC will allow on one line. Overlay colors are selected with the Color Change command.

Using the Display Image command will allow you to see any overlay in its true size. Unfortunately, you can view them only one at a time. Therefore, it is not possible to see the composite picture. If you confine yourself to the four left-hand areas, it is possible to Rotate the overlay 90 degrees counterclockwise. Or, if you prefer, you can Relocate your image anywhere within the design area. After executing either of these instructions, you can then obtain the hexadecimal values of the shifted image through use of the Code Image command.

"Using the Display Image command will allow you to see any overlay in its true size."

If you have a printer, you can get a listing of the overlays, their positions on the design area, color codes, and hexadecimal codes. Since each printed report can be individually labeled, you'll be able to identify a particular file. If you have a cassette recorder or disk drive, you can SAVE your characters and sprites. Later, these figures can be loaded back into *Graphics Code Generator* for review or editing.

Documentation

Graphics Code Generator comes with a four-page instruction manual. Anyone unfamiliar with cassette or disk operation will find the loading instructions helpful. The manual presents enough essential information on program operation to get you started working with graphics. The commands are catalogued in the same order in which they appear in the Action Menu. Command explanations are detailed and lucid. Wherever necessary, [ALPHA LOCK] prompts are included: up for joystick operation and down for printed reports. Also included in the documentation is a short program that will automatically extract your character's hexadecimal codes from disk and input this information into your program.

Human Engineering

Graphics Code Generator can be used by anyone who has had even moderate exposure to computers and graphics. The program is menu-driven. Where it is inconvenient to display the main Action Menu, a smaller option list directs you to either a command or the main menu. The program lets you save character definitions so you'll be able to build a library of custom characters. Most programmers will probably enjoy "painting" on *Graphics Code Generator*'s large design area. Because it is possible to work with four sprites at once, the benefits of this oversized design area will be obvious to those who use the Magnify 3 or Magnify 4 option.

"If you have a printer, you can get a listing of the overlays, their positions on the design area, color codes, and hexadecimal codes."

The *Graphics Code Generator* program does have one flaw. Its error-handling routines could be improved. Entering the wrong number can cause the program to terminate. If this happens, your painting will be lost.

Despite this pitfall, *Graphics Code Generator* has plenty of features to assist anyone struggling with graphics creation. If you need more information on program operation, or if you'd like to see a unique demonstration of *Graphics Code Generator*, a very interesting demonstration tape is available from Software Carousel. HCM

CP/M®

PROCESSOR FOR THE TEXAS INSTRUMENTS 99/4A

NOW YOUR TI-99/4A REALLY MEANS BUSINESS!

Morning Star Software introduces the CP/M Processor for the TI-99/4A. Now the advantages of the most accepted business programs are yours. With the CP/M Processor, your TI-99/4A is ready to enter the business world.

The Morning Star Software CP/M Processor fits into your peripheral expansion system like any expansion card. It's as simple as that. Once the CP/M Processor is activated, you have all the advantages of two processors: The CP/M Processor handles all the program computations, while the TI Processor channels the information to the screen, disks and printers. This gives you the flexibility you've always wanted.

CP/M is the industry standard business format for computers today. CP/M offers you the advantage of working with programs that were designed to meet the needs of the professional business world. Powerful commands, time saving functions and proven methods are the result.

The CP/M Processor by Morning Star is built to take advantage of both the TI computer and CP/M technology. The CP/M Processor contains an 8085 CPU which, with the 16-bit TI-9900 CPU, translates to tremendous processing power. Your TI-99/4A combined with the CP/M Processor becomes a true business tool at a fraction of the expense of a new computer system.

- *Needs no modifications to the TI-99/4A system.*
- *64K RAM, 8K ROM.*
- *Makes additional use of the TI 32K memory expansion.*
- *Supplemental installation and usage manual included.*
- *Contains sufficient memory for virtually all programs.*
- *Compatible with TI-99/4A storage diskettes.*
- *Provides 60K transient program area.*
- *Fully burned-in and test cycled to ensure reliability.*
- *Fully shielded and protective metal enclosure.*
- *Ultra-reliable gold connector assembly.*
- *No chip sockets for better reliability.*
- *90-day warranty.*
- *CP/M-80 2.2 disk-based operating system included.*
- *Digital Research™ CP/M user's manual included.*
- *Access to thousands of commercially available software packages.*

THE MORNING STAR CP/M PROCESSOR

$595

TO ORDER: Check, Money Order, Master Card, VISA, and C.O.D. orders accepted. Add $5.00 shipping and handling. Dealer inquiries welcomed.

COMPATIBLE CP/M® SOFTWARE

The Morning Star CP/M processor is the only way the TI 99/4A will run CP/M based programs. The programs have been selected and re-formatted to run with the Morning Star CP/M processor. As Morning Star reviews CP/M packages, we will continue to expand our software selection. The programs we offer are the products which demonstrate outstanding values. They have met our criteria of being extremely user friendly, comprehensive in content, plus make the best use of the CP/M card's capabilities. They also represent the tasks most needed to be covered by a business computer.

***BUSINESS MASTER® PLUS*—Improved**
Now enjoy this comprehensive business package without having to buy CBasic Interpreter! Designed to meet the needs of leading businesses. This package includes diskettes for General Ledger, Payroll, Fixed Assets Accounting, Accounts Receivable, Accounts Payable, Inventory, and Mailing List programs.
Regularly $495;
Introductory Price **$295**

SUPERWRITER™ by Sorcim
This is the word processing program you've been waiting for. Designed to be easily understood in ten minutes with straightforward language and screen prompts. Written specifically for the business person with word processing requirements. It provides all key features of a professional text editor. It incorporates form letters, report generation, list maintenance and even a spelling checker, the award winning SpellGuard™. Similar features cost as much as $900. SuperWriter™ only **$295**

PERSONAL PEARL™ Data-Base Manager
*Now easily create data systems for business, accounting, marketing, office management, education, and more. A complete user's manual, tutorial disk and screen menu guides even novices through the most sophisticated data base manager available. Unlimited files, forms and reports can be created. **Info World** reported "Personal Pearl offers you the strongest, easiest to use data-base manager I have seen at any price".* **$295**

THE RANDOM HOUSE PROOF READER
20,000 word Random House dictionary. Displays spelling errors in context and can make spelling suggestions. You edit the word. Corrections are automatically checked and substituted into the document. File new words into the dictionary with a keystroke. **$50**

CBASIC® INTERPRETER
The language of choice for many CP/M compatible software packages. This brings you a variety of established and time tested applications and allows you to write your own CP/M compatible programs. **$150**

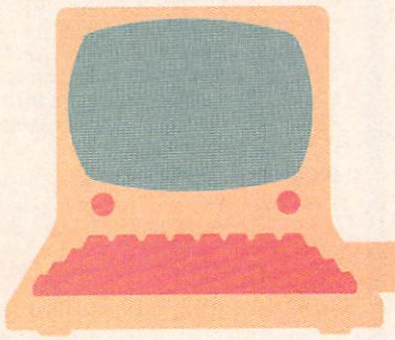
MORNING STAR SOFTWARE

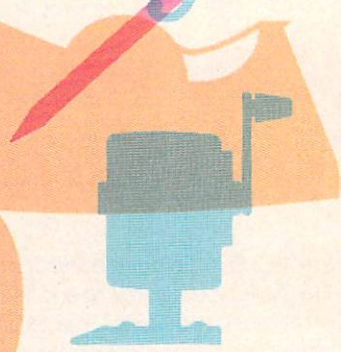
4325 S.W. 109th Ave., Beaverton, Oregon 97005. Toll-Free 1-800-824-2412, in Oregon 503/646-4695, TEX NET NO. TI3416

Peripheral Requirements: Peripheral Expansion System, disk, disk controller 32K RAM card.

The ELECTRONIC Home Secretary

by Malladi Subbaiah
and the HCM Staff





Note: The touch-tone phone relys on a *pair* of frequencies to decipher the number being pressed. These *two* frequencies must be mixed and placed on the telephone line *at the same time.* Because the Apple and the IBM PC with standard equipment are only capable of producing *one* tone with their built-in speakers, they aren't capable of producing the simultaneous tones necessary for dialing.

Now that you have a personal computer, you've probably been looking for ways to use it around the house. When writing software for home applications, you can create a *general* program that functions in a variety of household situations. The programs accompanying this article follow this design philosophy. You can create a personal phone and address directory, time events (such as elapsed connect time for long distance calls), and set up an inventory of household possessions for insurance and maintenance purposes. The TI-99/4A, Commodore 64, and IBM PCjr programs will even dial or redial any phone number in your directory.

Data Entry

When the program is first RUN, the screen options allow the user to select one of two primary program modes; either Phone Book or Inventory. Select the mode you want by pressing the corresponding number key. Both of these modes then display a menu of nine options for manipulating and using the data.

After you select your option by pressing a corresponding number key, this user friendly program will guide you, step by step, through the different utilities. Each option will perform a function on one of a maximum of sixty records contained in either a phone book or inventory file. Each record is further divided into five fields, with headings applicable to the type of file.

Initially you will want to use the Add option to create a data base for your files. Once this is done, you can save the file on disk (all systems) or on tape (C64, TI, and IBM).

Sort Routine

An efficient sort subroutine is presented in the program and is called, automatically, after there is an addition of, or change made in, a file. You will be asked to select one of the five fields, and a sort will take place before the program returns to the main menu. Note that if you are in Inventory mode and sort by the value field, the sort will be inaccurate. This is because the sort is done on the ASCII code of the characters in the field and not by the actual value.

Data Deletion and Alteration

This subroutine updates any existing data set. You can access any particular entry by its serial number (relative sorted position within the file),name field, or segment of its name. A search routine retrieves the data set with the specified name, or the next higher one if the name match is not exact. After altering the record, you have the option of searching the list for a different entry or ending the editing session. After any alteration of the file, before the directory can be displayed, the data set is always resorted.

Use Data Option

The function of this option is dependent on the primary mode selected. If in Phone Book mode, you will first be asked to select a record to work on. (This record is selected in the same way as the Change Data option.) Once the record you want is displayed, press a key, and the computer will generate the dual tones necessary to place a phone call. You may also redial or select the stopwatch function at this time. If you are in the Inventory mode, this option will sum up the value fields of all the records in the file and place a sum total on the screen.

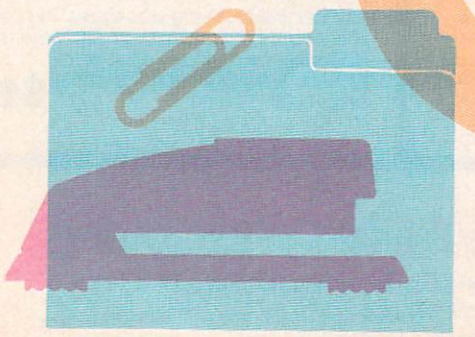
Display of the Directory

The program allows you to display the data directory in one of two formats. The first format provides a concise, quick-reference listing of the complete directory. This includes name and phone number for the Phone Book option, and item and cost for the Inventory option.

In the second format, you can display all the data contained in any single record. Access to individual records is either by serial number in the directory or by a string search (discussed in a previous section).

Additionally, each program contains a printer option that allows you to obtain a hard copy listing of the entire file. This option may have to be modified by each user to reflect the exact hardware configuration of the individual system.

Continued on p. 134



WOODINVILLE COMPUTER CENTER INC.

SPECIALIZING IN

BUSINESS & EDUCATIONAL SOFTWARE FOR APPLE, FRANKLIN & IBM COMPUTERS

OVER 2000 PRODUCTS IN STOCK READY FOR IMMEDIATE SHIPMENT

APPLE/FRANKLIN
SUPER SPECIAL
DISTAR THIN LINE DISK DRIVE $189.85
2 OR MORE ONLY $184.85

SUPER SPECIAL
DISKETTES
SS/SD $17.77 Box of 10
DS/DD $27.77 Box of 10

PRODUCTS FOR APPLE

SOFTWARE PUBLISHING CORP.
PFS: WRITE IIe ..... Reg. $125 **$99.85**

HARCOURT-BRAGE-JANONOVICH
COMPUTER SAT ..... Reg. $79.95 **$59.85**

KOALA TECHNOLOGIES CORP.
KOALAPADS ..... Reg. $124.95 **$99.85**

SPINNAKER SOFTWARE
KIDWRITER ..... Reg. $34.95 **$27.85**

MONOGRAM
DOLLARS & SENSE ..... Reg. $100 **$77.85**

TERRAPIN
TERRAPIN LOGO ..... Reg. $149.95 **$107.85**

SUBLOGIC CORP.
FLIGHT SIMULATOR II . Reg. $49.95 **$41.85**

DESIGNWARE
SPELLICOPTER ..... Reg. $39.95 **$31.85**

PRODUCTS FOR IBM

LOTUS DEVELOPMENT CORP
1-2-3 ..... Reg. $495 **$329.85**

MICROSOFT
MICROSOFT WORD ... Reg. $375 **$266.85**

HERCULES
HERCULES GRAPHICS CARD . Reg. $499 **$377.85**

ASHTON-TATE
dBASE II ..... Reg. $700 **$449.85**

BARRON'S EDUCATIONAL BASE
BARRON'S COMP. SAT ... Reg. $89.95 **$68.85**

PEACH TREE SOFTWARE (GL/AR/AP)
SERIES 4 PEACHPAK .. Reg. $395 **$271.85**

KOALA TECHNOLOGIES CORP.
KOALAPAD TOUCH TABLET Reg. $150 **$122.85**

SOFTWARE TECHNOLOGY
MULTISOLVER ..... Reg. $295 **$209.85**

Order Phone:
(206) 481-2699 M-F 9-6 PST

TECHNICAL ASSISTANCE & CUSTOMER SERVICE (206) 485-1707 M-F 10-2 PST

ADVERTISED SPECIALS ALSO AVAILABLE AT OUR NEW RETAIL OUTLET
13300 NE 175th
WOODINVILLE, WA

VISA

If you don't see it in our ad call for lower prices.

SERVING YOU SINCE 1978

***PLEASE!!! . . .* MINIMUM ORDER $50**

Be sure to ADD $2.50 shipping and handling for all softward orders. Shipping on hardware items — EXTRA. ADD additional $3.00 for Blue Label (Air). C.O.D.'s ADD $5.00. Oversees orders EXTRA. Washington residents add 7.8% sales tax. We accept MASTERCARD, VISA, and AMERICAN EXPRESS. (American Express requires additional service charge).

• • TERMS AND CONDITIONS • •

ALL SALES ARE FINAL! Merchandise is new and covered by factory warranties. Returned merchandise will be subject to a restocking charge. Defective or damaged merchandise will be repaired or replaced if returned to us within 30 days from the invoice date. User caused damage is excluded. For RETURN AUTHORIZATION OR TECHNICAL ASSISTANCE please call CUSTOMER SERVICE at (206) 485-1707 between 10:00 AM & 2:00 PM Monday through Friday PST. We will not accept your returned item without a valid return authorization number. Your shipping expenses on merchandise you have returned to use is not refundable.

Name _____

Street _____

City _____ State & Zip _____

Card # _____ Exp. Date _____

Signature

P.O. Box 1028, WOODINVILLE, WA 98072

SEND $2.00 for full color educational products software catalog. ***We accept purchase orders from schools & major corporations.***

Secretary . . .

Computerized Telephone Dialing

Now let's look at touch-tone dialing. Since the telephone company prohibits direct connections to the phone line of any user equipment not approved by the FCC, the method we will use involves simple proximity: Placing the microphone from the phone handset in front of the monitor speaker dials the phone without any direct connection to the phone lines.

The touch-tone system of telephone dialing operates by sending a specific pair of audio frequency tones over the voice channel of the phone line for each digit. The switching circuits at the telephone facility decode the tones and actuate the appropriate circuits to make the connection. Each tone pair consists of a low-frequency group (ranging from 697 to 974 Hz) having four members and a high-frequency group (1209-1477 Hz) having three members (shown in Figure 1). So to dial the number 5, for instance, we have to send the audio tones at 770Hz (low frequency) and 1336 Hz (high frequency) simultaneously for a long enough time for the tone pair to be recognized by the switching circuits. There should also be a time gap between tone pairs sufficient for each pair to be registered individually as a digit of the phone number. Although theoretically, a 40-millisecond signal duration followed by a 40-millisecond silence should be adequate, a 150–200 millisecond signal duration and a gap of about 100–150 milliseconds is actually required for reliable operation with this system.

With the Commodore 64 this phone dialing routine works perfectly and requires no external hardware. But for the IBM PCjr and TI-99/4A computers we must anticipate some problems. Each of these computers can generate the dual tones of Figure 1, but if we examine the monitor's output on an oscilloscope, we can observe that the so called "pure tone" from the computer is, in fact, a *square wave* and not a *sine wave*. By Fourier analysis, the square wave can be decomposed into its constituent sine waves. (Interested readers can refer to any elementary book of calculus for the details of this analysis.)

For example, if we wanted to produce a frequency of 500 Hz, the output from the computer would be a square wave of 500 Hz, which in reality would be a combination of sine waves at 500 Hz, 1500 Hz, 2500 Hz, and so on. This can pose a problem when we try to send either of the first two members of the four-member low-frequency group (i.e., 698 Hz and 770 hz). The third constituent harmonic of these low frequencies, 2091 Hz and 2310 Hz, respectively, are also recognized by the switching circuits, resulting in the rejection of the signal. The third harmonics of 852 Hz and 941 Hz seem to be outside the frequency response of the switching circuits and pose no problem.

There are several ways we can overcome this problem when dialing the digits 1 thru 6. One simple and inexpensive way is to use a passive low-pass filter with a cut-off frequency of about 1.5KHz in the audio line to the monitor to attenuate the higher frequencies. Figure 2 shows a block diagram for the installation. The circuit for the filter (which can be built for under five dollars) is shown in Figure 3.

Loading a Previously Created File

To load a previously stored data file, we select the Load Data option and follow the screen displays to operate the cassette player or disk. When loaded, the name of the data file, its size, and the date of the previous revision will be displayed; the program will then return you to the master selection list.

Low Frequency Group \ High Frequency Group	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Figure 1. Basic Frequencies for the Two-Tone System of Telephone Dialing

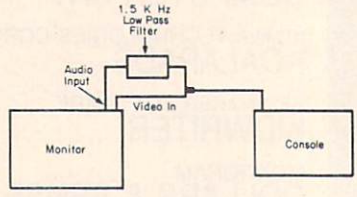


Figure 2. Schematic Layout of Filter Location

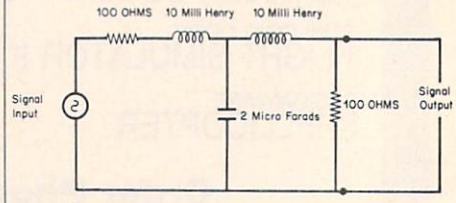


Figure 3. Circuit Diagram of the Filter

Note: On many touch-tone phone systems this filter will not be needed for correct dialing. We suggest you first try without it—Ed.

The original version of *Electronic Home Secretary* was created for the TI-99/4A. The program presented here was designed to be as versatile as possible for those who have only the 99/4A computer and a tape recorder. It is also capable of handling systems that have disk drives. This version is written entirely in TI BASIC with the only required peripheral being the tape or disk storage device to save the file. The program also provides hard copy reports for those who have printers, and the TI Extended BASIC cartridge can be used to speed the efficiency of the program.

A unique feature of this program is its ability to produce the tones necessary to dial a touch-tone telephone from the phone numbers stored in the file. To do this, it is necessary to produce two frequencies at the same time—something the 99/4A can do quite nicely. The problem is that the frequencies need to be filtered in order to be accepted through the phone. The Commodore 64 has this filtering ability built into it, so it doesn't require any external filtering. The TI-99/4A, however, does not have that capability and requires an external filter. If you are using a television to produce the sounds, you may be able to use the tone adjustment as the filter. Try this experiment with different tone settings until you find one that works. See the instructional section on building the filter if you can't get your TV or monitor to dial the phone for you.

The TI Home Computer doesn't have a built-in clock, but we can simulate a reasonably accurate one. This is done with the CALL SOUND statement. The CALL SOUND statement lets you specify the duration of the tone in thousandths of a second. Lines 3320 and 3330 set up a loop which will execute only once every second. The CALL SOUND in line 3330 will continue while the statements from line 3340 to 3510 are executing. This section of code updates the clock by adding one second and then displaying it on the screen. The routine then branches back to line 3320, which is another CALL SOUND statement. Because the durations are positive numbers, the SOUND statement in line 3320 can't start until the sound statement in line 3330 has finished. This process takes one second.

There is one limitation to using this method. The code between the CALL SOUND statements must take less than one second to execute. If the code between the CALL SOUND statements takes more than one second, the first SOUND statement will finish before the second SOUND statement is reached, defeating the purpose. The timing may vary slightly from machine to machine. To find out how accurate your machine is, run the Stop Watch option after the computer dials a number and time it against an accurate watch over a sufficient period of time. Several minutes will usually do the trick. To adjust the clock's speed, change the duration value in line 3330 from 900 to another value. A smaller value will speed the clock up, while a larger value will slow the clock down.

Continued on p. 143

The major difference in the Apple version of *Electronic Home Secretary* stems from its lack of a multi-tone sound system. The TI-99/4A, PCjr, and C-64 are all able to approximate the dual tones necessary to activate touch-tone dialing, while the Apple is unable to easily produce two tones simultaneously without added hardware such as a sound board for one of the expansion slots. Consequently, touch-tone dialing is not available for the Apple version of the program. We did, however, include the timer function as part of the USE THE DATA option with the electronic phone book. To calibrate your timer, vary the maximum value in the FOR-NEXT loop in line 1790.

The only other difference occurs in the way disk files are managed with the Apple. *Electronic Home Secretary* is actually two programs in one: a phone book and an inventory system. Each of these options allows you to save your records to disk. A problem that might arise is saving your inventory records over your phone book records or vice versa. To eliminate this possibility, the program adds an extension to the file name composed of a period and three letters (e.g., .PHO). Let's say you are about to save your phone book file, and when the program prompts you for the file name, you type in LIST1 as your file name. The program automatically ends the file name with .PHO. That way if your inventory file were also named LIST1, it would not be overwritten because it would have the extension .INV.

When you CATALOG your disk you will see these extensions on the filenames (i.e., LIST1.PHO). However, you should not add them when prompted because they are added by the program. If you did add them, your file name would not be LIST1.PHO, but LIST.PHO.PHO—there isn't anything wrong with that file name, but it could be a bit confusing.

If you attempt to load .PHO file while doing INVENTORY (or an .INV file when accessing the PHONE BOOK) you will get the error message EITHER FILE NOT FOUND OR WRONG TYPE,and you will be given the option of CATALOGing your disk to see whether you have simply misspelled a filename or tried to load the wrong type.

Continued on p. 138

The *Electronic Home Secretary* contains an interesting subroutine that will place phone calls for you from numbers you place in its directory. Actually, it doesn't place them for you, but merely relieves your fingers from pressing the buttons. To place a phone call, hold the phone receiver up to the television speaker, press a key, and your Commodore will produce the dual tones required for each number in the sequence. This article outlines the hardware filtering required to produce the exact tones. All of the hardware needed to implement this "phone dialer filter" is already contained inside the Commodore 64 in the form of the versatile Sound Interface Device (SID). This integrated circuit ultimately produces all the sounds initiated by our computer programs when they POKE certain values into its control

Continued on next page

registers. The control registers we will primarily be concerned with here are registers 21 (54293), 22 (54294), 23 (54295), and 24 (54296).

Registers 21 and 22 are used together to form a 13-bit value that sets the cutoff frequency of the filter. This value may be derived from the following BASIC statement lines:

```
10 RF = 2000:  X = (RF-30)/8:
   RH = INT(X/8):RL = X-8*RH
```

RF is set equal to the cutoff frequency. RH is POKEd into register 21 and RL is POKEd into register 22. X is a temporary variable.

Bits 0 through 2 of register 23 control which voice outputs are sent through the filter. If bit 0 is set to 1, then voice 1 will be sent through the filter. Bit 1 corresponds to voice 2, and bit 2 corresponds to voice 3.

Register 24, bits 4 through 6, selects the filter mode. If bit 4 is set to one, then the low-pass filter is selected. If bit 5 is set to one, then the band-pass filter is turned on. And, finally, if bit 6 is set to one, then the high pass filter is selected. Filter modes may be selected simultaneously to produce a notch or (band reject) filter response.

Filtering

The hardware configuration that produces the phone dialing tones uses the low-pass filter mode. This is because the telephone switchboard is very selective about which frequencies it responds to. If the tone is off the standard frequency or strong harmonics are present, the tone becomes useless. The frequencies corresponding to the standard tones are contained in the DATA statement of line 230. These are read into the arrays P1 and P2 in line 200.

The possibility of strong harmonics—above 2000 cycles—is eliminated by selecting a filter mode for voices 1 and 2 and setting a cutoff frequency of 2000 cycles. This is done in lines 270 and 280 of the program. The type of filtering is selected in the sound subroutine (line 2410 through 2510) in line 2450. The value of 31 sets low-pass mode and maximum volume.

The filter is perhaps the most important element of the SID chip when interfacing to the non-computer world. It allows the generation or degeneration of specific frequencies in harmonically rich tones. We invite you to send us your ideas about this powerful but rarely used feature of the Commodore 64.

Continued on p. 141

Do you hear what I hear? A tone, a tone, and another tone with a chord as rich as if it were from a piano. You probably do hear it if you have Cartridge BASIC for your IBM PCjr. If you have only Cassette BASIC, then you may be able to hear only single tones. But don't be too distressed, for your lack of ability to generate multiple tones will last only as long as it takes you to read this article. By the way, even if you do have Cartridge BASIC, don't disregard this article; it offers a method for generating multiple tones, at frequencies other than those that directly correlate to standard note frequencies. This is indeed a new and exciting concept for any PCjr user.

The difference between Cassette BASIC and Cartridge BASIC, as far as this article is concerned, is that Cassette BASIC does not have the PLAY command. The PLAY command, like many other commands in Cartridge BASIC, takes advantage of the unique "home computer" architecture of the PCjr, and in this case the advantage is in the form of the versatile Texas Instruments sound chip (SN76496N).

> ***Note:*** The TI SN76496N sound generator's output can be directed only to the IBM connector used for TV or external-amplifier interfacing. So you must have one of these external devices connected to your system if you are to be able to use this versatile sound chip

This sound chip has the ability to produce 3 different tones simultaneously, with each having its own frequency, duration, and volume. In Cassette BASIC, however, there is no command that takes advantage of this capability; it is virtually the same language resident in the PC—a machine without the TI sound chip. And that is why Cassette BASIC on the PCjr doesn't have the PLAY command. Fortunately, there is a solution—the OUT command that is a part of Cassette BASIC. The OUT command will allow us to get right down on a hardware level and control this sound chip to the full extent of its capabilities.

The PCjr version of *Electronic Home Secretary* incorporates a sound subroutine that will do just that. If you like, you may extract it (lines 200 through 210 and Lines 2250 through 2400) and use it in your own programs.

Duration Variables

The duration variables are D1, D2, and D3, respectively. D1 controls the duration of tone 1 (voice 1), D2 controls tone 2 (voice 2), and D3 controls tone 3 (voice 3). The approximate time of a duration unit is 28 milliseconds, and total duration can be calculated using the following formula:

Time of duration (in seconds) equals the value of the duration variable times .028 seconds.

or,

Duration = DX*.028.

If you take a look at lines 2340 through 2380 in the sound subroutine, you can see how the duration variables are used.

> ***Note:*** To get this program to run on the IBM PC without its dialing capability, it is necessary to change the PCjr version in the following way:
>
> **Line 2250 REM**
> **Line 2400 RETURN**

Above these lines the tones are turned on; once this is done, a loop is entered that will turn off the individual tones as a corresponding variable reaches a terminal count (less than zero). When all of the duration variables reach their terminal counts, the loop is exited and the duration variables are reinitialized to one. So duration variables must be initialized each time the sound subroutine is entered. If a particular voice is not needed, the duration variable need not be initialized because the variable will already be at its terminal count and will not affect the loop duration. It takes approximately 28 milliseconds to progress through the loop one time.

Frequency Variables

F1, F2, and F3 are the frequency variables and are set equal to the frequency (in Hertz) of the tone you want to generate before you enter the sound subroutine. Lines 200 and 210 are used both to initialize the frequency variables and to define the function IFREQ that will be used to derive values that will be written into the sound chip's control registers to generate our tones. If you are not planning on using one of the voices, then that corresponding frequency variable must be initialized to a value greater than zero or else a "division by zero" error will occur in the sound subroutine. In *Electronic Home Secretary* the third voice is never used, so voice 3 is set equal to an arbitrary value of 800 in line 210.

Volume Variables

The volume variables—VL1, VL2, and VL3—are used not only to set the volume, but also to turn the voices on or off. VL1 corresponds to voice 1 as well as to D1 and F1; VL2 corresponds to voice 2, D2, and F2; and VL3 corresponds to voice 3, D3, and F3.

In line 210, VL3 is set equal to 15. This value, in effect, sets the volume level of voice 3 at the lowest possible level, and to our ears this sound is off. We initialize VL3 to this value because voice 3 will not be used. This turns its level all the way down. You can see just how important the volume variables are when you realize that the sound subroutine always turns all voices on. It is the function of the VL variables, then, to turn the voice on and to select the volume level. If a volume variable is set equal to zero, the corresponding voice will have the maximum volume level.

Using the Sound Subroutine

In *Electronic Home Secretary* the sound subroutine is entered from line 1960 or line 1980. The duration and volume variables are derived in line 1950. For telephone tone dialing, the duration and volume remain constant while the frequency varies with the number being dialed. As you can see, all pertinent control variables are initialized to their proper values each time, before entering the sound subroutine.

Conclusion

Here it is. Take it away and see what you can do with it, and if you need a more detailed explanation of the OUT command and how it is used, please refer to "Jr. Sounds Off" in this issue.

HCM

ELECTRONIC HOME SECRETARY (PCjr)
Explanation of the Program

Line Nos.	
100-170	Header.
180-230	Variable initialization.
240	Read frequencies correlating to dial tones.
250-290	Read in record field identifiers.
300-400	Display first screen and receive input.
410-550	Key input for data set up.
560-680	Display main menu screen and accept input.
690-700	Add data subroutine.
710-880	Data alteration subroutine.
890-950	Display name list subroutine.
950-1020	Single item listing.
1030-1130	Routine to go forward or backward in an array.
1140-1230	Search routine for single item listing.
1240-1450	Load data subroutine.
1460-1610	Save data subroutine.
1620-1780	Sort routine.
1790-1840	Initialize utility routines.
1850-2070	Routine to initialize variables for sound subroutine.
2080-2120	Stopwatch subroutine.
2130-2180	Print data subroutine.
2190-2240	End of program routine.
2250-2400	Sound subroutine.
2410-2440	Get input from keyboard.
2450	Partial clearing of screen routine.

IBM PCjr

```
100 REM ************************************
110 REM *  ELECTRONIC HOME SECRETARY        *
120 REM ************************************
130 REM BY MALLADI SUBBAIAH AND THE HCM
    STAFF
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM PCjr CARTRIDGE BASIC
170 REM PCjr CASSETTE BASIC
180 DIM A$(5,60),O$(5),C$(6),P1(3),P2(3
    ),B$(2,5)
190 SCREEN 0:WIDTH 40:CLS
200 DEF FNIFREQ(F)=INT((35790)/(.32*F)
    )
210 VL3=15
220 F3=800
230 OP=1
240 READ P1(1),P1(2),P1(3),P2(1),P2(2),
    P2(3)
250 FOR I=1 TO 5:READ B$(1,I):NEXT
260 FOR I=1 TO 5:READ B$(2,I):NEXT
270 DATA 697,770,852,1210,1340,1481
280 DATA "NAME:","PHONE:","ADDRESS:","C
    ITY:","STATE&ZIP:"
290 DATA "ITEM:","VALUE:","LOCATION:","
    DATE PURCHASED:","SER# OR ID#:"
300 CLS
310 LOCATE 2,8:PRINT "**********
    **********"
320 PRINT TAB(8) "*** THE HOME SECRETARY
    ***"
330 PRINT TAB(8) "**********
    **********"
340 REM NEW SET UP
350 LOCATE 18,8:PRINT"PRESS":LOCATE 20,
    8:PRINT TAB(8) "1. PHONE BOOK":PRIN
    T TAB(8) "2. INVENTORY"
360 GOSUB 2420
370 IF K=49 THEN OP=1:GOTO 400
380 IF K=50 THEN OP=2:GOTO 400
390 PRINT TAB(8) "RE-ENTER 1 OR 2":GOTO
    360
400 N=0:GOTO 560
410 REM KEY INPUT FOR DATA SET UP
420 CLS:LOCATE 2:PRINT "ENTER":PRINT"
    :PRINT " TO EXIT":PRINT " R TO
    REENTER":G1=1:G2=2
430 I=N+1
440 IF I<=60 THEN 470
450 PRINT:PRINT "*** ARRAY FULL ***"
460 GOSUB 2410:RETURN
470 PRINT "
480 PRINT B$(OP,1);:A$(1,I)="":INPUT A$
    (1,I)
490 IF A$(1,I)="E" OR A$(1,I)="END" THE
    N 1630
500 IF A$(1,I)="" THEN 480
510 IF A$(1,I)<>"R" THEN 530
520 I=I-1:N=I-1:PRINT " ***REENTER LA
    ST SET***":GOTO 470
530 FOR Z=2 TO 5
540 PRINT B$(OP,Z);:A$(Z,I)="":INPUT A$
    (Z,I):NEXT Z
550 N=I:GOTO 420
560 SC=4:CLS:LOCATE 2:PRINT "
    HOME SECRETARY **"
570 IF OP=2 THEN 590
580 PRINT TAB(16) "PHONE BOOK":GOTO 600
590 PRINT TAB(16) "INVENTORY"
600 PRINT "**********
    **********"
610 PRINT:PRINT "PRESS":PRINT:PRINT "1.
    ADD DATA":PRINT"2. CHANGE DATA":P
    RINT "3. DISPLAY NAME LIST"
620 PRINT "4. DISPLAY ONE RECORD":PRINT
    "5. USE THE DATA":PRINT "6. SAVE D
    ATA FILE"
630 PRINT "7. LOAD DATA FILE":PRINT "8.
    LIST TO PRINTER":PRINT"9. END PROG
    RAM"
640 GOSUB 2420
650 IF K<49 OR K>57 THEN 560
660 CLS
670 ON (K-48) GOSUB 690,710,890,950,179
    0,1460,1240,2130,2190
680 GOTO 560
690 GOSUB 410 RETURN
700 RETURN
710 REM DATA ALTERATION
720 INPUT "WHICH ONE";M$:IF M$="" THEN
    M$="1"
```

Continued on next page

Secretary . . . from p. 137

IBM PCjr

```
730 PRINT:PRINT:PRINT:PRINT "ENTER":PRINT:PRI
    NT "  NEW DATA AT CURSOR":PRINT:PRIN
    T "  /D/ TO DELETE THE RECORD"
740 PRINT:PRINT "<RETURN> FOR NO CHANG
    ES":PRINT:PRINT:GOSUB 1150
750 G1=1:I=M
760 PRINT A$(1,M);:PS=""":INPUT PS
770 IF PS="" THEN 820
780 IF PS<>"/D/" THEN 810
790 FOR Z=1 TO 5:A$(Z,M)="-LAST RECORD
    IN FIL":NEXT
800 GOSUB 1630:N=N-1:RETURN
810 A$(1,M)=PS:G2=G2+1
820 FOR Z=2 TO 5:PRINT A$(Z,M);:PS="":I
    NPUT PS
830 IF PS="" THEN 850
840 A$(Z,M)=PS:G2=G2+1
850 NEXT Z
860 GOSUB 1030
870 ON T GOTO 760,720,880
880 RETURN
890 IF G2<>0 THEN GOSUB 1630
900 FOR XX=1 TO N:M=35-LEN(A$(2,XX))
910 PRINT TAB(4)XX;" ";A$(1,XX);TAB(M);
    A$(2,XX)
920 I1=INT(XX/10):IF I1<>XX/10 THEN 940
930 GOSUB 2410
940 NEXT XX:GOSUB 2410:RETURN
950 REM  SINGLE ITEM LISTING
960 M$="":INPUT"WHICH ONE";M$:IF M$="" 
    THEN 1020
970 GOSUB 1150
980 CLS:LOCATE 3
990 FOR I=1 TO 5:PRINT"      ";:PRINT A$(I
    ,M):NEXT
1000 GOSUB 1030
1010 ON T GOTO 980,960,1020
1020 RETURN
1030 PRINT:PRINT:PRINT "PRESS":PRINT:PRI
     NT "E FOR PREVIOUS RECORD"
1040 PRINT"X FOR NEXT RECORD":PRINT"F TO
     FIND ANOTHER RECORD":GOSUB 2410
1050 T=3:IF K<>69 THEN 1080
1060 T=1:IF M=1 THEN 1130
1070 M=M-1
1080 IF K<>88 THEN 1110
1090 T=1:IF M=N THEN 1130
1100 M=M+1
1110 IF K<>70 THEN 1130
1120 T=2
1130 RETURN
1140 REM  SEARCH ROUTINE FOR SINGLE ITEM
     LISTING
```

IBM PCjr

```
1150 IF ABS(ASC(M$)-53)>4 THEN 1190
1160 M=VAL(M$):IF M<N OR M=N THEN 1180
1170 M=N
1180 RETURN
1190 FOR I=1 TO N:M=I
1200 IF LEN(M$)>LEN(A$(1,I)) THEN 1220
1210 M$=LEFT$(A$(1,I),LEN(M$)) THEN 1
     230
1220 NEXT I
1230 RETURN
1240 REM LOAD DATA
1250 PRINT"ENTER 1. TAPE"
1260 PRINT"      2. DISK"
1270 PRINT:PRINT:INPUT"YOUR CHOICE";AS
1280 IF AS<>"1" OR AS>"2" THEN 1250
1290 IF AS="2" THEN AS$="A:":GOTO 1310
1300 AS$="CAS1:":PRINT"PLEASE REWIND CAS
     SETTE AND PRESS PLAY."
1310 PRINT:INPUT"ENTER FILENAME";N$
1320 OPEN"I",#2,AS$+N$
1330 INPUT #2,O,NT,F$,D$
1340 IF O=OP THEN 1410
1350 IF O=2 THEN 1370
1360 PRINT:PRINT"FILE IS A PHONE BOOK TY
     PE":GOTO 1380
1370 PRINT:PRINT"FILE IS AN INVENTORY TY
     PE"
1380 PRINT"PROCEED (Y/N)?"
1390 GOSUB 2420:IF K=78 THEN CLOSE 2:RET
     URN
1400 IF K<>89 THEN 1390
1410 OP=O:N=NT
1420 PRINT"LAST UPDATE:";D$
1430 FOR I=1 TO N:FOR J=1 TO 5
1440 INPUT #2,A$(J,I):NEXT J,I
1450 CLOSE 2:GOSUB 2410:RETURN
1460 REM SAVE DIRECTORY
1470 IF G2<>0 THEN GOSUB 1630
1480 PRINT"      ENTER 1. TAPE"
1490 PRINT"            2. DISK"
1500 PRINT:INPUT"YOUR CHOICE";AS
1510 IF AS<1 OR AS>2 THEN 1480
1520 IF AS=2 THEN AS$="A:":GOTO 1540
1530 AS$="CAS1:":PRINT"PLEASE REWIND CAS
     SETTE AND PRESS":PRINT"RECORD/PLAY.
1540 INPUT "ENTER DATE";D$
1550 PRINT:INPUT"ENTER FILENAME";N$
1560 OPEN"O",#2,AS$+N$
1570 WRITE #2,OP,N,S$,D$
1580 FOR I=1 TO N:FOR X=1 TO 5
1590 WRITE #2,A$(X,I):NEXT X,I
1600 CLOSE 2
```

Secretary . . . from p. 135

ELECTRONIC HOME SECRETARY (Apple)
Explanation of the Program

Line Nos.	
100-160	Program header.
170	Dim arrays.
180-200	Title screen.
210-220	Define constants.
230-250	Get input of program option.
260-340	Display menu.
350-370	Get input and branch.
380-390	Define title of program.
400-490	Data entry routine.
600-650	Sort routine.
660-790	Data alteration routine.
800-860	Search for single record.
870-1100	List directory routine.
1110-1340	Save file to disk routine.
1350-1600	Load file from disk routine.
1610-1710	List to printer routine.
1720-1730	Beginning of use data routine.
1740-1900	Telephone timer routine.
1910-1950	Total inventory routine.
1960-1970	End program.
1980-2010	Press Return to goto menu routine.

APPLE II Series

```
100 REM  ********************
110 REM  *  ELECTRONIC HOME SECRETARY  *
120 REM  ********************
130 REM  BY MALLADI SUBBAIAH AND THE HCM
     STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  APPLE II SERIES APPLESOFT
170 DIM A$(5,60),O$(5),C$(5,2):D$ = CH
    R$ (4)
180 HOME
```

APPLE II Series

```
190 VTAB 9: HTAB 8: PRINT "*********
    ********": HTAB 8: PRINT "*  THE
    HOME SECRETARY  *": HTAB 8: PRINT
    "**********": VTAB 14: HTAB 14:
    PRINT "PRESS EITHER:"
200 PRINT : HTAB 12: PRINT "1. FOR PHON
    E BOOKRY": HTAB 12: PRINT "2. FOR INV
    ENTORY"
210 C$(1,1) = "NAME":C$(2,1) = "PHONE":
    C$(3,1) = "ADDRESS":C$(4,1) = "CITY
    ":C$(5,1) = "STATE&ZIP"
220 C$(1,2) = "ITEM":C$(2,2) = "VALUE":
    C$(3,2) = "LOCATION":C$(4,2) = "DAT
    E PURCHASED":C$(5,2) = "SER# OR ID"
230 GET K$: IF K$ < "1" OR K$ > "2" THE
    N 230
240 IF K$ = "1" THEN OP = 1: GOTO 260
250 OP = 2
260 REM  MENU SCREEN
270 GOSUB 380:TL = 0
280 HOME : HTAB 8: PRINT "** THE HOME S
    ECRETARY **"
290 HTAB 15: PRINT TI$
300 VTAB 3: PRINT "**********PRESS:":
    VTAB 6: PRINT "1. ADD DATA":
    VTAB 8: PRINT "2. CHANGE DATA":
    VTAB 10: PRINT "3. DISPLAY NAME LIST":
    VTAB 12: PRINT "4. DISPLAY ONE ENTRY"
310 VTAB 14: PRINT "5. SAVE DATA FILE":
    VTAB 16: PRINT "6. LOAD DATA FILE":
    VTAB 18: PRINT "7. LIST TO PRINTER":
    VTAB 20: PRINT "8. USE DATA"
320 IF OP = 1 THEN PRINT "TIMER": VTAB
    22: PRINT "9. END PROGRAM": GOTO
    340
330 VTAB 20: PRINT "TOTAL": VTAB 22: PR
    INT "9. END PROGRAM"
340 PRINT : PRINT "YOUR CHOICE:";
350 GET K$: IF K$ < "1" OR K$ > "9" THE
    N 350
360 PRINT K$: ON VAL (K$) GOSUB 400,66
    0,870,930,1110,1350,1610,1720,1960
370 GOTO 260
380 IF OP = 1 THEN TI$ = "PHONE BOOK":
    RETURN
390 TI$ = "INVENTORY": RETURN
400 REM  ENTER DATA ROUTINE
```

IBM PCjr

```
1610 G1=0:RETURN
1620 REM SORTING ROUTINE
1630 PRINT:PRINT "SORT BY WHICH FIELD?"
1640 FOR Z=1 TO 5:PRINT Z;"   ";B$(OP,Z):NEXT Z
1650 INPUT F
1660 IF F<1 OR F>5 THEN 1630
1670 G2=0
1680 PRINT:PRINT "  ***** SORTING DATA *****"
1690 FOR I=1 TO N-1:FOR J=N TO I+1 STEP -1
1700 IF A$(F,1)<=A$(F,J) THEN 1720
1710 GOSUB 1770
1720 NEXT J
1730 IF A$(F,I)<>A$(F,I+1) THEN 1750
1740 I=I+1
1750 NEXT I
1760 RETURN
1770 FOR Z=1 TO 5:O$(Z)=A$(Z,I):A$(Z,I)=A$(Z,J)
1780 A$(Z,J)=O$(Z):NEXT Z:RETURN
1790 REM UTILITY PROGRAMS
1800 IF OP=1 THEN 1870
1810 SM=0
1820 FOR I=1 TO N:SM=SM+VAL(A$(2,I)):NEXT I
1830 LOCATE 10,6:PRINT"TOTAL COST OF ALL THE ITEMS:":PRINT:PRINT TAB(18) SM
1840 GOSUB 2410:RETURN
1850 REM DIAL PHONE
1860 REM CLOCK TIME FOLLOWS
1870 GOSUB 960:IF M$="" THEN 2010
1880 CLS:LOCATE 2
1890 T$=A$(2,M):L=LEN(T$)
1900 FOR I=1 TO 5:PRINT"   ";A$(I,M):NEXT:LOCATE 10,5
1910 IF L<10 THEN 1930
1920 L=L+1:T$="1"+T$
1930 FOR J=1 TO L:P$=MID$(T$,J,1)
1940 PRINT P$;:IF ASC(P$)<48 OR ASC(P$)>57 THEN 2000
1950 D1=15:D2=D1:VL1=0:VL2=0:T=VAL(P$):IF T<>0 THEN 1970
1960 F2=941:F1=1336:GOSUB 2250:GOTO 1990
1970 I=INT((T-1)/3)+1:IJ=T-3*(I-1)
1980 F1=P1(I):F2=P2(IJ):GOSUB 2250
1990 FOR X=1 TO 250:NEXT
2000 NEXT J
2010 PRINT:PRINT:PRINT" PRESS":PRINT
2020 PRINT" R TO REDIAL":PRINT" S TO START STOPWATCH"
2030 PRINT" N FOR NEW NUMBER"
2040 GOSUB 2410
2050 IF K=82 THEN 1880 ELSE IF K=78 THEN 1870 ELSE IF K<>83 THEN 2070
2060 GOSUB 2450:GOSUB 2090
2070 RETURN
2080 REM STOP WATCH
2090 TIME$="00:00:00"
2100 LOCATE 18,16:PRINT TIME$
2110 PRINT:PRINT"          PRESS ANY KEY TO CONTINUE"
2120 K$=INKEY$:IF K$="" THEN 2100 ELSE IF K$="R" THEN 1880 ELSE RETURN
2130 REM COMPLETE LISTING ON PRINTER
2140 OPEN "O",#2,"LPT1:"
2150 FOR I=1 TO N:FOR J=1 TO 5
2160 PRINT #2,A$(J,I):NEXT J
2170 PRINT #2," ":PRINT #2," ":NEXT I
2180 CLOSE #2:RETURN
2190 CLS:LOCATE 5:PRINT"       DO YOU WISH TO HALT THE PROGRAM"
2200 PRINT"      AND LOSE ALL THE DATA IN MEMORY?"
2210 PRINT:PRINT"       PRESS Y OR N"
2220 GOSUB 2420
2230 IF K<>89 THEN RETURN
2240 END
2250 SOUND ON:BEEP OFF
2260 F1=FNIFREQ(F1):F2=FNIFREQ(F2):F3=FNIFREQ(F3)
2270 F11=F1/16:F22=F2/16:F33=F3/16
2280 OUT 192,128+(F1 MOD 16):OUT 192,F11
2290 OUT 192,160+(F2 MOD 16):OUT 192,F22
2300 OUT 192,192+(F3 MOD 16):OUT 192,F33
2310 OUT 192,144+VL1
2320 OUT 192,176+VL2
2330 OUT 192,208+VL3
2340 D1=D1-1:D2=D2-1:D3=D3-1
2350 IF D1=0 THEN OUT 192,159
2360 IF D2=0 THEN OUT 192,191
2370 IF D3=0 THEN OUT 192,223
2380 IF D1<0 AND D2<0 AND D3<0 THEN 2390 ELSE 2340
2390 D1=1:D2=D1:D3=D1:VL1=15:VL2=VL1:VL3=VL1
2400 SOUND OFF:BEEP ON:RETURN
2410 PRINT:PRINT "          PRESS ANY KEY TO CONTINUE"
2420 K$=INKEY$:IF K$<>"" THEN 2420
2430 K$=INKEY$:IF K$="" THEN 2430
2440 K=ASC(K$):RETURN
2450 LOCATE 15:FOR I=1 TO 4:PRINT"                              ":NEXT:RETURN
```

HCM

APPLE II Series

```
410 HOME : HTAB 10: PRINT "DATA ENTRY ROUTINE": VTAB 3: PRINT "IF 'E' IS YOUR ONLY INITIAL DATA ENTRY  YOU WILL EXIT TO SORT THE DATA": VTAB 6: PRINT "IF 'R' IS YOUR ONLY INITIAL DATA ENTRY  YOU WILL REDO YOUR LAST RECORD"
420 POKE 34,8
430 I = N + 1: IF N = 60 THEN PRINT "  ***ARRAY FULL***": PRINT "  ***NO MORE ENTRIES ALLOWED***": POKE 34,0: GOTO 1980
440 IF I = 0 THEN I = 1:N = 0
450 HOME : VTAB 10: PRINT "  RECORD #";I: PRINT C$(1,OP);":";: INPUT "";A$(1,I)
460 IF A$(1,I) = "" THEN 450
470 IF A$(1,I) = "E" THEN POKE 34,0:F1 = 1: GOSUB 520: GOTO 280
480 IF A$(1,I) = "R" THEN I = I - 1:N = N - 1: GOTO 440
490 FOR J = 2 TO 5: PRINT C$(J,OP);":";: INPUT "";A$(J,I): NEXT :N = I: GOTO 430
500 REM   SORT ROUTINE
510 IF F1 = 0 THEN  RETURN
520 HOME : PRINT "SORT BY WHICH FIELD?": VTAB 3: FOR K = 1 TO 5: PRINT K;".  ";C$(K,OP): NEXT
530 GET K$: IF K$ < "1" OR K$ > "5" THEN 530
540 F =  VAL (K$): VTAB 10: PRINT "***SORTING DATA BY ";C$(F,OP);"***":F1 = 0
550 G = N * 1.5
560 G =  INT (G / 2)
570 IF G = 0 THEN  RETURN
580 FOR J = 1 TO (N - G):K = J
590 L = K + G: IF A$(F,K) <  = A$(F,L) THEN 620
600 GOSUB 650
610 K = K - G: IF K > 0 THEN 590
620 NEXT J
630 GOTO 560
640 REM   SWITCH MEMBERS
650 FOR S = 1 TO 5:O$(S) = A$(S,K):A$(S,K) = A$(S,L):A$(S,L) = O$(S): NEXT S: RETURN
660 REM   DATA ALTERATION
670 HOME : HTAB 10: PRINT "CHANGE DATA ROUTINE": PRINT
680 PRINT "SELECT RECORD TO BE CHANGED": PRINT "THEN EITHER:": PRINT "  1. ENTER CHANGES": PRINT "  2. PRESS RETURN TO NOT CHANGE DATA": PRINT "  3. ENTER /D/ TO DELETE RECORD": POKE 34,8
690 HOME : VTAB 9: INPUT "WHICH RECORD? ";M$: IF M$ = "" THEN 690
700 GOSUB 800
710 VTAB 11:F1 = 1:I = M: PRINT "CHANGE "C$(1,OP);":  ";A$(1,M): INPUT "TO ";P$: IF P$ = "" THEN 750
720 IF P$ <  > "/D/" THEN 740
730 FOR J = 1 TO 5:A$(J,M) = "_": NEXT : POKE 34,0: GOSUB 500:N = N - 1: RETURN
740 A$(1,M) = P$
750 FOR J = 2 TO 5: PRINT "CHANGE ";C$(J,OP);":  ";A$(J,M): PRINT "TO ";
760 INPUT O$(J): IF O$(J) = "" THEN 780
770 A$(J,M) = O$(J)
780 NEXT J
790 POKE 34,0: RETURN
800 REM SEARCH ROUTINE FOR SINGLE ITEM
810 IF  ABS ( ASC (M$) - 53) > 4 THEN 840
820 M =  VAL (M$): IF M <  = N THEN  RETURN
830 M = N: RETURN
840 FOR I = 1 TO N:M = I: IF  LEN (M$) >  LEN (A$(1,I)) THEN 860
850 IF M$ =  LEFT$ (A$(1,I), LEN (M$)) THEN  RETURN
860 NEXT I: RETURN
870 REM   LIST  DIRECTORY
880 IF F1 <  > 0 THEN  GOSUB 500
890 HOME : PRINT TI$: PRINT : FOR I = 1 TO N:M = 40 -  LEN (A$(2,I)):T$ =  STR$ (I) + ". "
900 HTAB (5 -  LEN (T$)): PRINT T$;A$(1,I);: HTAB M: PRINT A$(2,I)
910 IF  INT (I / 20) = (I / 20) THEN  GOSUB 1980: HOME : PRINT TI$: PRINT
920 NEXT I: GOSUB 1980: RETURN
930 REM   SINGLE ITEM LISTING
940 POKE 34,0: HOME : HTAB 10: PRINT TI$: VTAB 3: PRINT "SINGLE RECORD LISTING":
950 POKE 34,4: VTAB 5: INPUT "SELECT RECORD TO BE DISPLAYED:";M$: IF M$ = "" THEN 940
```

Continued on p. 142

SEND $2.00 for our current catalog of TI and 3rd PARTY CATALOG FOR THE 99/4A Home Computer. We include a $5.00 certificate for use with your first order!!

WHAT NOW?

The one to start with is the one to stay with.

Texas Instruments' Home Computer.

32K Memory Expansion Card
RS232 Interface Card

SPECIAL INTRODUCTORY OFFER:
ORDER BOTH CARDS TOGETHER AND DEDUCT $10.00.....

CORCOMP is a former TI-99/4A supplier and its Model Masters Division is the developer of the TI Mini-Writer software and soon to be released action game "Intercept" on module.

TEX-COMP™

TI USERS SUPPLY

TEX-COMP is pround to announce that it has been appointed an authorized dealer by CORCOMP the leading manufacturer of TI compatible peripherals and accessories. As reported in InfoWorld and the Los Angeles Times, CORCOMP is now marketing a fully compatible 32K memory expansion card and RS232 card with a disk controller card to follow soon.

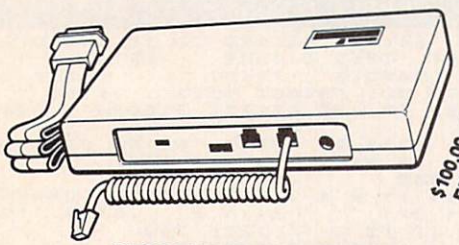
TEX-COMP is now taking orders on the 32K card and RS232 card for immediate delivery from stock. Both cards meet or exceed TI performance and quality specifications and carry a 12 month limited factory warranty.

32K Memory Expansion Card.....$119.95

RS232 Card.....$79.95

Visa and MasterCard
Order Direct
(818) 366-6631

Send Check or Money Order to: **TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •**



$100.00 SPECIAL $100.00 FREE Source/TEXNET with purchase of Signalman MK III

SIGNALMAN™ MARK III
TI 99/4A COMPATABLE
MODEM

Finally, a low cost, direct connect, high quality and super reliable TI-99/4 and 99/4A compatable modem that comes complete and ready to use — just plug it into a TI RS/232 expansion card.

Comes complete with cable and cord. This is a unit specially designed for the TI-99/4A and should not be confused with other Anchor/Signalman models that sell for less and/or are not compatible with the TI-99/4A.

SPECIAL: WE HAVE A LIMITED SUPPLY OF TI TERMINAL EMULATOR II WHICH CAN BE ORDERED WITH THE MODEM FOR $24.95. (one per modem please)

ACCESSORY: 9VOLT BATTERY ELIMINATOR FOR MARK III.....$10.95

Mfg. Suggested List Price $139.00	**TEX-COMP PRICE** Limited-Time Introductory Offer **$94.95** Including shipping & insurance

SUPER JOYSTICK II

The Only JOYSTICK that can control THE MUNCHMAN!

Available exclusively from Tex-Comp

Rated #1 August 1982 Creative Computing

Improved version

The case has been redesigned for easy hand-held operation. The solid steel, short-throw control shaft with an arcade-style ball on the top offers precise, 8-position action. The "soft-touch" fire button is located at the top end of the molded base allowing either right or left-hand play. This improved fire button location is designed to take advantage of the greater dexterity of the index finger resulting in quicker response, time and less fatigue during extended play

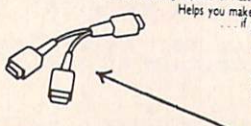
SAME ACTION AS THE REAL ARCADE GAMES

•SWITCHABLE GATE-PLATE™ Setting the switchable gate-plate indicator to the 4-way position allows only horizontal or vertical motion dramatically improving response for maze-type games

Helps you make higher levels of Munchman ... if you're fast enough

NEW PRICE 29.95

Free Shipping*
Second Unit Only $24.95
*Continental U.S.



With FREE Port Expander

Credit Cards add 3%

[818] 366-6631

MasterCard VISA

TEX-COMP™

TI USERS SUPPLY COMPANY

Send Check or Money Order to:

• P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •

Authorized Dealer

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100. and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads.*

IBM PCjr

```
1610 G1=0:RETURN
1620 REM SORTING ROUTINE
1630 PRINT:PRINT "SORT BY WHICH FIELD?"
1640 FOR Z=1 TO 5:PRINT Z;"   ";BS(OP,Z):
     NEXT Z
1650 INPUT F
1660 IF F<1 OR F>5 THEN 1630
1670 G2=0
1680 PRINT:PRINT "   ***** SORTING DATA *
     ****"
1690 FOR I=1 TO N-1:FOR J=N TO I+1 STEP
     -1
1700 IF A$(F,1)<=A$(F,J) THEN 1720
1710 GOSUB 1770
1720 NEXT J
1730 IF A$(F,I)<>A$(F,I+1) THEN 1750
1740 I=I+1
1750 NEXT I
1760 RETURN
1770 FOR Z=1 TO 5:O$(Z)=A$(Z,I):A$(Z,I)=
     A$(Z,J)
1780 A$(Z,J)=O$(Z):NEXT Z:RETURN
1790 REM UTILITY PROGRAMS
1800 IF OP=1 THEN 1870
1810 SM=0
1820 FOR I=1 TO N:SM=SM+VAL(A$(2,I)):NEX
     T I
1830 LOCATE 10,6:PRINT"TOTAL COST OF ALL
      THE ITEMS:":PRINT:PRINT TAB(18) SM
1840 GOSUB 2410:RETURN
1850 REM DIAL PHONE
1860 REM CLOCK TIME FOLLOWS
1870 GOSUB 960:IF M$="" THEN 2010
1880 CLS:LOCATE 2
1890 T$=A$(2,M):L=LEN(T$)
1900 FOR I=1 TO 5:PRINT"   ";A$(I,M):NEXT
     :LOCATE 10,5
1910 IF L<10 THEN 1930
1920 L=L+1:T$="1"+T$
1930 FOR J=1 TO L:P$=MID$(T$,J,1)
1940 PRINT P$;:IF ASC(P$)<48 OR ASC(P$)>
     57 THEN 2000
1950 D1=15:D2=D1:VL1=0:VL2=0:T=VAL(P$):I
     F T<>0 THEN 1970
1960 F2=941:F1=1336:GOSUB 2250:GOTO 1990
1970 I=INT((T-1)/3)+1:IJ=T-3*(I-1)
1980 F1=P1(I):F2=P2(IJ):GOSUB 2250
1990 FOR X=1 TO 250:NEXT
2000 NEXT J
2010 PRINT:PRINT:PRINT" PRESS":PRINT
2020 PRINT" R TO REDIAL":PRINT" S TO STA
     RT STOPWATCH"
2030 PRINT" N FOR NEW NUMBER"
```

IBM PCjr

```
2040 GOSUB 2410
2050 IF K=82 THEN 1880 ELSE IF K=78 THEN
      1870 ELSE IF K<>83 THEN 2070
2060 GOSUB 2450:GOSUB 2090
2070 RETURN
2080 REM STOP WATCH
2090 TIME$="00:00:00"
2100 LOCATE 18,16:PRINT TIME$
2110 PRINT:PRINT"          PRESS ANY KEY TO
      CONTINUE"
2120 K$=INKEY$:IF K$="" THEN 2100 ELSE I
     F K$="R" THEN 1880 ELSE RETURN
2130 REM COMPLETE LISTING ON PRINTER
2140 OPEN "O",#2,"LPT1:"
2150 FOR I=1 TO N:FOR J=1 TO 5
2160 PRINT #2,A$(J,I):NEXT J
2170 PRINT #2," ":PRINT #2,"  ":NEXT I
2180 CLOSE #2:RETURN
2190 CLS:LOCATE 5:PRINT"          DO YOU WISH
      TO HALT THE PROGRAM"
2200 PRINT"       AND LOSE ALL THE DATA IN
      MEMORY?"
2210 PRINT:PRINT"          PRESS Y OR N"
2220 GOSUB 2420
2230 IF K<>89 THEN RETURN
2240 END
2250 SOUND ON:BEEP OFF
2260 F1=FNIFREQ(F1):F2=FNIFREQ(F2):F3=FN
     IFREQ(F3)
2270 F11=F1/16:F22=F2/16:F33=F3/16
2280 OUT 192,128+(F1 MOD 16):OUT 192,F11
2290 OUT 192,160+(F2 MOD 16):OUT 192,F22
2300 OUT 192,192+(F3 MOD 16):OUT 192,F33
2310 OUT 192,144+VL1
2320 OUT 192,176+VL2
2330 OUT 192,208+VL3
2340 D1=D1-1:D2=D2-1:D3=D3-1
2350 IF D1=0 THEN OUT 192,159
2360 IF D2=0 THEN OUT 192,191
2370 IF D3=0 THEN OUT 192,223
2380 IF D1<0 AND D2<0 AND D3<0 THEN 2390
      ELSE 2340
2390 D1=1:D2=D1:D3=D1:VL1=15:VL2=VL1:VL3
     =VL1
2400 SOUND OFF:BEEP ON:RETURN
2410 PRINT:PRINT "          PRESS ANY KEY T
     O CONTINUE"
2420 K$=INKEY$:IF K$<>"" THEN 2420
2430 K$=INKEY$:IF K$="" THEN 2430
2440 K=ASC(K$):RETURN
2450 LOCATE 15:FOR I=1 TO 4:PRINT"
                                   ":N
     EXT:RETURN
```

HCM

APPLE II Series

```
410 HOME : HTAB 10: PRINT "DATA ENTRY R
    OUTINE": VTAB 3: PRINT "IF 'E' IS Y
    OUR ONLY INITIAL DATA ENTRY YOU WI
    LL EXIT TO SORT THE DATA": VTAB 6:
    PRINT "IF 'R' IS YOUR ONLY INITIAL
    DATA ENTRY YOU WILL REDO YOUR LAST
     RECORD"
420 POKE 34,8
430 I = N + 1: IF N = 60 THEN PRINT "
     ***ARRAY FULL***": PRINT " ***NO
     MORE ENTRIES ALLOWED***": POKE 34,
    0: GOTO 1980
440 IF I = 0 THEN I = 1:N = 0
450 HOME : VTAB 10: PRINT "  RECORD #";
    I: PRINT C$(1,OP);":";: INPUT "";A$
    (1,I)
460 IF A$(1,I) = "" THEN 450
470 IF A$(1,I) = "E" THEN POKE 34,0:F1
     = 1: GOSUB 520: GOTO 280
480 IF A$(1,I) = "R" THEN I = I - 1:N =
     N - 1: GOTO 440
490 FOR J = 2 TO 5: PRINT C$(J,OP);":";
    : INPUT "";A$(J,I): NEXT :N = I: GO
    TO 430
500 REM  SORT ROUTINE
510 IF F1 = 0 THEN  RETURN
520 HOME : PRINT "SORT BY WHICH FIELD?"
    : VTAB 3: FOR K = 1 TO 5: PRINT K;"
    . ";C$(K,OP): NEXT
530 GET K$: IF K$ < "1" OR K$ > "5" THE
    N 530
540 F =  VAL (K$): VTAB 10: PRINT "***S
    ORTING DATA BY ";C$(F,OP);"***":F1
    = 0
550 G = N * 1.5
560 G =  INT (G / 2)
570 IF G = 0 THEN  RETURN
580 FOR J = 1 TO (N - G):K = J
590 L = K + G: IF A$(F,K) <  = A$(F,L)
    THEN 620
600 GOSUB 650
610 K = K - G: IF K > 0 THEN 590
620 NEXT J
630 GOTO 560
640 REM  SWITCH MEMBERS
650 FOR S = 1 TO 5:O$(S) = A$(S,K):A$(S
    ,K) = A$(S,L):A$(S,L) = O$(S): NEXT
     S: RETURN
660 REM  DATA ALTERATION
670 HOME : HTAB 10: PRINT "CHANGE DATA
    ROUTINE": PRINT
```

APPLE II Series

```
680 PRINT "SELECT RECORD TO BE CHANGED"
    : PRINT "THEN EITHER:": PRINT " 1.
    ENTER CHANGES": PRINT " 2. PRESS RE
    TURN TO NOT CHANGE DATA": PRINT " 3
    . ENTER /D/ TO DELETE RECORD": POKE
     34,8
690 HOME : VTAB 9: INPUT "WHICH RECORD?
     ";M$: IF M$ = "" THEN 690
700 GOSUB 800
710 VTAB 11:F1 = 1:I = M: PRINT "CHANGE
     "C$(1,OP);": ";A$(1,M): INPUT "TO
    ";P$: IF P$ = "" THEN 750
720 IF P$ <  > "/D/" THEN 740
730 FOR J = 1 TO 5:A$(J,M) = "_": NEXT
    : POKE 34,0: GOSUB 500:N = N - 1: R
    ETURN
740 A$(1,M) = P$
750 FOR J = 2 TO 5: PRINT "CHANGE ";C$(
    J,OP);": ";A$(J,M): PRINT "TO ";
760 INPUT O$(J): IF O$(J) = "" THEN 780
770 A$(J,M) = O$(J)
780 NEXT J
790 POKE 34,0: RETURN
800 REM SEARCH ROUTINE FOR SINGLE ITEM
810 IF  ABS ( ASC (M$) - 53) > 4 THEN 8
    40
820 M =  VAL (M$): IF M <  = N THEN  RE
    TURN
830 M = N: RETURN
840 FOR I = 1 TO N:M = I: IF  LEN (M$)
    >  LEN (A$(1,I)) THEN 860
850 IF M$ =  LEFT$ (A$(1,I), LEN (M$))
    THEN  RETURN
860 NEXT I: RETURN
870 REM  LIST  DIRECTORY
880 IF F1 <  > 0 THEN  GOSUB 500
890 HOME : PRINT TI$: PRINT : FOR I = 1
     TO N:M = 40 -  LEN (A$(2,I)):T$ =
     STR$ (I) + ". "
900 HTAB (5 -  LEN (T$)): PRINT T$;A$(1
    ,I);: HTAB M: PRINT A$(2,I)
910 IF  INT (I / 20) = (I / 20) THEN  G
    OSUB 1980: HOME : PRINT TI$: PRINT
920 NEXT I: GOSUB 1980: RETURN
930 REM  SINGLE ITEM LISTING
940 POKE 34,0: HOME : HTAB 10: PRINT TI
    $: VTAB 3: PRINT "SINGLE RECORD LIS
    TING":
950 POKE 34,4: VTAB 5: INPUT "SELECT RE
    CORD TO BE DISPLAYED:";M$: IF M$ =
    "" THEN 940
```

Continued on p. 142

SEND $2.00 for our current catalog of TI and 3rd PARTY CATALOG FOR THE 99/4A Home Computer. We include a $5.00 certificate for use with your first order!!

WHAT NOW?

The one to start with is the one to stay with.

Texas Instruments' Home Computer.

32K Memory Expansion Card
RS232 Interface Card

SPECIAL INTRODUCTORY OFFER:
ORDER BOTH CARDS TOGETHER AND DEDUCT $10.00.....

CORCOMP is a former TI-99/4A supplier and its Model Masters Division is the developer of the TI Mini-Writer software and soon to be released action game "Intercept" on module.

TEX-COMP™

TI USERS SUPPLY

TEX-COMP is pround to announce that it has been appointed an authorized dealer by CORCOMP the leading manufacturer of TI compatible peripherals and accessories. As reported in InfoWorld and the Los Angeles Times, CORCOMP is now marketing a fully compatible 32K memory expansion card and RS232 card with a disk controller card to follow soon.

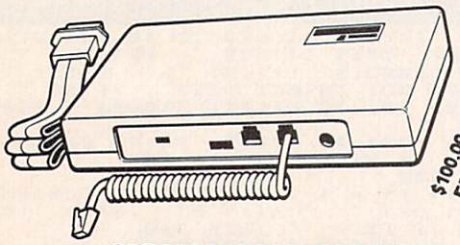
TEX-COMP is now taking orders on the 32K card and RS232 card for immediate delivery from stock. Both cards meet or exceed TI performance and quality specifications and carry a 12 month limited factory warranty.

32K Memory Expansion Card.....$119.95

RS232 Card.....$79.95

Visa and MasterCard
Order Direct
(818) 366-6631

Send Check or Money Order to: **TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •**



$100.00 SPECIAL $100.00 FREE Source/TEXNET with purchase of Signalman MK III

Comes complete with cable and cord. This is a unit specially designed for the TI-99/4A and should not be confused with other Anchor/Signalman models that sell for less and/or are not compatible with the TI-99/4A.

SPECIAL: WE HAVE A LIMITED SUPPLY OF TI TERMINAL EMULATOR II WHICH CAN BE ORDERED WITH THE MODEM FOR $24.95. (one per modem please)

ACCESSORY: 9VOLT BATTERY ELIMINATOR FOR MARK III.....$10.95

SIGNALMAN™ MARK III
TI 99/4A COMPATABLE
MODEM

Finally, a low cost, direct connect, high quality and super reliable TI-99/4 and 99/4A compatable modem that comes complete and ready to use — just plug it into a TI RS/232 expansion card.

Mfg. Suggested List Price $139.00

TEX-COMP PRICE
Limited-Time Introductory Offer
$94.95
Including shipping & insurance

SUPER JOYSTICK II

The Only JOYSTICK that can control THE MUNCHMAN!

Available exclusively from Tex-Comp

Rated #1 August 1982 Creative Computing

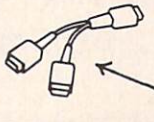
Improved version

The case has been redesigned for easy hand-held operation. The solid steel, short-throw control shaft with an arcade-style ball on the top offers precise, 8-position action. The "soft-touch" fire button is located at the top end of the molded base allowing either right or left-hand play. This improved fire button location is designed to take advantage of the greater dexterity of the index finger resulting in quicker response, time and less fatigue during extended play

SAME ACTION AS THE REAL ARCADE GAMES

•SWITCHABLE GATE-PLATE™: Setting the switchable gate-plate indicator to the 4-way position allows only horizontal or vertical motion dramatically improving response for maze-type games

Helps you make higher levels of Munchman . . . if you're fast enough



NEW PRICE 29.95

Free Shipping*
Second Unit Only $24.95
*Continental U.S.

With FREE Port Expander

TEX-COMP™

TI USERS SUPPLY COMPANY

Send Check or Money Order to:

• P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •

Authorized Dealer

Credit Cards add 3%

[818] 366-6631

MasterCard VISA

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100 and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads.*

ELECTRONIC HOME SECRETARY (C-64)
Explanation of the Program

Line Nos.	
100-160	Header.
170-280	Initialization.
290-340	Display first screen.
350-390	Initial selection of mode.
400-420	Initialization.
430-450	Array full subroutine.
470-540	Subroutine to add array items.
550-660	Subroutine to display menu.
680	Main GOSUB statement. All program functions initiate from here.
700-710	Add data subroutine.
720-890	Data alteration subroutine.
900-950	Display name list subroutine routine.
960-1030	Display one entry routine.
1040-1140	Routine to go forward or backward in array.
1150-1240	Search routine for single item listing.
1250-1500	Load data file subroutine.
1510-1700	Save data file subroutine.
1710-1870	Sorting routine.
1880-2190	Use the data subroutine.
2190-2270	Stopwatch subroutine.
2280-2330	Print array on 1525 printer.
2340-2400	End program routine.
2410-2510	Sound subroutine.
2520-2550	Get key subroutine.

COMMODORE 64

```
100 REM  ********************************
110 REM  *  ELECTRONIC HOME SECRETARY  *
120 REM  ********************************
130 REM  BY MALLADI SUBBAIAH AND THE HCM
     STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  C64 BASIC
170 DIM A$(5,60),O$(5),C$(6),P1(3),P2(3
    ),B$(2,5)
180 S=54272:SZ=0
190 OP=1
200 READ P1(1),P1(2),P1(3),P2(1),P2(2),
    P2(3)
210 FOR I=1TO5:READ B$(1,I):NEXT
220 FOR I=1TO5:READ B$(2,I):NEXT
230 DATA 697,770,852,1210,1340,1481
240 DATA "NAME:","PHONE:","ADDRESS:","C
    ITY:","STATE&ZIP:"
250 DATA "ITEM :","VALUE :","LOCATION :
    "," DATE PURCHASED :","SER # OR ID :
    "
260 RF=2000:RF=(RF-30)/8:RH=INT(RF/8):R
    L=RF-8*RH
270 FOR X=STOS+24:POKE X,0:NEXT
280 POKE S+23,3:POKE S+21,RH:POKE S+22,
    RL
290 PRINT "■CTRL BLK■■SHIFT CLR■■2CRSRD
    OWN■":POKE 53281,1
300 PRINT TAB(8)"**********************
    **"
310 PRINT TAB(8)"** THE HOME SECRETARY
    **"
320 PRINT TAB(8)"**********************
    **■16CRSRDOWN■"
330 REM  NEW SET UP
340 PRINT TAB(8)"1. PHONE BOOK":PRINT T
    AB(8)"■CRSRDOWN■2. INVENTORY"
350 GOSUB 2530
360 IF K=49 THEN OP=1:GOTO 390
370 IF K=50 THEN OP=2:GOTO 390
380 PRINT "RE-ENTER 1 OR 2":GOTO 350
390 N=0:GOTO550
400 REM KEY INPUT FOR DATA SET UP
410 PRINT "■2CRSRDOWN■ENTER":PRINT "■CR
    SRDOWN■■CRSRRIGHT■E TO EXIT":PRINT"
    ■CRSRRIGHT■R TO REENTER":G1=1:G2=2
420 I=N+1
430 IF I<60 OR I=60 THEN 460
440 PRINT "■2CRSRDOWN■ ***  ARRAY FULL
    (N=60)***■2CRSRDOWN■"
450 GOSUB 2520:RETURN
460 PRINT "
470 PRINT "  B$(OP,1);:A$(1,I)="":INPUTA$(
    1,I)
480 IF A$(1,I)="E" OR A$(1,I)="END" THE
    N 1720
490 IF A$(1,I)="" THEN 470
500 IF A$(1,I)<>"R" THEN 520
510 I=I-1:N=I-1:PRINT "  ***  REENTER L
    AST SET ***":GOTO 460
520 FOR Z=2 TO 5
530 PRINT B$(OP,Z);:A$(Z,I)="":INPUT A$
    (Z,I):NEXT Z
540 N=I:GOTO 410
550 SC=4:PRINT "■SHIFT CLR■■2CRSRDOWN■■
    9CRSRRIGHT■** HOME SECRETARY ***"
560 IF OP=2 THEN 580
```

COMMODORE 64

```
570 PRINT TAB(15)"PHONE BOOK":GOTO 590
580 PRINT TAB(15)"INVENTORY"
590 PRINT "**********************
    **********"
600 PRINT "PRESS":PRINT"1. ADD DATA"
610 PRINT "2. CHANGE DATA":PRINT"3. DIS
    PLAY NAME LIST"
620 PRINT "4. DISPLAY ONE ENTRY":PRINT"
    5. USE THE DATA"
630 PRINT"6. SAVE DATA FILE":PRINT"7. L
    OAD DATA FILE"
640 PRINT"8. LIST TO PRINTER":PRINT"9.
    END PROGRAM"
650 GOSUB 2530
660 IF K<49 OR K>57 THEN 550
670 PRINT"■SHIFT CLR■"
680 ON (K-48) GOSUB 700,720,900,960,188
    0,1510,1250,2280,2340
690 GOTO 550
700 GOSUB 400
710 RETURN
720 REM DATA ALTERATION
730 INPUT "WHICH ONE";M$:IF M$="" THEN
    M$="1"
740 PRINT"■2CRSRDOWN■ENTER":PRINT "■CRS
    RDOWN■ NEW DATA AT CURSOR":PRINT"■C
    RSRDOWN■ /D/ TO DELETE THE ITEM"
750 PRINT"■CRSRDOWN■ ' RETURN' FOR NO CH
    ANGES":PRINT"■2CRSRDOWN■":GOSUB 116
    0
760 G1=1:I=M
770 PRINT A$(1,M);:P$="":INPUT P$
780 IF P$="" THEN 830
790 IF P$<>"/D/" THEN 820
800 FOR Z=1 TO 5:A$(Z,M)="-LAST RECORD
    IN FILE":NEXT
810 GOSUB 1720:N=N-1:RETURN
820 A$(1,M)=P$:G2=1
830 FOR Z=2 TO 5:PRINT A$(Z,M);:P$="":I
    NPUT P$
840 IF P$="" THEN 860
850 A$(Z,M)=P$:G2=G2+1
860 NEXT Z
870 GOSUB 1040
880 ON T GOTO 770,730,890
890 RETURN
900 IF G2<>0 THEN GOSUB 1720
910 FOR X=1TON:M=35-LEN(A$(2,XX)):T$="
    "
920 PRINT TAB(4-LEN(T$))XX;"■SHIFT CRSR
    LEFT■";T$;A$(1,XX);TAB(M);A$(2,XX)
930 I1=INT(XX/10):IF I1<>XX/10 THEN 95
    0
940 GOSUB 2520
950 NEXT X:GOSUB 2520:RETURN
960 REM  SINGLE ITEM LISTING
970 M$="":INPUT"■2CRSRDOWN■WHICH ONE";M
    $:IF M$="" THEN 1030
980 GOSUB 1160
990 PRINT"■SHIFT CLR■■3CRSRDOWN■■3CRSRR
    IGHT■"
1000 FOR I=1TO5:PRINT "■5CRSRRIGHT■";A$(
    1,M):NEXT
1010 GOSUB 1040
1020 ON T GOTO 990,970,1030
1030 RETURN
1040 PRINT "■2CRSRDOWN■PRESS■CRSRDOWN■":
     PRINT "E FOR PREVIOUS RECORD"
1050 PRINT"■CRSRDOWN■X FOR NEXT RECORD":
     PRINT"■CRSRDOWN■F TO FIND ANOTHER R
     ECORD":GOSUB 2520
1060 T=3:IF K<>69 THEN 1090
1070 T=1:IF M=1 THEN 1140
1080 M=M-1
1090 IF K<>88 THEN 1120
1100 T=1:IF M=N THEN 1140
1110 M=M+1
1120 IF K<>70 THEN 1140
1130 T=2
1140 RETURN
1150 REM  SEARCH ROUTINE FOR SINGLE ITEM
     LISTING
1160 IF ABS(ASC(M$)-53)>4 THEN 1200
1170 M=VAL(M$):IF M<N OR M=N THEN 1190
1180 M=N
1190 RETURN
1200 FOR I=1 TO N:M=I
1210 IF LEN(M$)>LEN(A$(1,I)) THEN 1230
1220 IF M$=LEFT$(A$(1,I),LEN(M$)) THEN 1
     240
1230 NEXT I
1240 RETURN
1250 REM  LOAD DATA
1260 PRINT "ENTER 1. TAPE"
1270 PRINT "      2. DISK"
1280 INPUT "■2CRSRDOWN■YOUR CHOICE";AS
1290 IF AS<1 OR AS>2 THEN 1260
1300 IF AS=2 THEN AS=2:SS=8:GOTO 1320
1310 AS=0:SS=1
1320 HS=CHR$(13):INPUT "ENTER FILE NAME"
     ;N$
1330 IF AS<>2 THEN 1370
1340 INPUT "ENTER DEVICE NUMBER";D
1350 OPEN2,SS,AS,"@0:"+N$+",S,R"
1360 GOTO 1380
1370 OPEN 2,SS,AS,N$
1380 INPUT#2,O,NT,F$,D$
```

Continued on next page

Secretary . . . from p. 141

COMMODORE 64

```
1390 IF O=OP THEN 1460
1400 IF O=2 THEN 1420
1410 PRINT "■CRSRDOWN■FILE IS A PHONE BO
     OK TYPE":GOTO 1430
1420 PRINT "■CRSRDOWN■FILE IS AN INVENTO
     RY TYPE"
1430 PRINT "PROCEDE (Y/N)?"
1440 GOSUB 2530:IF K=78 THEN RETURN
1450 IF K<>89 THEN 1440
1460 OP=O:N=NT
1470 PRINT "LAST UPDATE:";D$
1480 FOR I=1TON:FORJ=1TO5
1490 INPUT#2,A$(J,I):NEXT J:NEXT I
1500 CLOSE2:GOSUB 2520:RETURN
1510 REM SAVE DIRECTORY
1520 IF G2<>0 THEN GOSUB 1720
1530 PRINT "ENTER 1. TAPE"
1540 PRINT " 2. DISK"
1550 INPUT "■2CRSRDOWN■YOUR CHOICE";AS
1560 IF AS<1 OR AS>2 THEN 1530
1570 IF AS=2 THEN AS=2:SS=8:GOTO 1590
1580 AS=1:SS=1
1590 H$=CHR$(13):INPUT "ENTER FILE NAME"
     ;N$
1600 INPUT "ENTER DATE.";D$
1610 IF AS<>2 THEN 1650
1620 INPUT "ENTER DEVICE NUMBER";D
1630 OPEN 2,SS,AS,"@0:"+N$+",S,W"
1640 GOTO 1660
1650 OPEN 2,SS,AS,N$
1660 PRINT#2,OP;H$;N;H$;S$;H$;D$;H$
1670 FOR I=1 TO N:FOR X=1 TO 5:IF A$(X,I
     )="" THEN A$(X,I)="  "
1680 PRINT#2,A$(X,I);H$:NEXT X:NEXT I
1690 PRINT#2,;::CLOSE2
1700 G1=0:RETURN
1710 REM SORTING ROUTINE
1720 PRINT "SORT BY WHICH FIELD?"
1730 FOR Z=1 TO 5:PRINT Z;". ";B$(OP,Z):
     NEXT Z
1740 INPUT F
1750 IF F<1 OR F>5 THEN 1720
1760 G2=0
1770 PRINT "■3CRSRDOWN■*****  SORTING DATA
     *****■3CRSRDOWN■"
1780 FOR I=1 TO N-1:FOR J=N TO I+1 STEP
     -1
1790 IF A$(F,I)<A$(F,J) OR A$(F,I)=A$(F,
     J) THEN 1810
1800 GOSUB 1860
```

COMMODORE 64

```
1810 NEXT J
1820 IF A$(F,I)<>A$(F,I+1) THEN 1840
1830 I=I+1
1840 NEXT I
1850 RETURN
1860 FOR Z=1TO5:O$(Z)=A$(Z,I):A$(Z,I)=A$
     (Z,J)
1870 A$(Z,J)=O$(Z):NEXT Z:RETURN
1880 REM UTILITY PROGRAMS
1890 IF OP=1 THEN 1960
1900 SM=0
1910 FOR I=1TON:SM=SM+VAL(A$(2,I))):NEXT
     I
1920 PRINT "■2CRSRDOWN■■6CRSRRIGHT■TOTAL
     COST OF ALL THE ITEMS":PRINT TAB(1
     8)"■2CRSRDOWN■";SM
1930 GOSUB 2520:RETURN
1940 REM DIAL PHONE
1950 REM CLOCK TIME DELAYS FOLLOW
1960 GOSUB 970:IF M$="" THEN 2100
1970 PRINT "■SHIFT CLR■■2CRSRDOWN■■2CRSR
     RIGHT■"
1980 T$=A$(2,M):L=LEN(T$)
1990 FOR I=1TO5:PRINT "■2CRSRRIGHT■";A$(I
     ,M):NEXT:PRINT "■2CRSRDOWN■■15CRSRR
     IGHT■";
2000 IF L<10 THEN 2020
2010 L=L+1:T$="1"+T$
2020 FOR J=1 TO L:P$=MID$(T$,J,1)
2030 PRINT P$;:IF ASC(P$)<48 OR ASC(P$)>
     57 THEN 2090
2040 T=VAL(P$):IF T<>0 THEN 2060
2050 F2=941:F1=1336:GOSUB 2410:GOTO 2080
2060 I=INT((T-1)/3)+1:IJ=T-3*(I-1)
2070 F1=P1(I):F2=P2(IJ):GOSUB 2410
2080 FOR X=1 TO 250:NEXT
2090 NEXT J
2100 PRINT "■2CRSRDOWN■":PRINT "■CRSRRIG
     HT■PRESS":PRINT "■CRSRDOWN■"
2110 PRINT "■CRSRRIGHT■R TO REDIAL":PRINT
     " S TO START STOPWATCH"
2120 PRINT "■CRSRRIGHT■N FOR NEW NUMBER"
2130 GOSUB 2520
2140 IF K=82 THEN 1970
2150 IF K=78 THEN 1960
2160 IF K<>83 THEN 2180
2170 GOSUB 2560:GOSUB 2200
2180 RETURN
2190 REM STOP WATCH
2200 TI$="000000"
```

Secretary . . . from p. 139

APPLE II Series

```
 960 GOTO 980
 970 HOME: VTAB 5: PRINT "THIS IS RECOR
     D#";M: GOTO 990
 980 GOSUB 800
 990 PRINT: FOR I = 1 TO 5: PRINT C$(I,
     OP);": ";A$(I,M): NEXT I
1000 IF K$ = "8" THEN RETURN
1010 PRINT: PRINT "NOW CHOOSE ONE OF TH
     E FOLLOWING": PRINT: PRINT "1. VI
     EW PREVIOUS RECORD": PRINT "2. VIEW
      NEXT RECORD": PRINT "3. FIND ANOTH
     ER RECORD": PRINT "4. GO BACK TO ME
     NU"
1020 GET K$: IF K$ < "1" OR K$ > "4" THE
     N 1020
1030 K = VAL (K$): IF K = 4 THEN POKE
     34,0: RETURN
1040 IF K < 1 THEN 1070
1050 IF M > 1 THEN M = M - 1
1060 HOME: GOTO 970
1070 IF K < 2 THEN 1100
1080 IF M < N THEN M = M + 1
1090 HOME: GOTO 970
1100 GOTO 940
1110 REM SAVE FILE TO DISK
1120 IF F1 = 1 THEN GOSUB 500
1130 HOME: PRINT "SAVE FILE TO DISK": P
     RINT: PRINT "LAST FILE LOADED WAS
     ";F$: INPUT "WHAT FILE NAME?";F$
1140 PRINT "WHAT DATE DO YOU WANT ON THE
      FILE?": INPUT DT$
1150 FL$ = F$ + "." + LEFT$ (T1$,3)
1160 ONERR GOTO 1250
1170 PRINT D$;"OPEN ";FL$
1180 PRINT D$;"DELETE ";FL$
1190 PRINT D$;"OPEN ";FL$
1200 PRINT D$;"WRITE ";FL$
1210 PRINT OP: PRINT N: PRINT DT$
1220 FOR I = 1 TO N: FOR J = 1 TO 5: PRI
     NT A$(J,I): NEXT J: NEXT I
1230 PRINT D$;"CLOSE ";FL$
1240 GOTO 260
1250 REM ERROR HANDLING ROUTINE
1260 ER = PEEK (222): POKE 216,0
1270 IF ER = 13 THEN PRINT "FILE EXISTS
     BUT IS NOT A TEXT FILE": GOTO 1290
1280 IF ER = 9 THEN PRINT "DISK FULL"
1290 IF ER = 10 THEN PRINT "FILE LOCKED
     ; SAVE WITH A DIFFERENT NAME"
1300 IF ER = 4 THEN PRINT "DISK IS WRIT
     E PROTECTED; REMOVE TAB"
```

APPLE II Series

```
1310 PRINT: PRINT "PRESS <RETURN> TO TR
     Y AGAIN": PRINT "PRESS <ESC> TO
     GO BACK TO MENU": GET K$
1320 IF K$ = CHR$ (13) THEN GOTO 1130
1330 IF K$ = CHR$ (27) THEN GOTO 260
1340 GOTO 1310
1350 REM LOAD DISK FILE
1360 HOME: PRINT "LOAD FROM DISK": PRIN
     T
1370 INPUT "WHAT FILENAME? ";F$
1380 ONERR GOTO 1500
1390 FL$ = F$ + "." + LEFT$ (T1$,3)
1400 PRINT D$;"LOAD ";FL$: REM FORCE E
     RROR
1410 PRINT D$;"OPEN ";FL$
1420 PRINT D$;"READ ";FL$
1430 INPUT OP: INPUT N: INPUT DT$: FOR I
     = 1 TO N: FOR J = 1 TO 5: INPUT A$
     (J,I): NEXT J: NEXT I
1440 PRINT D$;"CLOSE";FL$
1450 VTAB 8: PRINT "FILE ";F$;" LOADED
1460 PRINT "IT WAS SAVED ON ";DT$
1470 PRINT: PRINT "PRESS RETURN TO GO B
     ACK TO MENU"
1480 GET K$: IF K$ < > CHR$ (13) THEN
     1480
1490 GOTO 260
1500 REM ERROR HANDLING ROUTINE
1510 ER = PEEK (222): POKE 216,0
1520 IF ER = 13 THEN GOTO 1410: REM
     FILE EXISTS
1530 PRINT
1540 IF ER = 6 THEN PRINT "EITHER FILE
     NOT FOUND OR WRONG TYPE": PRINT "D
     O YOU WANT A CATALOG OF THE DISK? Y
     /N": GET K$: IF K$ = "Y" THEN PRIN
     T CHR$ (13)D$;"CATALOG"
1550 IF ER = 8 THEN PRINT "DRIVE NOT R
     EADY"
1560 IF ER = 5 THEN PRINT "FILE NOT PR
     OPERLY SAVED FROM BEFORE"
1570 PRINT: PRINT "PRESS <RETURN> TO TR
     Y AGAIN
     PRESS <ESC> TO
     GO BACK TO MENU": GET K$
1580 IF K$ = CHR$ (13) THEN GOTO 1360
1590 IF K$ = CHR$ (27) THEN GOTO 260
1600 GOTO 1570
1610 REM LIST FILE TO PRINTER
1620 HOME: PRINT "SEND DATA TO PRINTER"
1630 PRINT: INPUT "WHAT SLOT # IS YOUR
     PRINTER?";SN: IF SN < 1 OR SN > 6
     THEN 1620
1640 PRINT: PRINT "WHAT DATE DO YOU WAN
     TO LISTING?": INPUT DN$
1650 IF DN$ < > "" THEN DT$ = DN$
```

COMMODORE 64

```
2210 FOR I=1 TO 6:C$(I)=MID$(TI$,I,1):NE
     XT
2220 PRINT "■HOME■22CRSRDOWN■16CRSRRIG
     HT■";
2230 PRINT C$(1);C$(2);":";C$(3);C$(4);"
     :";C$(5);C$(6)
2240 FOR I=1 TO 250:NEXT
2250 I=PEEK(197):IF I=64 THEN 2210
2260 IF I=17 THEN 1970
2270 RETURN
2280 REM COMPLETE LISTING ON PRINTER
2290 OPEN 4,4
2300 FOR I=1TON:FOR J=1TO5
2310 PRINT#4,A$(J,I):NEXT
2320 PRINT#4,"":PRINT#4,"":NEXT
2330 PRINT#4,:CLOSE4:RETURN
2340 PRINT "■SHIFT CLR■2CRSRDOWN■CRSRR
     IGHT■DO YOU WISH TO HALT THE PROGRA
     M"
2350 PRINT "■CRSRRIGHT■AND LOSE ALL THE
     DATA IN MEMORY?"
2360 PRINT "■CRSRRIGHT■CRSRDOWN■PRESS Y
     OR N."
2370 GOSUB 2530
2380 IF K=78 THEN 710
2390 IF K<>89 THEN RETURN
2400 END
2410 FD=INT(F1/.06097):H1=INT(FD/256):L1
     =FD-(256*H1)
2420 FD=INT(F2/.06097):H2=INT(FD/256):L2
     =FD-(256*H2)
2430 POKE S+5,0:POKE S+12,0
2440 POKE S+6,224:POKE S+13,240
2450 POKE S+24,31
2460 POKE S+1,H1:POKE S,L1
2470 POKE S+8,H2:POKE S+7,L2
2480 POKE S+4,33:POKE S+11,33
2490 FOR ZX=1TO100:NEXT
2500 POKE S+4,32:POKE S+11,32
2510 POKE S+24,0:RETURN
2520 PRINT "■CRSRDOWN■7CRSRRIGHT■PRESS
     ANY KEY TO CONTINUE"
2530 GET K$:IF K$<>"" THEN 2530
2540 GET K$:IF K$="" THEN 2540
2550 K=ASC(K$):RETURN
2560 PRINT"■4SHIFT CRSRUP■";:FOR I=1TO2:
     PRINT":NEXT:RETURN
```

HCM

APPLE II Series

```
1660 PR# SN
1670 HTAB 15: PRINT TI$;" AS OF ";DT$
1680 VTAB 3: FOR I = 1 TO N
1690 FOR J = 1 TO 5: PRINT C$(J,OP);": "
     ;A$(J,I): NEXT J
1700 PRINT : NEXT I
1710 PR# 0: RETURN
1720 REM USE DATA
1730 HOME : IF OP = 2 THEN 1910
1740 PRINT " TELEPHONE TIMER": PRINT "
     ******************": GOSUB 950
1750 VTAB 15: PRINT " PRESS ANY KEY TO S
     TART "
1760 PRINT " PRESS S WHEN READY TO STOP
     TIMER."
1770 GET K$
1780 HTAB 5: VTAB 20: PRINT "TIME: ";"0:
     00": S = 0:M = 0
1790 FOR DE = 1 TO 645: NEXT DE
1800 S = S + 1
1810 IF S = 60 THEN S = 0:M = M + 1
1820 HTAB 5: VTAB 20: PRINT "TIME: ";M;"
     :";: IF S < 10 THEN PRINT "0";S: G
     OTO 1840
1830 FOR J = 1 TO 2: NEXT J: PRINT S
1840 KY = PEEK (-16384): IF KY > 127
     THEN ST = 1:KY = KY - 128: POKE -
     16368,0: GOTO 1860
1850 ST = 0: POKE -16368,0
1860 IF ST = 0 THEN 1790
1870 IF KY < > 83 THEN 1790
1880 PRINT : PRINT " PRESS <RETURN> TO G
     O BACK TO THE MENU"
1890 GET K$: IF K$ < > CHR$ (13) THEN
     GOTO 1890
1900 POKE 34,0: RETURN
1910 HOME : VTAB 3: PRINT "FIGURING TOTA
     L VALUE. . . "
1920 FOR I = 1 TO N:TL = TL + VAL (A$(2
     ,I)): NEXT I
1930 VTAB 7: PRINT "TOTAL VALUE OF INVEN
     TORY IS=";TL: VTAB 9: PRINT "
     PRESS RETURN TO GO BACK TO MENU"
1940 GET K$: IF K$ < > CHR$ (13) THEN
     1940
1950 RETURN
1960 REM END PROGRAM
1970 END
1980 REM PRESS TO GO TO MENU
1990 PRINT : PRINT "PRESS RETURN TO CONT
     INUE"
2000 GET K$: IF K$ < > CHR$ (13) THEN
     2000
2010 RETURN
```

HCM

ELECTRONIC HOME SECRETARY (TI-99/4A) Explanation of the Program

Line Nos.

100-170	Program header.
180-220	Initialize arrays and variables.
230-380	Title screen and option to use either the phone book or inventory.
390-640	Input new data.
650-720	Check file size to make sure there is enough memory.
730-890	Main option menu.
900-910	Branch to the add data routine.
920-1270	Change data routine.
1280-1390	Display all of the records in the file on the screen.
1400-1620	Display one entry on the screen.
1630-1740	Search routine to find a record.
1750-2140	Load file into memory.
2150-2410	Save file to a storage device.
2420-2660	Sort the file in memory by the field selected.
2670-2700	Clear the screen and set the screen color.
2710-2750	Key input routine. Will wait for a key to be pressed.
2760-2890	Calculate and display the total cost of the items in an inventory file.
2900-3270	Routine to find an entry in the phone book and dial the phone by producing the proper frequencies over the TV speaker.
3280-3530	Stop watch routine to time yourself on the phone.
3540-3750	Routine to list the entire contents of the file to the printer.
3760-3810	Routine to safely end the program, giving the user the option to go back to the menu if an option has been selected accidentally.
3820-3830	Read data into the category title array.
3840-3860	Program data for the phone frequencies and the category titles.

TI-99/4A

```
100 REM *********************
110 REM * ELECTRONIC HOME SECRETARY *
120 REM *********************
130 REM BY MALLADI SUBBAIAH
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM TI BASIC
170 REM
180 OPTION BASE 1
190 DIM A$(5,60),O$(5),C$(6),P1(3),P2(3
    )
200 READ P1(1),P1(2),P1(3),P2(1),P2(2),
    P2(3)
210 LSIZE=0
220 OPT=1
230 CALL CLEAR
240 CALL SCREEN(12)
250 PRINT "**********":::"
    *** THE HOME SECRETARY ***":::"
    **********":::::::::
260 PRINT TAB(8);"1. PHONE BOOK":::TAB(8
    );"2. INVENTORY"
270 GOSUB 2730
280 IF KEY=49 THEN 350
290 IF KEY=50 THEN 320
300 PRINT "RE-ENTER 1 OR 2"
310 GOTO 270
320 OPT=2
330 RESTORE 3860
340 GOTO 360
350 RESTORE 3850
360 GOSUB 3820
370 N=0
380 GOTO 730
390 REM KEY INPUT FOR DATA SET UP
400 PRINT :" ENTER":" TO EXIT"
    :" TO REENTER"
410 FLAG1=1
420 FLAG2=1
430 I=N+1
440 IF I<=60 THEN 480
450 PRINT :::" *** ARRAY FULL (N=60)***
    ":::
460 GOSUB 2710
470 RETURN
480 PRINT
490 INPUT C$(1):A$(1,I)
500 IF (A$(1,I)="E")+(A$(1,I)="END")THE
    N 2430
510 IF A$(1,I)="" THEN 490
520 IF A$(1,I)<>"R" THEN 570
530 I=I-1
540 N=I-1
550 PRINT :" *** REENTER LAST SET ***"
560 GOTO 480
```

Continued on next page

Secretary . . . from p. 143

TI-99/4A

```
570 FOR Z=2 TO 5
580 INPUT C$(Z):A$(Z,I)
590 NEXT Z
600 GOSUB 650
610 IF T>190 THEN 550
620 GOSUB 680
630 N=I
640 GOTO 430
650 REM MEMORY CHECK
660 T=LEN(A$(1,I)&A$(2,I)&A$(3,I)&A$(4,
    I)&A$(5,I))
670 RETURN
680 LSIZE=LSIZE+T
690 IF LSIZE<3300 THEN 720
700 CALL SOUND(200,800,4)
710 PRINT :"** WARNING ** MEMORY GETTIN
    G FULL":" (LSIZE=";STR$(LSIZE)&"/3
    800)":
720 RETURN
730 SC=4
740 GOSUB 2670
750 PRINT "** HOME SECRETARY **"
760 IF OPT=2 THEN 790
770 PRINT TAB(10);"PHONE BOOK"::
780 GOTO 800
790 PRINT TAB(10);"INVENTORY"::
800 PRINT "**********************
    PRESS":::" 1. ADD DATA":" 2. CHAN
    GE DATA":::" 3. DISPLAY NAME LIST":
    :
810 PRINT " 4. DISPLAY ONE ENTRY"::"
    5. USE THE DATA":::" 6. SAVE DATA F
    ILE":::" 7. LOAD DATA FILE"::
820 PRINT " 8. LIST TO PRINTER":::" 9.
    END PROGRAM"
830 IF FLAG1=0 THEN 840
840 GOSUB 2730
850 IF KEY<49 THEN 730
860 IF KEY>57 THEN 730
870 GOSUB 2670
880 ON (KEY-48)GOSUB 900,920,1290,1400,
    2760,2150,1750,3540,3760
890 GOTO 730
900 GOSUB 390
910 RETURN
920 REM DATA ALTERATION
930 INPUT "WHICH ONE? ":M$
940 IF M$="" THEN 1270
950 PRINT :::"ENTER":::" NEW DATA AT CUR
    SOR":::" /D/ TO DELETE THE ITEM"::"
    'ENTER' FOR NO CHANGES":
960 GOSUB 1640
970 PRINT ::
980 FLAG1=1
990 I=M
1000 GOSUB 650
1010 T=-T
1020 GOSUB 680
1030 INPUT A$(1,M)&"? ":P$
1040 IF P$="" THEN 1140
1050 IF P$<>"/D/" THEN 1120
1060 FOR Z=1 TO 5
1070 A$(Z,M)="--"
1080 NEXT Z
1090 GOSUB 2430
1100 N=N-1
1110 RETURN
1120 A$(1,M)=P$
1130 FLAG2=1
1140 FOR Z=2 TO 5
1150 INPUT A$(Z,M)&" ":P$
1160 IF P$="" THEN 1190
1170 A$(Z,M)=P$
1180 FLAG2=FLAG2+1
1190 NEXT Z
1200 GOSUB 650
1210 IF T<192 THEN 1240
1220 PRINT :"** REENTER LAST SET **"
1230 GOSUB 1030
1240 GOSUB 680
1250 GOSUB 1490
1260 ON T GOTO 1030,930,1270
1270 RETURN
1280 REM DISPLAY ENTIRE DIRECTORY
1290 IF FLAG2=0 THEN 1310
1300 GOSUB 2430
1310 FOR I=1 TO N
1320 M=28-LEN(A$(2,I))
1330 T$=STR$(I)&". "
1340 PRINT TAB(4-LEN(T$));T$;A$(1,I);TAB
     (M);A$(2,I)
1350 IF INT(I/10)<>I/10 THEN 1370
1360 GOSUB 2710
1370 NEXT I
1380 GOSUB 2710
1390 RETURN
1400 REM SINGLE ITEM LISTING
1410 INPUT "WHICH ONE? ":M$
1420 IF M$="" THEN 1480
1430 GOSUB 1640
1440 GOSUB 2670
1450 PRINT A$(1,M)::A$(2,M)::A$(3,M)::A$
     (4,M)::A$(5,M)::
1460 GOSUB 1490
1470 ON T GOTO 1440,1410,1480
1480 RETURN
```

TI-99/4A

```
1490 PRINT :::"PRESS":::"  E FOR PREVIOUS R
     ECORD":::"  X FOR NEXT RECORD":::"  F
     TO FIND ANOTHER RECORD"
1500 GOSUB 2710
1510 T=3
1520 IF KEY<>69 THEN 1560
1530 T=1
1540 IF M=1 THEN 1620
1550 M=M-1
1560 IF KEY<>88 THEN 1600
1570 T=1
1580 IF M=N THEN 1620
1590 M=M+1
1600 IF KEY<>70 THEN 1620
1610 T=2
1620 RETURN
1630 REM SEARCH ROUTINE FOR SINGLE ITEM
     LISTING
1640 IF ABS(ASC(M$)-53)>4 THEN 1690
1650 M=VAL(M$)
1660 IF M<=N THEN 1680
1670 M=N
1680 RETURN
1690 FOR I=1 TO N
1700 M=I
1710 IF LEN(M$)>LEN(A$(1,I))THEN 1730
1720 IF M$=SEG$(A$(1,I),1,LEN(M$))THEN 1
     740
1730 NEXT I
1740 RETURN
1750 REM LOAD DATA
1760 PRINT "ENTER":::"1. CS1":::"2. DISK1"
     :::"3. OTHER"
1770 INPUT DEV
1780 IF DEV<>1 THEN 1810
1790 S$="CS1"
1800 GOTO 1860
1810 IF DEV<>2 THEN 1850
1820 INPUT "ENTER FILE NAME: DSK1.":F$
1830 S$="DSK1."&F$
1840 GOTO 1860
1850 INPUT "ENTER DEVICE NAME:":S$
1860 OPEN #2:S$,INTERNAL,INPUT ,FIXED 19
     2
1870 INPUT #2:O,NT,F$,D$,LSIZE
1880 IF O=OPT THEN 1980
1890 IF O=2 THEN 1920
1900 PRINT :::"FILE IS A PHONE BOOK TYPE"
     :::
1910 GOTO 1930
1920 PRINT :::"FILE IS AN INVENTORY TYPE"
     :::
1930 PRINT "PROCEDE (Y/N)?"
1940 GOSUB 2730
1950 IF KEY<>78 THEN 1970
1960 RETURN
1970 IF KEY<>89 THEN 1940
1980 OPT=O
1990 N=NT
2000 IF OPT=1 THEN 2040
2010 RESTORE 3860
2020 GOSUB 3820
2030 GOTO 2060
2040 RESTORE 3850
2050 GOSUB 3820
2060 PRINT :::F$:::"LSIZE(3800)=";LSIZE:::"
     LAST UPDATE: ";D$:::
2070 FOR I=1 TO N
2080 INPUT #2:A$(1,I),A$(2,I),A$(3,I),A$
     (4,I),A$(5,I)
2090 NEXT I
2100 IF DEV=1 THEN 2130
2110 FOR TD=1 TO 1000
2120 NEXT TD
2130 CLOSE #2
2140 RETURN
2150 REM SAVE DIRECTORY
2160 IF FLAG2=0 THEN 2180
2170 GOSUB 2430
2180 PRINT "ENTER 1. CS1"
2190 PRINT "  2. DSK1"
2200 PRINT "  3. OTHER":::::
2210 INPUT "YOUR CHOICE?":ANS
2220 IF (ANS<1)+(ANS>3)THEN 2180
2230 ON ANS GOTO 2240,2260,2320
2240 S$="CS1"
2250 GOTO 2330
2260 INPUT "ENTER FILE NAME: DSK1.":N$
2270 IF LEN(N$)<11 THEN 2300
2280 PRINT "ENTER NO MORE THAN TEN
     LETTERS PLEASE."
2290 GOTO 2180
2300 S$="DSK1."&N$
2310 GOTO 2330
2320 INPUT "ENTER DEVICE NAME.":S$
2330 INPUT "ENTER DATE.":D$
2340 OPEN #3:S$,INTERNAL,OUTPUT,FIXED 19
     2
2350 PRINT #3:OPT,N,S$,D$,LSIZE
2360 FOR I=1 TO N
2370 PRINT #3:A$(1,I),A$(2,I),A$(3,I),A$
     (4,I),A$(5,I)
2380 NEXT I
2390 CLOSE #3
2400 FLAG1=0
2410 RETURN
2420 REM SORTING ROUTINE
```

Continued on p. 148

HCM CLASSIFIEDS
RATES AND INFORMATION

The *HCM* Classifieds is an economical alternative to general display advertising. All listings appear in a standard format, typeset by *HCM* and are published in the first available issue after receipt.

RATES: $175 each ad, per issue

AD FORMAT: 1 line heading (25 characters maxium)
8 lines body copy (40 characters per line)
4 lines for company name, address and telephone

ORDERING: Send listing information with payment* to:
HCM Classifieds,
1500 Valley River Drive, Suite 250
Eugene, Oregon 97401

*Check, Money Order, VISA or MasterCard are accepted.

THE BEST SOFTWARE BARGAIN!
A complete library of TI-99/4A cassette programs ONLY 50¢EACH!! Games-Educational-Demonstration-Informational/Useful programs.
Package #1 - 100 programs at $49.95.
Package #2 - 120 programs at $59.95.
Package #3 - 140 programs at $69.95.
Free catalog. VISA/MasterCard/CHOICE welcome.
Great software at ROCK-BOTTOM PRICES.
THE EXCHANGE, HCM-1
P.O. Box 46
Sterling, VA 22170
(703) 435-2789

TI-99/4A OWNERS
Two player joystick, ex. basic game. All games are $20 ea. for tape and $25 ea. for disk. Ex. battleship: hunt down and blow enemy ships out of the water. World War IV tanks: using telelaser guns, 20 tanks vs. 14. World War IV tanks Part 2. 40 tanks vs. 10. 30-day replacement warranty.
Send check or money order to:
GOODTIME (99)
P.O. Box 19802
Milwaukee, WI. 53219

YOUR TI-99/4A SPECIALIST
True standard IEEE cards by Cor-Comp with 1 year warranty. Includes: RS-232 card for peripheral expansion box, 2 serial ports and 1 parallel port, only $119.95. Also RS-232 Stand Alone, $139.95. And 32 K cards for expansion box, $149.95. We're adding new products for your TI-99/4A.
BAY CITY ELECTRONICS
624 Woodside Road
Redwood City, CA 94061
(415) 367-1880

FREE GAME FOR TI 99/4A
with ANY purchase. Games in Basic or X-Basic at $7. Additional games $4. Choose from: Alien Attack, Space Tunnel, Freeway Frogs, Cosmic Battle, Word Search + many more. Cassettes or disk ($2 extra) Powerful Word Processor for Apple II+, IIe and Commodore 64 ....(disk) $29.
FREE CATALOG AVAILABLE
BIT BYTE BIT SOFTWARE, INC.
P.O. Box 565, Coram NY 11727
516-928-6538 ($1.50 postage)

QUALITY TI-99/4A SOFTWARE
For home, school and business featuring *Create-a-File* menu driven database management system with user search/sort.
On disk $39.95.
Teachers Record II grading and storage program for up to 40 names and 48 games.
On disk $44.95.
Send $1 for catalog. Refundable.
ARRO-SOFT SYSTEMS
P.O. Box 1761
Edmond, OK 73083
(405)341-8567

CHECKBOOK DATABASE
For TI-99/4A with Cassette & Minimemory. List and total Checks by payee name, by dates or check numbers (from-to), or if Tax deductible. List and total Deposits, Interest, and Bank Charges by dates. 400 Checks, 53 Deposits per year. Database saved on tape. Minimum keystroke data entry. User Manuel. Send $15 check to:
FINELINE SOFTWARE
3 Wendy Lane
Marlboro, N.J. 07746

TEACHERS AND PARENTS
TI software developed by teachers. Use in home or school. Classroom tested. Program for States or Capitals (XB, DSK, graphics, 4th-adult) $15 ea. Spell game (DSK, CS, 1st-8th) $10. Also print or non-print programs for grade book, charts, notices, etc. Will custom write per request for school or Indiv. For info or order (check or m.o.):
LANDSCAPE LTD.
P.O. Box 8424
Bartlett, IL 60103
Allow 4-6 weeks del.

NEW! VIDEO CHESS/CHECKERS
For those who enjoy playing Checkers or Chess these TI-99/4A cassette programs let 2 players compete on TV. Has superior full screen color board and piece graphics over other games. An excellent learning aid with new game or problem setup mode designed into the chess program. $7 ea. or $12 for both. Postpd.
VIDCOR
P.O. Box 3923
Wichita, Kansas 67201

NEW SOFTWARE FOR TI 99/4A
Home Account Library $19.95. For easy filing of all monthly expenses. Auto name carry over. Simple entry and retrieval. Screen Dump $19.95. Reproduce screen contents on TI printer. 300 or 4800 BD option. Both programs require disk drive, ext. basic and ext. memory. All post paid. Send check or money order to:
SBC
24 Sandpiper Road
Enfield, CT 06082

SPECIAL OFFER FROM SPC
Latest version TI FORTH; disk and documentation $25. Advanced DATABASE Management program and Form Letter Utility for TI. High speed data processor uses "Virtual Memory" concept. Does EVERYTHING at high speed and is extremely flexible. For details on this and more:
SPC SOFTWARE COMPANY
P.O. Box 121
Brightwaters, NY 11718
(516) 587-5462

IF YOU DARE!
IF YOU DARE is an exciting game unlike any you have seen before. Fun for 2 or great at parties, it's a hilarious good time for up to 8 players. Three levels to suit your taste. Receive commands to perform or pay the penalty for passing. On tape-TI/99 (TI BASIC) or C 64 $24.95
Illinois residents add 6% sales tax.
PARAGON SYSTEMS
P.O. Box 893
Hillside, IL 60162
Allow 3-4 weeks for delivery.

2 PROGRAMS FOR THE 99/4A
Expense record keeping-XBAS, mem exp req. store & process up to 7500 individual income & expen. items, printer opt. Ideal for personal bookkeeping, SML business. Easier than a spreadsheet, $25 per disk. Racquetball plus - assembly language.XBAS or ED/ASM & memory expansion req. 1-4 plyrs(jystks req for 4) $9 per disk. Send check or money order to:
KRONITON SOFTWARE
4007 ANNIE STREET
ALEXANDRIA, LOUISIANNA 71301

Spellakazam . . . from p. 122

try to follow the bouncing Magician instead of trying to figure out how to spell the word all by yourself.

Because of loading difficulties (Commodore version only), customizing features, and the various ways the game can be played, parents and educators will want to supervise the use of *Spellakazam*.

Graphics

Many educational games have merely cosmetic graphics...not *Spellakazam*. The screen has the appearance of a board game with its bold letters, colorful barriers and little characters that represent you, the spaces in words, and apostrophes. The magic birds that send the magician back to the start are placed to please the eye of artist and strategist alike. The friendliest aspects of the graphics are the repetition of the player's name and the animated reward graphics that pop out of the magic hat.

At Home With Spelling

Whether playing in a living room or classroom learning center, with their own list or the game's built-in list, children will enjoy practicing their spelling, reading, and reasoning skills using *Spellakazam*.

HCM

Adventures . . . from p. 115

graphics awaiting those who successfully escape from the castle.

Ease Of Use

Most users will be able to play the game immediately without consulting the folding-card documentation. The card is useful for quick reference, particularly the Things To Remember list and the list of what the function keys do. For the most part though, the on-screen prompts are sufficient to guide you through the game.

Users who are good at solving math problems in their heads might be comfortable staying with the default option of entering the answers left to right. However, to make the software easier to use and more consistent with the training most youngsters receive in math, parents might want to use the function keys to make the entry order of the answer right to left. We noticed that the adults who played this game liked to give their mental math muscles more exercise by solving the problems left to right.

In School Or At Home

Adventures in Math is an educational game that teachers will use with students during the day and take home for their own children at night. Families can enjoy playing it together, and it's a good game to fill rainy days during Spring Break. Because it is not geared specifically to male or female interests, it could make math more attractive to more children. We'd like to see more of this kind of game because *Adventures in Math*, and games like it, will provide not hours but years of "edutainment" for youngsters. HCM

MAKE SOMEONE HAPPY!

AND GET FREE SOFTWARE TO BOOT!

Give a gift-subscription to

HOME COMPUTER™ magazine

And we'll send *YOU* 2 issues of ON TAPE™ or ON DISK™ ABSOLUTELY FREE!

Comments From Our Happy Readers

"I purchased my computer without knowing about you, but after seeing many other "computer" magazines, you are light years ahead of the pack."
Tim Williams, Gig Harbor, WA

"...your publication is better than Compute, Byte, Popular Computing and Personal Computing. We have cancelled our other subscriptions..."

This is the first time I've ever written a magazine publication and it's only because I'm impressed with your magazine."
Cindy Eckhardt, Palatka, FL

"...I had considered BYTE the most useful of the genre until I saw yours. Absolutely the most valuable source of ideas..."
Dennis E. Clouse, Concord, CA

"I am a programmer/analyst and I just subscribed to your magazine.

The question is where have you been? I was looking for just the magazine you and your staff publish. I must say it is the best. I have subscribed to Byte, Softside, Creative Computing and Kilobaud, and nothing—can compare!!"
Sal Melillo, Scotch Plains, NJ

"...Your magazine is a help as well as a great source for software games, books, and hardware. For me the super-novice, this information is invaluable! Keep up the good work!"
David Hurd, Lubbock, TX

"...After spending a small fortune subscribing to computer magazines...the investment has finally realized some fantastic gains with your magazine."
Robert L. Nelson, Suring, WI

FREE software—ON TAPE™ or ON DISK™ —the same high-quality programs published monthly in the magazine. This cassette tape or floppy disk program service—normally a $3.95 per month extra cost—is the convenient, accurate and affordable way to save hundreds of typing hours.

Thanks to your thoughtfulness. . . your friends, family, and associates can enjoy a gift that keeps on giving all-year-through! They'll enjoy **12 BIG issues** of *HCM* delivered right to their door each and every month. . . **AND YOU** will receive **2 months** of our magazine program service—ON TAPE™ or ON DISK™ —**FREE!!**

PLEASE ENTER A 1-YEAR GIFT-SUBSCRIPTION FOR:

Name _____

Address _____

City _____ State _____ Zip _____

Does recipient have a home computer?

☐No, not yet

☐Yes — What Brand? _____

FROM: (Gift-giver must complete this section)

Name _____

Address _____

City _____ State _____ Zip _____

Subscriber label code number of gift giver

SATISFACTION GUARANTEED

—or the unfilled portion of the subscription will be refunded, less the cost of any premium items you have received.

FREE SOFTWARE FOR GIFT-GIVERS

Indicate your choice of media: ☐ON TAPE™ ☐ON DISK™

Indicate which computer media is for:(pick one)

☐Apple ☐C-64 ☐VIC-20 ☐IBM PC ☐IBM PCjr ☐TI-99/4A

Select your 2 FREE software issues

☐Vol. 4, No. 1 ☐Vol. 4, No. 2 ☐Vol. 4, No. 3

☐Vol. 4, No. 4 ☐Vol. 4, No. 5

☐Enclosed is $25 for each 1-year gift-subscription

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription only. Free software not available outside USA.

☐Check or Money Order Enclosed

MUST BE IN US FUNDS DRAWN ON A US BANK

Bill my ☐VISA ☐MasterCard

Account No. _____

Signature _____ Exp. Date _____

Enclose payment or credit card information and mail with completed form to:

Home Computer Magazine
P.O. Box 5537 • Eugene, OR 97405

Or use our TOLL-FREE ORDER LINE
FOR VISA/MasterCard orders only:
1-800-828-2212
In Oregon, Alaska, Hawaii Tel. (503)485-8796
Allow 6-8 weeks for your first issue.

Jr. Sounds Off . . . from p. 127

selected a register, you can output the frequency bits to port 193. You can think of the six bits going to port 193 as the rough adjustment and the four bits included in the output to port 192 as the fine adjustment. Because the volume is constant at all times in this loop, output values that adjust the volume were not used.

In line 330 the volume for all three voices is set to 15 (value= 159, 191, and 223), which is off. Notice that outputs to port 193 were not necessary because all of the volume control bits are contained within the four low-order bits of port 192.

The loop that runs from line 340 through line 370 ramps the volume for the noise from the maximum volume to a volume of eight (decimal 248 here), about halfway through the scale.

The last loop between lines 390 and 420 assigns a random source to the noise output with the volume set to zero (loudest) in line 380. Line 400 generates a random number between 0 and 7 and adds it to 224, which is the beginning value for selecting the noise source.

When the last loop has finished, the volume is turned off for the noise channel. A time delay loop then waits a few seconds before returning to line 200 to repeat the whole process over again.

SOUND CHIP ACCESS (IBM PCjr) Explanation of the Program

Line Nos.	
100-190	Program header.
200	Set up the sound multiplexer to accept its source from the TI complex sound generator.
210	Set attenuation for voice #1 to 1.
220	Set attenuation for voice #2 to 1.
230	Set attenuation for voice #3 to 1.
240	Set the noise channel up with a periodic noise. This is equivalent to a value of 1 as the source when using the NOISE command in Cartridge BASIC.
250-320	Loop to control first scale of sounds.
260-270	Set the frequency of the first voice.
280-290	Set the frequency of the second voice.
300-310	Set the frequency of the third voice.
330	Turns off the volume for all three voices.
340-370	This loop will ramp the volume of the noise channel from a maximum (240 = volume 0) to mid scale (248 = volume 8).
380	Set the volume for the noise channel to the maximum.
390-420	This loop will randomly select different noise sources, causing a ricochet sound.
430	Turn the noise channel's volume off.
440	Time delay before restarting the sounds.
450	Branch back to the beginning and do it again.

IBM PCjr

```
100 REM  **********************
110 REM  *  SOUND  CHIP  ACCESS  *
120 REM  **********************
130 REM  BY WILLIAM K.  BALTHROP
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  IBM PCjr
170 REM  CASSETTE BASIC OR
180 REM  CARTRIDGE BASIC
190 REM
200 OUT 97,124
210 OUT 192,145
220 OUT 192,177
230 OUT 192,209
240 OUT 192,226
250 FOR X=0 TO 32 STEP .5
260 OUT 192,128
270 OUT 193,X
280 OUT 192,160
290 OUT 193,X*2
300 OUT 192,192
310 OUT 193,64-X*2
320 NEXT X
330 OUT 192,159:OUT 192,191:OUT 192,223
340 FOR X=240 TO 248
350 OUT 192,X
360 FOR TD=1 TO 15:NEXT TD
370 NEXT X
380 OUT 192,240
390 FOR X=1 TO 20
400 OUT 192,INT(RND*8)+224
410 FOR TD=1 TO 20:NEXT TD
420 NEXT X
430 OUT 192,255
440 FOR TD=1 TO 1000:NEXT TD
450 GOTO 200
```

HCM

Secretary . . . from p. 145

TI-99/4A

```
2430 PRINT "SORT BY WHICH FIELD:"
2440 FOR Z=1 TO 5
2450 PRINT Z;". ";C$(Z)
2460 NEXT Z
2470 INPUT F
2480 IF (F<1)+(F>5)THEN 2430
2490 FLAG2=0
2500 CALL SOUND(100,800,6)
2510 PRINT ::::" ***** SORTING DATA *****
     ":::
2520 FOR I=1 TO N-1
2530 FOR J=N TO I+1 STEP -1
2540 IF A$(F,I)<=A$(F,J)THEN 2560
2550 GOSUB 2610
2560 NEXT J
2570 IF A$(F,I)<>A$(F,I+1)THEN 2590
2580 I=I+1
2590 NEXT I
2600 RETURN
2610 FOR Z=1 TO 5
2620 O$(Z)=A$(Z,I)
2630 A$(Z,I)=A$(Z,J)
2640 A$(Z,J)=O$(Z)
2650 NEXT Z
2660 RETURN
2670 REM  CLEAR & SET SCREEN
2680 CALL CLEAR
2690 CALL SCREEN(SC)
2700 RETURN
2710 REM  KEY RETURN
2720 PRINT ::"PRESS ANY KEY TO CONTINUE"
2730 CALL KEY(0,KEY,STATUS)
2740 IF STATUS=0 THEN 2730
2750 RETURN
2760 REM  UTILITY PROGRAMS
2770 IF OPT=1 THEN 2920
2780 SUM=0
2790 FOR I=1 TO N
2800 SUM=SUM+VAL(A$(2,I))
2810 NEXT I
2820 PRINT "TOTAL COST OF ALL THE ITEMS"
     :::
2830 PRINT :::::TAB(10);SUM:::::::::
2840 CALL HCHAR(11,7,36,18)
2850 CALL HCHAR(19,7,36,18)
2860 CALL VCHAR(12,7,36,7)
2870 CALL VCHAR(12,24,36,7)
2880 GOSUB 2710
2890 RETURN
2900 REM  DIAL PHONE
2910 REM  CLOCK TIME DELAYS FOLLOW
2920 GOSUB 1410
2930 IF M$="" THEN 3180
2940 CALL CLEAR
2950 H=23
2960 V=8
2970 T$=A$(2,M)
2980 L=LEN(T$)
2990 PRINT A$(1,M):A$(2,M):A$(3,M):A$(4,
     M):A$(5,M):::
3000 IF L<10 THEN 3030
3010 L=L+1
3020 T$="1"&T$
3030 FOR J=1 TO L
3040 P$=SEG$(T$,J,1)
3050 CALL HCHAR(H,V,ASC(P$),1)
3060 V=V+1
3070 IF ASC(P$)<48 THEN 3170
3080 IF ASC(P$)>57 THEN 3170
3090 T=VAL(P$)
3100 IF T<>0 THEN 3130
3110 CALL SOUND(300,941,0,1336,2)
3120 GOTO 3160
3130 I=INT((T-1)/3)+1
3140 IJ=T-3*(I-1)
3150 CALL SOUND(300,P1(I),0,P2(IJ),2)
3160 CALL SOUND(250,44000,29)
3170 NEXT J
3180 PRINT ::"PRESS ":"          R TO REDIAL
     ":"          S TO START STOPWATCH":"
          N FOR NEW NUMBER"
3190 GOSUB 2710
3200 IF KEY=82 THEN 2940
3210 IF KEY=78 THEN 2920
3220 IF KEY<>83 THEN 3270
3230 CALL HCHAR(14,1,32,320)
3240 PRINT ::"HOLD DOWN ":"R TO DIAL AGAI
     N":"ANY KEY TO RETURN TO MENU":::
3250 GOSUB 3290
3260 IF T=82 THEN 2920
3270 RETURN
3280 REM  STOP WATCH
3290 MN=0
3300 SEC=0
3310 CALL HCHAR(23,25,58)
3320 CALL SOUND(5,660,0)
3330 CALL SOUND(900,44000,30)
3340 IF MN>9 THEN 3380
3350 CALL HCHAR(23,23,48)
3360 CALL HCHAR(23,24,ASC(STR$(MN)))
3370 GOTO 3400
3380 CALL HCHAR(23,23,ASC(STR$(MN)))
3390 CALL HCHAR(23,24,ASC(SEG$(STR$(MN),
     2,1)))
3400 IF SEC>9 THEN 3440
3410 CALL HCHAR(23,26,48)
3420 CALL HCHAR(23,27,ASC(STR$(SEC)))
3430 GOTO 3460
```

Secretary . . .

TI-99/4A

```
3440  CALL HCHAR(23,26,ASC(STR$(SEC)))
3450  CALL HCHAR(23,27,ASC(SEG$(STR$(SEC)
      ,2,1)))
3460  SEC=SEC+1
3470  IF SEC<60 THEN 3500
3480  SEC=0
3490  MN=MN+1
3500  CALL KEY(0,K,S)
3510  IF S=0 THEN 3320
3520  IF K=82 THEN 2940
3530  RETURN
3540  REM COMPLETE LISTING ON PRINTER
3550  ST$=PS$
3560  M$="ENTER DEVICE NAME: "&P
      S$
3570  INPUT M$:PS$
3580  IF PS$<>"" THEN 3610
3590  PS$=ST$
3600  IF PS$="" THEN 3750
3610  PRINT "IS ";PS$;" READY?(Y/N)"
3620  GOSUB 2730
3630  IF KEY<>89 THEN 3750
3640  OPEN #3:PS$,OUTPUT
3650  T=INT(N/9)
3660  FOR J=0 TO T
3670  FOR M=1 TO 9
```

TI-99/4A

```
3680  I=9*J+M
3690  PRINT #3:TAB(5);I;")";TAB(10);A$(1,
      I);TAB(10);A$(2,I);TAB(10);A$(3,I);
      TAB(10);A$(4,I);TAB(10);A$(5,I)::
3700  IF I=N THEN 3740
3710  NEXT M
3720  GOSUB 2710
3730  NEXT J
3740  CLOSE #3
3750  RETURN
3760  PRINT "DO YOU WISH TO HALT THE
      PROGRAM AND LOSE ALL DATA INMEMORY?
      (Y/N)"
3770  CALL KEY(0,K,S)
3780  IF S=0 THEN 3770
3790  IF K=ASC("N") THEN 730
3800  IF K<>ASC("Y") THEN 3750
3810  END
3820  READ C$(1),C$(2),C$(3),C$(4),C$(5)
3830  RETURN
3840  DATA 697,770,852,1210,1340,1481
3850  DATA NAME:,PHONE:,ADDRESS:,CITY:,ST
      ATE&ZIP:
3860  DATA ITEM:,VALUE:,LOCATION:,DATE
      PURCHASED:,SER # OR ID:
```

HCM

Mystery Words . . . from p. 55

IBM PCjr

```
620   AN$=A$:HP=5:FOR NT=1 TO LEN(A$):HP=
      HP+4:ON ASC(MID$(A$,NT,1))-64 GOSUB
      630,640,650,660,670,680,690:NEXT N
      T:RETURN
630   IF CLEF=1 THEN Z=45:N$="A":OCT=2:GO
      TO 700 ELSE IF CLEF=2 THEN Z1=INT(R
      ND*2):Z=45+(Z1+1)*35:N$="A":OCT=(1-
      Z1):GOTO 700 ELSE Z1=INT(RND*3):Z=4
      5+(Z1*35):N$="A":OCT=(2-Z1):GOTO 70
      0
640   IF CLEF=1 THEN Z=40:N$="B":OCT=2:GO
      TO 700 ELSE IF CLEF=2 THEN Z1=INT(R
      ND*2):Z=40+(Z1+1)*35:N$="B":OCT=(1-
      Z1):GOTO 700 ELSE Z1=INT(RND*3):Z=4
      0+(Z1*35):N$="B":OCT=(2-Z1):GOTO 70
      0
650   IF CLEF=2 THEN Z=105:N$="C":OCT=1:G
      OTO 700 ELSE IF CLEF=1 THEN Z1=INT(
      RND*2):Z=35+(Z1*35):N$="C":OCT=(3-Z
      1):GOTO 700 ELSE Z1=INT(RND*3):Z=35
      +(Z1*35):N$="C":OCT=(3-Z1):GOTO 700
660   IF CLEF=2 THEN Z=100:N$="D":OCT=1:G
      OTO 700 ELSE IF CLEF=1 THEN Z1=INT(
      RND*2):Z=30+(Z1*35):N$="D":OCT=(3-Z
      1):GOTO 700 ELSE Z1=INT(RND*3):Z=30
      +(Z1*35):N$="D":OCT=(3-Z1):GOTO 700
670   IF CLEF=2 THEN Z=95:N$="E":OCT=1:GO
      TO 700 ELSE IF CLEF=1 THEN Z1=INT(R
      ND*2):Z=25+(Z1*35):N$="E":OCT=(3-Z1
      ):GOTO 700 ELSE Z1=INT(RND*3):Z=25+
      (Z1*35):N$="E":OCT=(3-Z1):GOTO 700
```

IBM PCjr

```
680   IF CLEF=2 THEN Z=90:N$="F":OCT=1:GO
      TO 700 ELSE IF CLEF=1 THEN Z1=INT(R
      ND*2):Z=20+(Z1*35):N$="F":OCT=(3-Z1
      ):GOTO 700 ELSE Z1=INT(RND*3):Z=20+
      (Z1*35):N$="F":OCT=(3-Z1):GOTO 700
690   IF CLEF=1 THEN Z=50:N$="G":OCT=2:GO
      TO 700 ELSE IF CLEF=2 THEN Z1=INT(R
      ND*2):Z=50+(Z1+1)*35:N$="G":OCT=(1-
      Z1):GOTO 700 ELSE Z1=INT(RND*3):Z=5
      0+(Z1*35):N$="G":OCT=(2-Z1):GOTO 70
      0
700   Y=HP*8:DRAW "C8;BM=Y;,=Z;XNOTE$;":P
      LAY "O=OCT;XN$;":RETURN
710   CLS:FOR Z=20 TO 60 STEP 10:LINE (0,
      Z)-(319,Z),13:NEXT Z:FOR Z=80 TO 12
      0 STEP 10:LINE (0,Z)-(319,Z),13:NEX
      T Z
720   DRAW "C8BM17,52;XTREB$;BM18,52;XTRE
      B$;BM19,522;XTREB$;BM8,90;XBASS$;BM
      9,90;XBASS$;BM10,90;XBASS$;":RETURN
730   LINE (0,127)-(319,199),1,BF:RETURN
740   DATA AGE,ACE,ADD,ABE,BAD,BAA,BAG,BE
      E,BEG,BED,BEA,CAB,CAD,DAB,DAD,DEB,D
      EE,EBB,EGG,FAB,FAD,FED,GAG,GEE,GAB,
      ADDED,DABBED,CAGED,DEFACED,BAGGED
750   DATA AGED,BABE,BEEF,BEAD,CAFE,CAGE,
      DEAD,DEAF,DEED,EDGE,FACE,FEED,GAGE,
      FEE,CAGED,DEFACE,DECADE,CABBAGE,EDG
      ED,BAGGAGE,EBBED,FACED,BADE,ACCEDE,
      AGAGE,BAGDAD,DECA,FACADE,EFFACE,
760   END
```

HCM

Matrix Muncher . . . from p. 101

VIC-20

```
390   PRINT"■CRSRDOWN■NOW INPUT B ELEMENT
      S"
400   FOR I=1TON:I$=STR$(I):P$="B("+MID$(
      I$,2,1)+")"
410   PRINT P$:INPUT B(I)
420   NEXT I
430   PRINT"■SHIFT CLR■"
440   REM INVERT MATRIX A
450   FOR L=1 TO N:PRINT TAB(6)"M U N C H
      "
460   IF Z(L,L)<>0 THEN 480
470   GOSUB 730
480   Z(L,L)=1/Z(L,L)
490   FOR K=1TON
500   IF (K-L)=0 THEN 560
510   Z(K,L)=Z(K,L)*Z(L,L)
520   FOR M=1TON
530   IF (M-L)=0 THEN 550
540   Z(K,M)=Z(K,M)-Z(K,L)*Z(L,M)
550   NEXT M
560   NEXT K
570   FOR M=1TON
580   IF (M-L)=0 THEN 600
590   Z(L,M)=-Z(L,L)*Z(L,M)
600   NEXT M
610   NEXT L
620   PRINT"■2CRSRDOWN■SOLUTION VALUES AR
      E:■CRSRDOWN■"
```

VIC-20

```
630   FOR I=1TON
640   X(I)=0
650   FOR J=1TON
660   X(I)=X(I)+Z(I,J)*B(J)
670   NEXT J
680   I$=STR$(I):P$="  X("+MID$(I$,2,1)+"
      )=":PRINT P$;X(I)
690   NEXT I
700   PRINT""
710   PRINT"**  DONE  **":END
720   REM SUB TO SWITCH ROWS
730   FOR LL=L+1 TO N
740   IF Z(LL,L)=0 THEN 840
750   FOR M=1TON
760   DZ=Z(L,M)
770   Z(L,M)=Z(LL,M)
780   Z(LL,M)=DZ
790   NEXT M
800   DB=B(L)
810   B(L)=B(LL)
820   B(LL)=DB
830   RETURN
840   NEXT LL
850   PRINT"■2CRSRDOWN■SORRY, A DETERMINA
      NT":PRINT"EQUALS 0. THERE IS"
860   PRINT"NO UNIQUE SOLUTION."
870   END
```

HCM

Tablut . . . from p. 61

TABLUT (C-64)
Explanation of the Program

Line Nos.

100-160	Header.
170-230	Move character set from ROM to RAM.
240-310	Load programmable character definitions.
320-330	Initialize Extended Background Color mode.
340	Initialization.
350-420	Load initial gameboard matrix.
430-480	Display header screen.
490-510	Display gameboard.
520-610	Main program loop.
620-820	Routine to see if move is valid.
830-860	Subroutine to display gameboard.
870-1060	Subroutines to move different pieces around gameboard.
1070-1110	Illegal move subroutine.
1120-1190	Subroutine for king's escape.
1200-1250	Subroutine for king's capture.
1260-1500	Subroutines to get coordinates from players.
1510-1650	Subroutine to see if white captures red.
1660-1800	Subroutine to see if red captures white.
1810-1940	Subroutine to see if king has escaped or been captured.

COMMODORE 64

```
100 REM  *********
110 REM  * TABLUT *
120 REM  *********
130 REM  BY JAMES J. MULLIGAN AND THE HC
    M STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.2.1
160 REM  C64 BASIC
170 POKE 52,56:POKE 56,56:CLR:PRINT"■SH
    IFT CLR■"
180 POKE 56334,PEEK(56334)AND254
190 POKE 1,PEEK(1)AND251
200 FOR I=0 TO 511:POKE I+14336,PEEK(I+
    53248):NEXT
210 POKE 1,PEEK(1)OR4
220 POKE 56334,PEEK(56334)OR1
230 POKE 53272,(PEEK(53272)AND240)OR14:
    POKE 53281,1
240 FOR I=14600 TO 14600+(12*8)-1
250 READ A:POKE I,A:NEXT
260 DATA 0,4,14,31,31,14,15,0,0,8,8,
    8,8,8,248
270 DATA 15,63,63,63,25,25,57,0,8,200,2
    00,200,136,136,200,0
280 DATA 0,0,0,33,51,59,63,63,0,0,0,132
    ,204,220,252,252
290 DATA 0,63,63,63,0,0,0,0,0,252,252,252
    ,0,0,0,0
300 DATA 0,127,127,127,127,127,127,127,
    0,254,254,254,254,254,254,254
310 DATA 127,127,127,127,127,127,127,0,
    254,254,254,254,254,254,254,0
320 POKE 53281,14:POKE 53282,1:POKE 532
    83,0
330 POKE 53265,PEEK(53265)OR64
340 DIM PA(11,11),SA(11,11):S=54272
350 FOR I=0 TO 80
360 READ A:RW=INT(I/9)+1:CL=I+1-((RW-1)
    *9)
370 SA(RW,CL)=A:NEXT
380 DATA 1,1,1,2,2,2,1,1,1,1,1,1,1,2,1,
    1,1,1
390 DATA 1,1,1,1,3,1,1,1,1,2,1,1,1,3,1,
    1,1,2
400 DATA 2,2,3,3,4,3,3,2,2,2,1,1,1,3,1,
    1,1,2
410 DATA 1,1,1,1,3,1,1,1,1,1,1,1,1,2,1,
    1,1,1
420 DATA 1,1,1,2,2,2,1,1,1
430 PRINT"■SHIFT CLR■":PRINT"■CTRL WHT
    ■10CRSRDOWN■"
440 PRINT TAB(15)"* * * * * * * * * *"
450 PRINT TAB(15)"*  TABLUT  *"
460 PRINT TAB(15)"* * * * * * * * * *"
470 PRINT:PRINT"■5CRSRRIGHT■< PRE
    SS ANY KEY TO CONTINUE >"
480 Z=PEEK(197):IF Z=64 THEN 480
490 GOSUB 830
500 PRINT "■CTRL WHT■■HOME■■CRSRDOWN■"T
    AB(14)"A B C D E F G H I"
510 FOR I=1TO9:PRINT TAB(32)I:PRINT:NEX
    T
520 CV=0:TN=3:GOSUB 1260
530 GOSUB 620:IF CV<>0 THEN 520
540 KF=0:GOSUB 1660
550 IF KF=1 THEN GOTO 1200
560 IF KF=2 THEN GOTO 1120
570 CV=0:TN=2:GOSUB 1310
580 GOSUB 620:IF CV<>0 THEN 570
590 KF=0:GOSUB 1510
600 IF KF=1 THEN GOTO 1200
610 GOTO 520
```

COMMODORE 64

```
620 IF RF=RT OR CF=CT THEN 640
630 GOSUB 1070:CV=1:GOTO 820
640 IF PA(RT,CT)=1 AND PA(RF,CF)=TN THE
    N 670
650 IF PA(RT,CT)=1 AND TN=3 AND PA(RF,C
    F)=4 THEN 670
660 GOSUB 1070:CV=2:GOTO 820
670 CV=0:IF RF=RT THEN 730
680 Z=SGN(RT-RF):FOR I=RF+Z TO RT STEP
    Z
690 IF PA(I,CF)<>1 OR (PA(RF,CF)<>4 AND
    CF=5 AND I=5) THEN CV=1
700 NEXT
710 IF CV=0 THEN 790
720 GOSUB 1070:CV=3:GOTO 820
730 Z=SGN(CT-CF)
740 FOR I=CF+Z TO CT STEP Z
750 IF PA(RF,I)<>1 OR (PA(RF,CF)<>4 AND
    I=5 AND RT=5) THEN CV=1
760 NEXT
770 IF CV=0 THEN 790
780 GOSUB 1070:CV=4:GOTO 820
790 X=PA(RF,CF):RW=RF:CL=CF:GOSUB 870
800 RW=RT:CL=CT
810 ON X GOSUB 870,920,970,1020
820 RETURN
830 PRINT"■SHIFT CLR■":FOR RW=1TO9:FOR
    CL=1TO9
840 ON SA(RW,CL) GOSUB 870,920,970,1
    020
850 NEXT CL:NEXT RW
860 RETURN
870 AD=1036+(RW*80)+(CL*2)
880 POKE AD,169:POKE AD+1,170:POKE AD+4
    0,171:POKE AD+41,172
890 PA(RW,CL)=1
900 POKE S+AD,5:POKE S+AD+1,5:POKE S+(A
    D+40),5:POKE S+AD+41,5
910 RETURN
920 AD=1036+(RW*80)+(CL*2)
930 POKE AD,161:POKE AD+1,162:POKE AD+4
    0,163:POKE AD+41,164
940 PA(RW,CL)=2
950 POKE S+AD,1:POKE S+AD+1,1:POKE S+(A
    D+40),1:POKE S+AD+41,1
960 RETURN
970 AD=1036+(RW*80)+(CL*2)
980 POKE AD,97:POKE AD+1,98:POKE AD+40,
    99:POKE AD+41,100
990 PA(RW,CL)=3
1000 POKE S+AD,2:POKE S+AD+1,2:POKE S+(A
    D+40),2:POKE S+AD+41,2
1010 RETURN
1020 AD=1036+(RW*80)+(CL*2)
1030 POKE AD,101:POKE AD+1,102:POKE AD+4
    0,103:POKE AD+41,104
1040 PA(RW,CL)=4:KR=RW:KC=CL
1050 POKE S+AD,2:POKE S+AD+1,2:POKE S+(A
    D+40),2:POKE S+AD+41,2
1060 RETURN
1070 PRINT"■HOME■■21CRSRDOWN■"
1080 PRINT "■7CRSRRIGHT■ILLEGAL MOVE.  T
    RY AGAIN."
1090 FOR X=1 TO 2000:NEXT
1100 PRINT "■SHIFT CRSRUP■"
1110 RETURN
1120 PRINT "■HOME■■21CRSRDOWN■■2CRSRRIGH
    T■"
1130 PRINT "THE GAME IS OVER THE KING HA
    S ESCAPED"
1140 PRINT "WOULD YOU LIKE TO PLAY AGAIN
    "
1150 PRINT "ENTER Y OR N"
1160 Z=PEEK(197):IF Z=64 THEN 1160
1170 IF Z=25 THEN 490
1180 IF Z=39 THEN END
1190 GOTO 1160
1200 PRINT "■HOME■■21CRSRDOWN■■2CRSRRIGH
    T■"
1210 PRINT "THE GAME IS OVER THE KING HA
    S BEEN"
1220 PRINT "CAPTURED"
1230 PRINT "WOULD YOU LIKE TO PLAY AGAIN
    ?"
1240 PRINT "ENTER Y OR N"
1250 GOTO 1160
1260 GET A$:IF A$<>"" THEN 1260
1270 PRINT "■HOME■"
1280 FOR X=1TO20:GOSUB 1500:NEXT
1290 PRINT "■HOME■■4CRSRDOWN■■RED":PRINT"
    ■CRSRRIGHT■FROM"
1300 GOTO 1350
1310 GET A$:IF A$<>"" THEN 1310
1320 PRINT "■HOME■"
1330 FOR X=1TO20:GOSUB 1500:NEXT
1340 PRINT"■HOME■■4CRSRDOWN■■CRSRRIGHT■"
    :PRINT "WHITE":PRINT"■CRSRRIGHT■FRO
    M"
1350 GET C$:IF C$="" THEN 1350
1360 CF=ASC(C$):IF CF<65 OR CF>73 THEN 1
    350
1370 PRINT"■SHIFT CRSRUP■■7CRSRRIGHT■";C
    $
1380 GET R$:IF R$="" THEN 1380
1390 RF=ASC(R$):IF RF<49 OR RF>57 THEN 1
    380
```

Continued on p. 153

Tablut . . . from p. 61

TABLUT (IBM PCjr)
Explanation of the Program

Line Nos.	
100-180	Program header.
190-210	Dimension arrays; define shape strings for DRAW command; display title screen.
220-280	Initialize the game.
290-340	Display playing board.
350-420	Place pieces in starting positions.
430-460	Main control loop.
470-700	Input player's move and check legality.
710	Error routine.
720-800	Move player's piece.
810-970	Check for captured pieces.
980-1150	Check for capture of king.
1160-1270	DATA for TB( ) array (current position of pieces).
1280-1310	DATA for the TP( ) array (screen coordinates of positions on the board.)

IBM PC & PCjr

```
100 REM ***********
110 REM *  TABLUT  *
120 REM ***********
130 REM BY JAMES J. MULLIGAN AND HCM STAFF
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.2.1
160 REM IBM PCjr
170 REM CARTRIDGE BASIC AND 64K MEMORY EXPANSION
180 REM
190 CLS:DIM TB(9,9),TP(1,9):PS$="BR7BD2L2DLDLD3ND3RDR3UR2UR3UH2E2F2GLHEFLD3R3DRDRDR3U5LEFGD8BL2BD1L2UGULURHU4LD5HU4LD5LNU5LFL2FL3"
200 K$="BR2BD2DRDRGR2GR4UE3RFL2G2DRUER3FL4DR6E4DGDLDLFL15FR13DL13DR13GL11DR11DL11FR9"
210 KEY OFF:LOCATE 12,17:PRINT "TABLUT":FOR X=1 TO 2000:NEXT X:CLS
220 REM **************
230 REM INITIALIZE GAME
240 REM **************
250 SCREEN 1:COLOR 0,0
260 KING=0:KC=5:KR=5:PN=1
270 RESTORE 1250:FOR R=1 TO 9:FOR C=1 TO 9:READ TB(R,C):NEXT C:NEXT R
280 FOR X=0 TO 1:FOR Y=1 TO 9:READ TP(X,Y):NEXT Y:NEXT X
290 REM ******************
300 REM DISPLAY BOARD ON SCREEN
310 REM ******************
320 FOR R=12 TO 156 STEP 16:LINE (1,R)-(216,R),3:NEXT R:FOR C=1 TO 217 STEP 24:LINE (C,12)-(C,156),3:NEXT C
330 LOCATE 1,2:PRINT "A B C D E F G H I";
340 FOR R=1 TO 21:LOCATE R+2,29:PRINT MID$("1 2 3 4 5 6 7 8 9",R,1):NEXT R
350 REM ****************
360 REM PLACE MEN ON BOARD
370 REM ****************
380 FOR R=1 TO 9:FOR C=1 TO 9
390 IF TB(R,C)=1 THEN PSET (TP(0,R),TP(1,C)):DRAW "C1;XK$;":GOTO 420
400 IF TB(R,C)=2 THEN PSET (TP(0,R),TP(1,C)):DRAW "C1;XPS$;":GOTO 420
410 IF TB(R,C)=3 THEN PSET (TP(0,R),TP(1,C)):DRAW "C2;XPS$;"
420 NEXT C:NEXT R
430 REM ***************
440 REM MAIN CONTROL LOOP
450 REM ***************
460 GOSUB 500:GOSUB 750:GOSUB 840:GOSUB 1010:IF KING=1 THEN GOTO 1090 ELSE IF KING=2 THEN GOTO 1120 ELSE GOTO 460
470 REM ************
480 REM PLAYER INPUT
490 REM ************
500 PN=ABS(PN-1)
510 LOCATE 22,1:PRINT "                                ";
520 IF PN=1 THEN GOTO 570
530 LOCATE 22,1:PRINT "GREEN MOVES--:PRINT "FROM: ";:GOSUB 1200:PRINT A$;:F1=ASC(A$)-64:GOSUB 1210:PRINT A$;:F2=VAL(A$)
540 IF TB(F1,F2)<>1 AND TB(F1,F2)<>2 THEN GOTO 710
550 PRINT "                TO: ";:GOSUB 1200:PRINT A$;:T1=ASC(A$)-64:GOSUB 1210:PRINT A$;:T2=VAL(A$)
560 GOTO 600
570 LOCATE 22,1:PRINT "RED MOVES--:PRINT "FROM: ";:GOSUB 1200:PRINT A$;:F1=ASC(A$)-64:GOSUB 1210:PRINT A$;:F2=VAL(A$)
580 IF TB(F1,F2)<>3 THEN GOTO 710
590 PRINT "                TO: ";:GOSUB 1200:PRINT A$;:T1=ASC(A$)-64:GOSUB 1210:PRINT A$;:T2=VAL(A$)
600 IF T1<>F1 AND T2<>F2 THEN GOTO 710
610 IF T1=F1 THEN GOTO 650
620 FOR CM=F1+SGN(T1-F1) TO T1 STEP SGN(T1-F1):IF TB(CM,F2)>0 THEN GOTO 710
630 IF (CM=5 AND F2=5) AND (TB(F1,F2)<>1 OR PN=1) THEN GOTO 710
640 NEXT CM:GOTO 690
650 IF F2=T2 THEN GOTO 710
660 FOR CM=F2+SGN(T2-F2) TO T2 STEP SGN(T2-F2):IF TB(F1,CM)>0 THEN GOTO 710
670 IF (CM=5 AND F1=5) AND (TB(F1,F2)<>1 OR PN=1) THEN GOTO 710
680 NEXT CM
690 IF TB(F1,F2)=1 THEN KCT=KC:KRT=KR:KC=T1:KR=T2:GOSUB 1010:IF KING=2 THEN KING=0:KC=KCT:KR=KRT:GOTO 710
700 RETURN
710 LOCATE 22,1:PRINT "                                BAD INPUT                                TRY AGAIN                                ";CHR$(7);CHR$(7);CHR$(7);:FOR TD=1 TO 200:NEXT TD:GOTO 510
720 REM **********
730 REM MOVE PLAYER
740 REM **********
750 PSET (TP(0,F1),TP(1,F2)):IF TB(F1,F2)=1 THEN DRAW "C0;XK$;" ELSE DRAW "C0;XPS$;"
760 PSET (TP(0,T1),TP(1,T2))
770 IF TB(F1,F2)=1 THEN DRAW "C1;XK$;":KC=T1:KR=T2:GOTO 800
780 IF TB(F1,F2)=2 THEN DRAW "C1;XPS$;":GOTO 800
790 DRAW "C2;XPS$;"
800 TB(T1,T2)=TB(F1,F2):TB(F1,F2)=0:RETURN
810 REM **************
820 REM CHECK FOR CAPTURES
830 REM **************
840 IF TB(T1,T2)=1 THEN GOTO 960
850 IF PN=1 THEN MP=3:CP=2 ELSE MP=2:CP=3
860 REM *** WEST ***
870 IF T1>2 THEN IF TB(T1-1,T2)=CP AND TB(T1-2,T2)=MP THEN TB(T1-1,T2)=0:X=T1-1:Y=T2:GOSUB 950
880 REM **** EAST ****
890 IF T1<8 THEN IF TB(T1+1,T2)=CP AND TB(T1+2,T2)=MP THEN TB(T1+1,T2)=0:X=T1+1:Y=T2:GOSUB 950
900 REM **** NORTH ****
910 IF T2>2 THEN IF TB(T1,T2-1)=CP AND TB(T1,T2-2)=MP THEN TB(T1,T2-1)=0:X=T1:Y=T2-1:GOSUB 950
920 REM **** SOUTH ****
930 IF T2<8 THEN IF TB(T1,T2+1)=CP AND TB(T1,T2+2)=MP THEN TB(T1,T2+1)=0:X=T1:Y=T2+1:GOSUB 950
940 RETURN
950 PLAY "L64 T255 O4 BAGF#FEDC#C O3 BAGF#FEDC#C O2 BAGF#FEDC#C O1 BAGF#FEDC#C0;XPS$;":RETURN
960 IF T1=1 OR T1=9 OR T2=1 OR T2=9 THEN KING=1
970 RETURN
980 REM **********
990 REM IS KING CAPTURED
1000 REM **********
1010 IF NOT(TB(KC-1,KR)=3 OR KC=6 AND KR=5) THEN RETURN
1020 IF NOT(TB(KC+1,KR)=3 OR KC=4 AND KR=5) THEN RETURN
1030 IF NOT(TB(KC,KR-1)=3 OR KC=5 AND KR=6) THEN RETURN
1040 IF NOT(TB(KC,KR+1)=3 OR KC=5 AND KR=4) THEN RETURN
1050 KING=2:RETURN
1060 REM **********
1070 REM END OF GAME ROUTINES
1080 REM **********
1090 LOCATE 22,1:PRINT "THE KING HAS ARRIVED AT THE BOARDER                GREEN WINS                THE GAME                PLAY AGAIN (Y/N) ";:
1100 PLAY "T90 L16 O2 C#BDBEBF#BDBEBGB T80 C#D T60 EF#3C#BDBEBF#BDBEBGB T80 C#D T60 EF#T45 GA O4 T32 C#D"
1110 GOTO 1140
1120 LOCATE 22,1:PRINT "THE KING HAS BEEN CAPTURED BY THE RED WINS THE GAME                PLAY AGAIN (Y/N) ";:
1130 PLAY "T80 O1 L4 A L6 A L12 A L4 A O2 L6 C L12 O1 B L6 B L12 A L6 A L12 O1 G# L4 A"
1140 A$=INKEY$:IF A$<>"Y" AND A$<>"N" THEN GOTO 1140
1150 IF A$="Y" THEN PN=1:CLS:GOTO 250
1160 SCREEN 0:PRINT "BYE":END
1170 REM **********
1180 REM INPUT SUBROUTINES
1190 REM **********
```

Continued on p. 153

Tablut . . . from p. 152

IBM PC & PCjr

```
 1200 A$=INKEY$:IF A$<"A" OR A$>"I" THEN
      GOTO 1200 ELSE PRINT CHR$(7);:RETUR
      N
 1210 A$=INKEY$:IF A$<"1" OR A$>"9" THEN
      GOTO 1210 ELSE PRINT CHR$(7);:RETUR
      N
 1220 REM ***************
 1230 REM DATA FOR TB() ARRAY
 1240 REM ***************
 1250 DATA 0,0,0,3,3,3,0,0,0
 1260 DATA 0,0,0,0,3,0,0,0,0
 1270 DATA 0,0,0,0,2,0,0,0,0
 1280 DATA 3,0,0,0,2,0,0,3,3
 1290 DATA 3,3,2,2,1,2,2,3,3
 1300 DATA 3,0,0,0,2,0,0,0,3
 1310 DATA 0,0,0,0,2,0,0,0,0
 1320 DATA 0,0,0,0,3,0,0,0,0
 1330 DATA 0,0,0,3,3,3,0,0,0
 1340 REM ***************
 1350 REM DATA FOR TP() ARRAY
 1360 REM ***************
 1370 DATA 1,25,49,73,97,121,145,169,193,
      13,29,45,61,77,93,109,125,141
```

HCM

Tablut . . . from p. 151

COMMODORE 64

```
 1400 PRINT"■2CRSRRIGHT■SHIFT CRSRUP■7C
      RSRRIGHT■";R$
 1410 PRINT "■3CRSRRIGHT■TO"
 1420 GET C1$:IF C1$="" THEN 1420
 1430 CT=ASC(C1$):IF CT<65 OR CT>73 THEN
      1420
 1440 PRINT "■7CRSRRIGHT■SHIFT CRSRUP■";
      C1$
 1450 GET R1$:IF R1$="" THEN 1450
 1460 RT=ASC(R1$):IF RT<49 OR RT>57 THEN
      1450
 1470 PRINT "■9CRSRRIGHT■SHIFT CRSRUP■";
      R1$
 1480 CF=CF-64:CT=CT-64:RF=RF-48:RT=RT-48
 1490 RETURN
 1500 PRINT " ":RETURN
 1510 R1=RW:C1=CL
 1520 IF PA(R1-1,C1)<>3 THEN 1550
 1530 IF PA(R1-2,C1)<>2 THEN 1550
 1540 RW=R1-1:CL=C1:GOSUB 870
 1550 IF PA(R1+1,C1)<>3 THEN 1580
 1560 IF PA(R1+2,C1)<>2 THEN 1580
 1570 RW=R1+1:CL=C1:GOSUB 870
 1580 IF PA(R1,C1-1)<>3 THEN 1610
 1590 IF PA(R1,C1-2)<>2 THEN 1610
 1600 RW=R1:CL=C1-1:GOSUB 870
 1610 IF PA(R1,C1+1)<>3 THEN 1640
 1620 IF PA(R1,C1+2)<>2 THEN 1640
 1630 RW=R1:CL=C1+1:GOSUB 870
 1640 KF=0:GOSUB 1810
 1650 RETURN
 1660 R1=RW:C1=CL:IF PA(R1,C1)=4 THEN 179
      0
 1670 IF PA(R1-1,C1)<>2 THEN 1700
 1680 IF PA(R1-2,C1)<>3 THEN 1700
 1690 RW=R1-1:CL=C1:GOSUB 870
 1700 IF PA(R1+1,C1)<>2 THEN 1730
 1710 IF PA(R1+2,C1)<>3 THEN 1730
 1720 RW=R1+1:CL=C1:GOSUB 870
 1730 IF PA(R1,C1-1)<>2 THEN 1760
 1740 IF PA(R1,C1-2)<>3 THEN 1760
 1750 RW=R1:CL=C1-1:GOSUB 870
 1760 IF PA(R1,C1+1)<>2 THEN 1790
 1770 IF PA(R1,C1+2)<>3 THEN 1790
 1780 RW=R1:CL=C1+1:GOSUB 870
 1790 KF=0:GOSUB 1810
 1800 RETURN
 1810 KV=0
 1820 IF PA(KR-1,KC)=2 THEN KV=KV+1:GOTO
      1840
 1830 IF KR-1=5 AND KC=5 AND PA(KR-1,KC)=
      1 THEN KV=KV+1
 1840 IF PA(KR+1,KC)=2 THEN KV=KV+1:GOTO
      1860
 1850 IF KR+1=5 AND KC=5 AND PA(KR+1,KC)=
      1 THEN KV=KV+1
 1860 IF PA(KR,KC-1)=2 THEN KV=KV+1:GOTO
      1880
 1870 IF KC-1=5 AND KR=5 AND PA(KR,KC-1)=
      1 THEN KV=KV+1
 1880 IF PA(KR,KC+1)=2 THEN KV=KV+1:GOTO
      1900
 1890 IF KC+1=5 AND KR=5 AND PA(KR,KC+1)=
      1 THEN KV=KV+1
 1900 IF KV>3 THEN KF=1:GOTO 1940
 1910 IF TN<>3 THEN 1940
 1920 IF KC<>9 AND KC<>1 AND KR<>9 AND KR
      <>1 THEN 1940
 1930 KF=2
 1940 RETURN
```

HCM

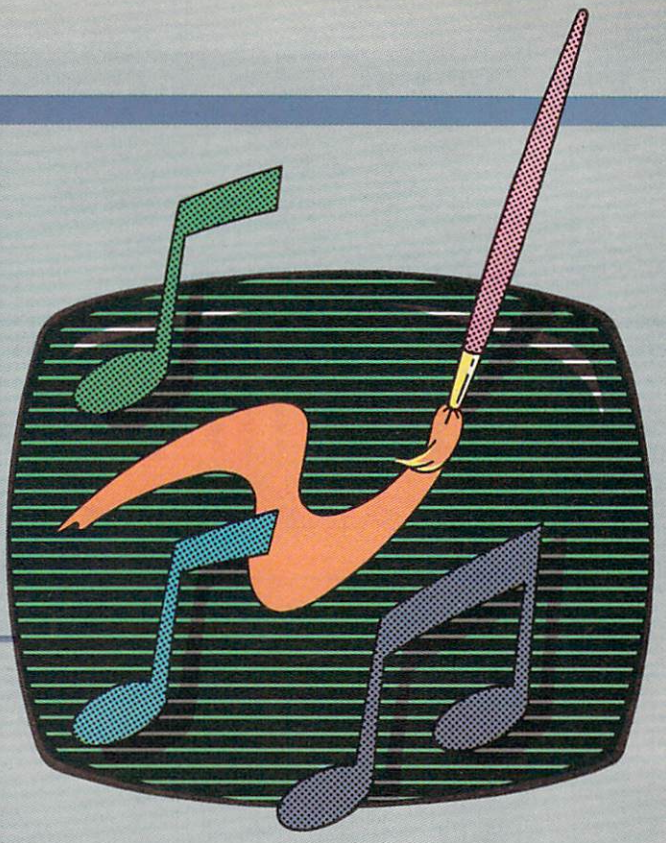
Tablut . . . from p. 121

TI-99/4A

```
  830 CALL HCHAR(T1*2,T2*2+9,CH)::: CALL H
      CHAR(T1*2,T2*2+10,CH+1)::: CALL HCHA
      R(T1*2+1,T2*2+9,CH+2)::: CALL HCHAR(
      T1*2+1,T2*2+10,CH+3)
  840 IF TB(F1,F2)=1 THEN KC=T1 ::: KR=T2
  850 TB(T1,T2)=TB(F1,F2)::: TB(F1,F2)=0 :
      : RETURN
  860 REM ***CHECK CAPTURES***
  870 IF TB(T1,T2)=1 THEN 980
  880 IF PN=1 THEN MP=3 :: CP=2 ELSE MP=2
      ::: CP=3
  890 REM ***WEST***
  900 IF T1>1 THEN IF TB(T1-1,T2)=CP AND
      TB(T1-2,T2)=MP THEN TB(T1-1,T2)=0 :
      : X=T1-1 ::: Y=T2 :: GOSUB 1150
  910 REM***EAST***
  920 IF T1<8 THEN IF TB(T1+1,T2)=CP AND
      TB(T1+2,T2)=MP THEN TB(T1+1,T2)=0 :
      : X=T1+1 ::: Y=T2 :: GOSUB 1150
  930 REM***NORTH***
  940 IF T2>1 THEN IF TB(T1,T2-1)=CP AND
      TB(T1,T2-2)=MP THEN TB(T1,T2-1)=0 :
      : X=T1 ::: Y=T2-1 :: GOSUB 1150
  950 REM***SOUTH***
  960 IF T2<8 THEN IF TB(T1,T2+1)=CP AND
      TB(T1,T2+2)=MP THEN TB(T1,T2+1)=0 :
      : X=T1 ::: Y=T2+1 :: GOSUB 1150
  970 RETURN
  980 IF T1=1 OR T1=9 OR T2=1 OR T2=9 THE
      N KING=1
  990 RETURN
 1000 REM***KING CAPTURED?***
 1010 IF NOT(TB(KC-1,KR)=3 OR KC=6 AND KR
      =5)THEN RETURN
 1020 IF NOT(TB(KC+1,KR)=3 OR KC=4 AND KR
      =5)THEN RETURN
 1030 IF NOT(TB(KC,KR-1)=3 OR KC=5 AND KR
      =6)THEN RETURN
 1040 IF NOT(TB(KC,KR+1)=3 OR KC=5 AND KR
      =4)THEN RETURN
 1050 KING=2 :: RETURN
 1060 REM***WINNING***
 1070 DISPLAY AT(21,1):"THE KING HAS ARRI
      VED AT THE":"EDGE OF THE BOARD. RED
       WINS."
 1080 GOSUB 1160 :: FOR Z=1 TO 12 :: CALL
       SOUND(-10,110*Z,0):: NEXT Z :: GOT
      O 1100
 1090 DISPLAY AT(21,1):"THE KING HAS BEEN
       CAPTURED.":"WHITE WINS." :: GOTO 1
      080
 1100 DISPLAY AT(23,1):"ANOTHER GAME? (Y
      OR N)" :: ACCEPT AT(23,24)VALIDATE(
      "YN"):SIZE(1):ANS$
 1110 IF ANS$="N" THEN CALL CLEAR :: GOTO
       1170
 1120 PN=1 :: GAME=99 :: CALL CLEAR :: GO
      TO 260
 1130 CALL KEY(0,K1,S)::: IF S<>1 OR K1<64
       OR K1>73 THEN GOTO 1130 ELSE CALL
      SOUND(10,880,0)::: RETURN
 1140 CALL KEY(0,K2,S)::: IF S<>1 OR K2<49
       OR K2>57 THEN GOTO 1140 ELSE CALL
      SOUND(10,880,0)::: RETURN
 1150 CALL HCHAR(X*2,Y*2+9,120)::: CALL HC
      HAR(X*2,Y*2+10,121)::: CALL HCHAR(X*
      2+1,Y*2+9,122)::: CALL HCHAR(X*2+1,Y
      *2+10,123)::: RETURN
 1160 FOR X=1 TO 10 :: CALL VCHAR(1,X,32,
      20)::: NEXT X :: RETURN
 1170 END
```

HCM

POWER PAD A REVIEW



by Robert Keller
HCM Staff

What is a "PowerPad?" Does it give you special powers in your home? Yes, in a way. PowerPad (made by Chalk Board, Inc.) is a computer peripheral more generically known as a touch-tablet. As such, it acts as an input device to a variety of home computers.

Chalk Board plans a total of 30 packages of what it calls "learning/discovery" software, intended to cover the visual arts, music, mathematics, science, language arts, and social studies. Eight packages are available now in varying combinations for the VIC-20, C-64, Atari, Apple II, and the IBM series—PCjr, PC and XT computers; 6 additional packages are in various stages of release. All carry a suggested retail price of between $24.95 and $49.95.

The PowerPad surface is blank, and each software application comes with a polycarb overlay that is imprinted with areas for the application's menu and functions. The overlay fits onto the PowerPad surface.

A touch-sensitive pad has a surface upon which, with a finger or a stylus, information can be sent directly into the computer. The PowerPad, in addition, can sense multiple touches. It has a large 12" by 12" active surface and is apparently unique in that more than one person can use it at a time and one person can do more than one thing at a time with it—such as play chords (in the piano software package).

Most pads use a "membrane switch," a sandwich of two conducting foils that press together when the pad is touched. Chalk Board calls pads built in this fashion "analog" pads and points out that they can sense only one contact at a time. Chalk Board inscribes their membranes with 10 lines per inch to produce 100 discrete contact points per square inch, and the software, which samples this "digital" pad through a serial (usually a game) port 20 times per second, can read discrete contact points separately. [See the accompanying interview with Robert Ranson, Chalk Board's president, for more about the PowerPad hardware and software, including a discussion of its durability.—*Ed.*]

PowerPad's Software

We had a chance to look at four software packages from Chalk Board: *Micro Illustrator* for the Apple II, and *LogicMaster*, *MusicMaestro*, and *Leo's Links* for the C-64. We tried *Micro Illustrator* first. It was written by a third- party software firm, Island Corporation, and is also bundled with the Koala Pad, another touch-sensitive device reviewed in our last issue of (Vol. 4 No.1, p 186).

Getting the package to run was a challenge. The first piece of literature we encountered was the PowerPad Manual, which mentioned that the enclosed stylus ought to be used to draw upon the pad. We were unable to find a stylus in either of two packages. We decided to draw with our fingers, but five tries at booting with the *Micro Illustrator* disk were fruitless. Several calls to Georgia and several days later we learned that the PowerPad as designed did not work on our Apple and that a new cable design would have to be shipped to us. The company assured us that no such disastrous problems existed in any of the rest of the products, and we found none.

The PowerPad version of *Micro Illustrator* is substantially the same as Koala's version. Moving the finger or the stylus on the pad moves the cursor over an initial screen menu. Pressing the appropriate spot on the pad overlay moves you quickly to a full screen for drawing, and you can quickly return to the menu as many times as necessary. You can draw with eight different "brushes," fill figures with selected colors, manipulate and place geometrical shapes, and erase and save pictures—all easily and directly.

Island has added two menu options to this version of the software. MIRROR combines with any other design command to produce a four-part mirror image extending from the center of the screen, and SCALED helps you find your place. In the standard drawing mode, a line begins at the point corresponding to where you place your stylus on the pad and ends at the point where you lift your stylus. This can prove difficult if you lift your pen once in a while when you draw, for it is often difficult to find your starting point again. Remember that the pad remains blank and the drawing appears only on the screen. When you select SCALED, your drawing continues from the last point drawn on the screen, regardless of where you put your stylus down on the pad when you resume.

Micro Illustrator is really part of a new era in computer graphics. I wrote HELLO across the screen in smooth pink script, added a purple background, and a line of fellow workers eager to try their hand soon formed beside me.

The second package, *MusicMaestro*, has a two-octave piano keyboard and command buttons on its overlay, and you can play on the keyboard while the computer screen more or less writes out your efforts on a music staff. Writing the correct time values for notes is a difficult problem that *MusicMaestro* doesn't pretend to have solved. The package

PowerPad
Chalk Board,Inc.
3772 Pleasantdale Rd.
Atlanta, GA 30340
$99.95

is content to show you the correct placement of the note that you just played and write all notes as quarter notes.You can play chords, and they faithfully appear on the staff. You can also save what you play to disk and recall it. This package is a real boon for anyone with a fledgling interest in music, although slow access through the serial port and the moderate speed of the venerable 6500-series microprocessor make playing anything in tempo difficult. Too often, when you press a key, nothing happens for a split second. The computer does remember what you play and will play it back for you, embarrassing pauses and all. The difficulty is partially circumvented with the STAR commands, which removes all space between notes on playback. (STAR also allows you to stop and think in the middle of a composition and to remove the pause later.) A REST command will put spaces in that can't be removed by STAR, but I'm not sure that these software commands can totally circumvent the hardware speed problem.

The software also has two levels of playback, PLAY and PLAY ALL. You can record music in little sections and combine them into a longer song. You can use a variation of the same procedure to add additional voices to a song. All you do is play along with a song that you've already recorded, and your notes are added. You can then erase only the added notes if you like, and add new ones. It's a great way for someone to learn about harmony.

In a section called "Circus Songs," the manual suggests that the user (evidently a child) imagine various scenes in a circus. "Try thinking of one part of the circus at a time—such as the merry-go-round," says the manual. "Play with PowerPad and MusicMaestro until you find the sounds that make you think of a merry-go-round." That's a flash of brilliance.

We tried out *Leo's Links* on a Saturday. The first step was to design a golf course, complete with sand-traps, dog-legs, rough, and strategic trees. I had just stopped in for a short work session, intending to take my nephew golfing on a more traditional course later in the day, but the sky clouded over; so with a colleague we formed a threesome and ventured onto *Leo's Links* . My nephew is a 14-year-old who has never come close to beating me in golf, though he thinks he really ought to. On *Leo's Links* he took the lead on the first hole, weathered our gaze as he teed off first on each subsequent hole, and finally withstood our desperate attempts to drive over dog-legs and through hazards to catch him. On the 18th fairway he had a five-stroke lead and needed only to chip onto the final green for the win, but in his only display of adolescent nerves, he proceeded to hit two into the brush,

Continued on p. 160

A CONVERSATION WITH
ROBERT RANSON

President of Chalk Board, Inc.

HCM talked recently with Robert Ranson, President of Chalk Board, Inc., the developers of the PowerPad touch-sensitive device. In the course of our conversation we talked about the PowerPad itself, about the market implications of such products, and about future developments in the industry that Ranson sees coming.

HCM: What were the market needs that led you to develop the PowerPad?

RR: We began Chalk Board in January of 1983, and we originally envisioned it as a software company. We wanted to write software that would involve the widest possible family audience in the Piaget method of *learning by discovery*. We became aware that if our audience—we wanted to include the youngest child who could reason and deal with the computer, as well as adults—was visually or physically or psychologically intimidated by the keyboard, then we were going to be constrained by that problem no matter what we did in software.

We began to look around for an alternative device. We also wanted something that could be used by more than one person at the same time: a parent and a child, a teacher and a child, or two children; we wanted it to be cost-justified, and we wanted it to be what we called multi-informational—in that you could be pressing different things at the same time, and the computer would know that two hands were doing something, and not just one finger.

We went about looking through the marketplace: We looked at light pens, we looked at bar codes, and we didn't find anything we thought met those criteria. So we just sat down and designed something.

HCM: Probably the most striking aspect of your pad is the size. Can you tell us more about that?

RR: If entertainment or education is going to work in the home, you can't expect children to do it on their own—especially if they're young. You've got to have the parent enthusiastic and willing

Continued on p. 160

2 IMPORTANT PERIPHERALS

For Your Texas Instruments Personal Computer

$309

Direct-Connect Printer, GP-100TI

Because of our built-in interface, all you have to do to connect a GP-100TI printer to your Texas Instruments 99/4A computer is plug it in. The GP-100TI has adjustable tractors, cartridge ribbon, uses standard fan-fold paper and has full dot-addressable graphics. It provides clear, dot-matrix printout from all TI software including cartridge, which allows you to set number of line feeds per line, line length, margins and self-test. Another plus — expansion modules can be daisy-chained through the built-in edge connector.

$99

ParallAx TI Expansion Interface

You no longer have to settle for Texas Instruments' bulky expansion box with its messy cable when for just $99 you can connect any parallel-input printer to your TI computer using our ParallAx TI printer interface. It's smaller, neater, and plugs into the side of your TI computer. ParallAx TI has the same features as our GP-100TI printer (described above) and you can daisy-chain expansion modules through the built-in edge connector.

THESE QUALITY AXIOM PRODUCTS ARE AVAILABLE AT LEADING DEPARTMENT AND DISCOUNT STORES.

FOR THE OUTLET NEAREST YOU, CALL 818/365-9521.

AXIOM

AXIOM CORPORATION
1014 Griswold Avenue, San Fernando, California 91340
818/365-9521 • TWX 910-496-1746

3D-IIe. . . from p. 41

APPLESOFT 3-D EDITOR Listing 5
Explanation of the Program

Line Nos.	
100-170	Program Header.
180-240	Redraws object from editor array.
250-300	Searches to see if point to be added to array is new or a previously plotted point.
310-400	Searches to determine type of line to be erased.
410-500	Fills XYZ%( ) array with point information: coordinates, number assignment, type of point.
510-660	Fills PC%( ) array with point numbers to be connected, in proper order.
670-710	Fills XE%( ) and YE%( ) editor arrays used to redraw object.
720-770	Plots cursor by XDRAWing shape, after making adjustments.
780-800	Erases cursor at old position.
810-840	Toggles cursor speed by adjusting number of units (one or ten) that cursor moves per keypress.
850-940	Decrements cursor movement.
950-1040	Increments cursor movement.
1050-1140	Plots point.
1150-1260	Plots line.
1270-1370	Plots ray.
1380-1550	Erases last line drawn by decrementing top-of-array pointers, after first determining type of line erased.
1560-1590	Initializes screen with HGR, draws border.
1600-1640	Clears screen, reinitializes.
1650-1790	Main driver, gets command input.
1800-1900	Four lines at bottom of editing screen.
1910-1960	Updates cursor coordinates.
1970-2060	Screen with list of editor commands.
2070-2220	First menu.
2230-2360	Second menu.
2370-2430	Editor reinitialization.
2440-2690	Load object from disk.
2700-3230	Save object to disk.
3240-3340	Disk error-handling routine.
3350-3430	Quits Editor, returns to System menu.
3440-3520	Creates cursor by poking shape table into memory.
3530-3630	Main initialization.

APPLE II Series

```
100 REM  **********
110 REM  *  APPLESOFT 3-D
120 REM  *  EDITOR
130 REM  **********
140 REM  BY M. D. BROWNSWORTH
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.2.1
170 REM  APPLE II SERIES APPLESOFT
180 GOTO 3530
190 REM  REDRAW SCREEN*****
200 GOSUB 1560: REM  SCREEN INITIALIZE
210 FOR J = 2 TO C STEP 2
220 HPLOT XE%(J - 1),YE%(J - 1) TO XE%(
    J),YE%(J)
230 NEXT J
240 RETURN
250 REM  SEARCH 1*****
260 IF P = 1 THEN RETURN
270 FOR J = 1 TO P - 1
280 IF X = XYZ%(J,1) AND Y = XYZ%(J,2)
    AND Z = XYZ%(J,3) THEN M = J: J = P
    - 1: REM  IF MATCH FOUND, PUT ARRAY
    ELEMENT NUMBER IN VARIABLE M
290 NEXT J
300 RETURN
310 REM  SEARCH 2*****
320 IF XYZ%(P - 1,5) = 1 THEN W = 1: GO
    TO 340
330 W = 0
340 FOR I = 0 TO W
350 FOR J = 1 TO P - (I + 1)
360 IF XYZ%(J,4) = XYZ%(P - I,4) THEN M
    = M + 1: J = P - (I + 1)
370 NEXT J
380 NEXT I
390 M = M + 1
400 RETURN
410 REM  FILL XYZ%()*****
420 SFLAG = OFF
430 P = P + 1: REM  POINT COUNTER
440 XYZ%(P,1) = X: XYZ%(P,2) = Y: XYZ%(P,
    3) = Z
450 IF C > 1 THEN GOSUB 250: REM  SEAR
    CH TO SEE IF POINT IS IN ARRAY
460 IF M > 0 THEN XYZ%(P,4) = XYZ%(M,4)
    : M = 0: GOTO 490: REM  MATCH WITH P
    REVIOUS POINT
470 N = N + 1
480 XYZ%(P,4) = N: REM  NO MATCH FOUND
    ; ASSIGN NEW NUMBER
490 XYZ%(P,5) = TYPE: REM  POINT, LINE
    OR RAY
500 RETURN
510 REM  FILL PC%()*****
520 I = 0: J = 0: K = 1: R = 0
530 J = J + 2
540 I = I + 1: IF I > P THEN J = J - 2:
    RETURN
550 ON XYZ%(I,5) GOTO 560,570,590
560 PC%(J - K) = XYZ%(I,4): GOTO 640
570 IF R > 0 THEN PC%(J - K) = R: R = 0:
    GOTO 640
580 PC%(J - K) = XYZ%(I - K,4): GOTO 64
    0
590 IF K = 0 THEN PC%(J) = XYZ%(I,4): G
    OTO 640
600 IF R > 0 THEN GOTO 630
610 IF XYZ%(I - 1,5) = 2 THEN R = XYZ%(
    I - 1,4): GOTO 630
620 R = XYZ%(I - 2,4)
630 PC%(J - K) = R
640 IF K = 0 THEN K = 1: GOTO 530
650 IF XYZ%(I,5) = 1 THEN K = 0: GOTO 5
    40
660 K = 0: GOTO 550
670 REM  FILL XE%(),YE%()*****
680 C = C + 2: REM  LINE (POINT-PAIR) C
    OUNTER
690 XE%(C - 1) = XP: YE%(C - 1) = YP: RE
    M  LINE START
700 XE%(C) = XE: YE%(C) = YE: REM  LINE
    END
710 RETURN
720 REM  PLOT CURSOR*****
730 XADJ = X: YADJ = ABS (Y - 160): ZADJ
    = Z * .5: REM  REVERSE DIRECTI
    ON OF Y-AXIS, SCALE Z-AXIS
740 XCUR = XADJ + ZADJ: YCUR = YADJ - ZA
    DJ: REM  CURSOR
750 XDRAW 1 AT XCUR,YCUR: REM  DRAW AT
    NEW POSITION
760 XOLD = XCUR: YOLD = YCUR: REM  READY
    TO ERASE AFTER MOVE
770 RETURN
780 REM  ERASE CURSOR*****
790 XDRAW 1 AT XOLD,YOLD: REM  ERASE CU
    RSOR AT OLD POSITION
800 RETURN
810 REM  CURSOR SPEED TOGGLE*****
820 IF INC = 1 THEN INC = 10: RETURN
830 IF INC = 10 THEN INC = 1
840 RETURN
850 REM  CURSOR DECREMENT*****
860 X = X - INC
```

APPLE II Series

```
460 IF M > 0 THEN XYZ%(P,4) = XYZ%(M,4)
    : M = 0: GOTO 490: REM  MATCH WITH P
    REVIOUS POINT
470 N = N + 1
480 XYZ%(P,4) = N: REM  NO MATCH FOUND
    ; ASSIGN NEW NUMBER
490 XYZ%(P,5) = TYPE: REM  POINT, LINE
    OR RAY
500 RETURN
510 REM  FILL PC%()*****
520 I = 0: J = 0: K = 1: R = 0
530 J = J + 2
540 I = I + 1: IF I > P THEN J = J - 2:
    RETURN
550 ON XYZ%(I,5) GOTO 560,570,590
560 PC%(J - K) = XYZ%(I - K,4): GOTO 64
    0
570 IF R > 0 THEN PC%(J - K) = R: R = 0:
    GOTO 640
580 PC%(J - K) = XYZ%(I - K,4): GOTO 64
    0
590 IF K = 0 THEN PC%(J) = XYZ%(I,4): G
    OTO 640
600 IF R > 0 THEN GOTO 630
610 IF XYZ%(I - 1,5) = 2 THEN R = XYZ%(
    I - 1,4): GOTO 630
620 R = XYZ%(I - 2,4)
630 PC%(J - K) = R
640 IF K = 0 THEN K = 1: GOTO 530
650 IF XYZ%(I,5) = 1 THEN K = 0: GOTO 5
    40
660 K = 0: GOTO 550
670 REM  FILL XE%(),YE%()*****
680 C = C + 2: REM  LINE (POINT-PAIR) C
    OUNTER
690 XE%(C - 1) = XP: YE%(C - 1) = YP: RE
    M  LINE START
700 XE%(C) = XE: YE%(C) = YE: REM  LINE
    END
710 RETURN
720 REM  PLOT CURSOR*****
730 XADJ = X: YADJ = ABS (Y - 160): ZADJ
    = Z * .5: REM  REVERSE DIRECTI
    ON OF Y-AXIS, SCALE Z-AXIS
740 XCUR = XADJ + ZADJ: YCUR = YADJ - ZA
    DJ: REM  CURSOR
750 XDRAW 1 AT XCUR,YCUR: REM  DRAW AT
    NEW POSITION
760 XOLD = XCUR: YOLD = YCUR: REM  READY
    TO ERASE AFTER MOVE
770 RETURN
780 REM  ERASE CURSOR*****
790 XDRAW 1 AT XOLD,YOLD: REM  ERASE CU
    RSOR AT OLD POSITION
800 RETURN
810 REM  CURSOR SPEED TOGGLE*****
820 IF INC = 1 THEN INC = 10: RETURN
830 IF INC = 10 THEN INC = 1
840 RETURN
850 REM  CURSOR DECREMENT*****
860 X = X - INC
```

Continued on p. 158

Editor Commands at a Glance

AXIS SELECTION: the X, Y, and Z keys select the axes along which the cursor moves.

ARROWS: The right and left arrows move the cursor along the three axes. If the Editor is run on an Apple IIe, which is equipped with up and down arrows as well, these keys may be used to move the cursor along the *y* axis.

POINT: The P key sets a point to use as the beginning of a line or ray.

LINE: The L key draws a line to the cursor position from either a set point or the end of a previous line.

RAY: The R key draws a special type of line known as a *ray* from a common vertex formed by other rays to the cursor position.

SPEED: The S key toggles the number of units per arrow keypress that the cursor moves, from one unit to ten units.

CLEAR: The C key clears the screen and lets you start over.

ERASE: The E key erases the last line or ray drawn. It will not erase a point. To change your mind about a point set, draw a line and then erase.

MENU: The [ESC] key is used to return to the menu at any point in the editing process.

APPLE II Series

```
 870 IF  X  <  10  THEN  X  =  10:  REM     SCREEN
     LIMITS,  MINIMUM
 880 RETURN
 890 Y  =  Y  -  INC
 900 IF  Y  <  10  THEN  Y  =  10
 910 RETURN
 920 Z  =  Z  -  INC
 930 IF  Z  <  0  THEN  Z  =  0
 940 RETURN
 950 REM    CURSOR  INCREMENT*****
 960 X  =  X  +  INC
 970 IF  X  +  ZADJ  >  270  THEN  X  =  270  -  ZA
     DJ:  REM     SCREEN  LIMITS,  MAXIMUM
 980 RETURN
 990 Y  =  Y  +  INC
1000 IF  Y  +  ZADJ  >  150  THEN  Y  =  150  -  ZA
     DJ
1010 RETURN
1020 Z  =  Z  +  INC
1030 IF  XADJ  +  Z  *  .5  >  270  OR  YADJ  -  Z
     *  .5  <  10  THEN  Z  =  Z  -  INC:  REM     LI
     MITS  MAXIMUM  CURSOR  TRAVEL  ON  Z-AXI
     S
1040 RETURN
1050 REM    PLOT  POINT*****
1060 IF  TYPE  =  1  THEN  RETURN  :  REM     T
     WO  CONSECUTIVE  POINTS  NOT  ALLOW
     ED
1070 PRINT  BEEP$;:TYPE  =  1
1080 XP  =  XCUR:YP  =  YCUR
1090 GOSUB  720:  REM    REMOVE  CURSOR
1100 HPLOT  XP,YP
1110 GOSUB  720:  REM    REDRAW  CURSOR
1120 GOSUB  410:  REM    SAVE  COORDINATES  IN
     POINT  ARRAYS
1130 GOSUB  1850:  REM    LAST  PLOT-TYPE  DIS
     PLAY
1140 RETURN
1150 REM    PLOT  LINE*****
1160 IF  SFLAG  =  SET  THEN  RETURN  :  REM
     MUST  SET  POINT  FIRST
1170 PRINT  BEEP$;:TYPE  =  2
1180 XL  =  XCUR:YL  =  YCUR
1190 GOSUB  720
1200 HPLOT  XP,YP  TO  XCUR,YCUR
1210 GOSUB  720
1220 XE  =  XCUR:YE  =  YCUR:  REM    PASS  VALU
     ES  FOR  EDITOR  ARRAY
1230 GOSUB  410:  GOSUB  670:  REM    SAVE  IN
     ALL  ARRAYS
1240 XP  =  XCUR:YP  =  YCUR:  REM    SET  BEGIN
     NING  OF  NEXT  LINE  TO  END  OF  THIS  ON
     E
1250 GOSUB  1850
1260 RETURN
1270 REM    PLOT  RAY*****
1280 IF  SFLAG  =  SET  THEN  RETURN
1290 PRINT  BEEP$;:TYPE  =  3
1300 GOSUB  720
1310 HPLOT  XP,YP  TO  XCUR,YCUR
1320 GOSUB  720
1330 XE  =  XCUR:YE  =  YCUR
1340 GOSUB  410:  GOSUB  670
1350 REM    DOES  NOT  SET  LINE  END  AS  START
     OF  NEXT  LINE
1360 GOSUB  1850
1370 RETURN
1380 REM    ERASE  LAST  PLOT*****
1390 IF  C  =  0  THEN  RETURN  :  REM    DETERM
     INE  A  LINE  EXISTS  TO  ERASE  BEFORE  C
     ONTINUING
1400 IF  XYZ%(P,5)  =  1  THEN  RETURN  :  REM
     NO  ERASE  IF  ONLY  POINT  SET
1410 GOSUB  310:  REM    SEARCH
1420 IF  XYZ%(P  -  1,5)  =  1  THEN  PDEC  =  2:
     ON  M  GOTO  1440,1450,1460
1430 PDEC  =  1:  ON  M  GOTO  1450,1460,1460
1440 PDEC  =  2:  GOTO  1470
1450 PDEC  =  1:  GOTO  1470
1460 PDEC  =  0
1470 P  =  P  -  PDEC:N  =  N  -  NDEC:M  =  0
1480 C  =  C  -  2:  REM    DECREMENT  LINE  INDE
     X
1490 IF  C  =  0  THEN  SFLAG  =  SET:XE%(1)  =
     0:YE%(1)  =  0:  GOSUB  2370:  RETURN  :
     REM    RESTART  WITH  BLANK  SCREEN
1500 X  =  XYZ%(P,1):Y  =  XYZ%(P,2):Z  =  XYZ
     %(P,3):XP  =  XE%(C):YP  =  YE%(C):  REM
     RESET  START  POINT
1510 IF  XYZ%(P,5)  =  3  THEN  XP  =  XE%(C  -
     1):YP  =  YE%(C  -  1):  GOTO  1530:  REM
     RESET  RAY  START  POINT
1520 XP  =  XE%(C):YP  =  YE%(C)
1530 GOSUB  190:  REM    CLEAR  SCREEN,  REDRA
     W  FIGURE  MINUS  LAST  LINE  PLOTTED
1540 GOSUB  720
1550 RETURN
1560 REM    SCREEN  INITIALIZE*****
1570 HGR
1580 HPLOT  0,0  TO  0,159  TO  279,159  TO  27
     9,0  TO  0,0:  REM    BORDER
1590 RETURN
1600 REM    CLEAR  ROUTINE*****
1610 VTAB  21:  HTAB  1:  CALL  SCNCLR:  VTAB
     22:  PRINT  "OK  TO  CLEAR?  Y";
```

APPLE II Series

```
1620 HTAB  14:  POKE  KEYCLR,0:  GET  K$
1630 IF  K$  =  "Y"  OR  K$  =  CHR$(13)  THEN
     SFLAG  =  SET:XE%(1)  =  0:YE%(1)  =  0:
     GOSUB  2370:  GOSUB  1800:  RETURN  :  R
     EM    REINITIALIZE
1640 RETURN
1650 REM    MAIN  DRIVER*****
1660 GOSUB  1910:  REM    UPDATE  COORDINATE
     READOUT
1670 VTAB  V:  HTAB  2:  POKE  KEYCLR,0:  GET
     K$
1680 IF  K$  =  CHR$(27)  THEN  RETURN  :  R
     EM    MAIN  MENU
1690 IF  K$  =  "X"  OR  K$  =  "Y"  OR  K$  =  "Z"
     THEN  AXIS  =  ASC(K$)  -  87:  REM
     CHANGE  AXIS
1700 IF  K$  =  CHR$(8)  OR  K$  =  CHR$(10
     )  THEN  ON  AXIS  GOSUB  860,890,920:
     GOSUB  780:  GOSUB  720:  REM    <-  MOVES
     CURSOR  BY  DECREMENTING  X,Y,OR  Z
1710 IF  K$  =  CHR$(21)  OR  K$  =  CHR$(11
     1)  THEN  ON  AXIS  GOSUB  960,990,1020
     :  GOSUB  780:  GOSUB  720:  REM    ->  MOV
     ES  BY  INCREMENTING
1720 IF  K$  =  CHR$(13)  THEN  GOSUB  1970
     :  REM    COMMAND  LIST
1730 IF  K$  =  "P"  THEN  GOSUB  1050:  REM
     PLOT  POINT
1740 IF  K$  =  "L"  THEN  GOSUB  1150:  REM
     PLOT  LINE
1750 IF  K$  =  "R"  THEN  GOSUB  1270:  REM
     PLOT  RAY
1760 IF  K$  =  "E"  THEN  GOSUB  1380:  REM
     ERASE  LAST  LINE  DRAWN
1770 IF  K$  =  "S"  THEN  GOSUB  810:  REM    C
     URSOR  SPEED  TOGGLE
1780 IF  K$  =  "C"  THEN  GOSUB  1600:  REM
     CLEAR  SCREEN  &  START  OVER
1790 GOTO  1660
1800 REM    EDITING  HEADER*****
1810 VTAB  21:  HTAB  1:  CALL  SCNCLR
1820 VTAB  22:  HTAB  2:  PRINT  "X:  ";X;:  HT
     AB  15:  PRINT  "<RETURN>  FOR  COMMAND
     LIST"
1830 HTAB  2:  PRINT  "Y:  ";Y;:  HTAB  15:  PR
     INT  "<ESC>  FOR  MENU"
1840 HTAB  2:  PRINT  "Z:  ";Z;:  HTAB  15:  PR
     INT  "<-ARROWS  MOVE  CURSOR->";
1850 IF  TYPE  =  0  THEN  RETURN
1860 ON  TYPE  GOTO  1870,1880,1890
1870 VTAB  21:  HTAB  2:  CALL  LNCLR:  PRINT
     CHR$(91);"POINT";CHR$(93):  GOTO
     1900
1880 VTAB  21:  HTAB  2:  CALL  LNCLR:  PRINT
     CHR$(91);"LINE";CHR$(93):  GOTO
     1900
1890 VTAB  21:  HTAB  2:  CALL  LNCLR:  PRINT
     CHR$(91);"RAY";CHR$(93)
1900 RETURN
1910 REM    COORDINATE  UPDATE*****
1920 ON  AXIS  GOTO  1930,1940,1950:  REM    U
     PDATE  APPROPRIATE  AXIS
1930 V  =  22:  VTAB  V:  HTAB  5:  PRINT  "  "
     ;:  HTAB  5:  PRINT  X;:  RETURN
1940 V  =  23:  VTAB  V:  HTAB  5:  PRINT  "  "
     ;:  HTAB  5:  PRINT  Y;:  RETURN
1950 V  =  24:  VTAB  V:  HTAB  5:  PRINT  "  "
     ;:  HTAB  5:  PRINT  Z;
1960 RETURN
1970 REM    COMMAND  LIST*****
1980 VTAB  21:  HTAB  1:  CALL  SCNCLR
1990 HTAB  2:  PRINT  "X,Y,Z:  SELECT  AXIS";
     :  HTAB  27:  PRINT  "C)LEAR  SCREEN"
2000 HTAB  2:  PRINT  "P)OINT  PLOT";:  HTAB
     24:  PRINT  "E)RASE  LAST  LINE"
2010 HTAB  2:  PRINT  "L)INE  PLOT";:  HTAB  2
     0:  PRINT  "S)PEED  TOGGLE"
2020 HTAB  2:  PRINT  "R)AY  PLOT";:  HTAB  14
     :  PRINT  "<RETURN>  TO  RESUME  EDITING
     ";
2030 VTAB  1:  HTAB  1:  POKE  KEYCLR,0:  GET
     K$
2040 IF  K$  =  CHR$(13)  THEN  GOSUB  1800
     :  RETURN  :  REM    BACK  TO  EDIT  MODE
2050 GOTO  2030
2060 RETURN
2070 REM    MENU  1
2080 IF  SFLAG  =  OFF  THEN  GOTO  2230:  REM
     MENU  2  IF  OBJECT  IN  MEMORY
2090 MENU  =  1:  TEXT  :  HOME  :  HTAB  11:  PR
     INT  "3-D  OBJECT  EDITOR"
2100 HTAB  11:  FOR  J  =  1  TO  17:  PRINT  "-"
     ;:  NEXT  J
2110 VTAB  6:  HTAB  1:  PRINT  "OPTIONS:"
2120 VTAB  9:  HTAB  5:  PRINT  "1)  CREATE  NE
     W  OBJECT"
2130 VTAB  11:  HTAB  5:  PRINT  "2)  LOAD  OBJ
     ECT  FROM  DISK  TO  EDIT"
2140 VTAB  13:  HTAB  5:  PRINT  "3)  RETURN  T
     O  SYSTEM  MENU"
2150 VTAB  16:  HTAB  1:  PRINT  "CHOOSE  1-3:
     ";
2160 POKE  KEYCLR,0:  GET  K$:K  =  VAL  (K$)
     :  IF  K  <  1  OR  K  >  3  THEN  2160
2170 PRINT  K$;
```

APPLE II Series

```
2180 IF K$ = "1" THEN GOSUB 2370: REM
     INITIALIZE
2190 IF K$ = "2" THEN GOSUB 2440: GOTO
     2080: REM LOAD OBJECT FROM DISK
     FIRST
2200 IF K$ = "3" THEN GOSUB 3350: REM
     SYSTEM MENU
2210 GOSUB 1800: GOSUB 1650: REM EDIT H
     EADER, MAIN DRIVER
2220 GOTO 2080
2230 REM MENU ******
2240 MENU = 2: TEXT : HOME : HTAB 11: PR
     INT "3-D OBJECT EDITOR"
2250 HTAB 11: FOR J = 1 TO 17: PRINT "-"
     ;: NEXT J
2260 VTAB 6: HTAB 1: PRINT "OPTIONS:"
2270 VTAB 9: HTAB 5: PRINT "1) RESUME ED
     ITING OBJECT IN MEMORY"
2280 VTAB 11: HTAB 5: PRINT "2) SAVE OBJ
     ECT IN MEMORY TO DISK"
2290 VTAB 13: HTAB 5: PRINT "3) RETURN T
     O SYSTEM MENU"
2300 VTAB 16: HTAB 1: PRINT "CHOOSE 1-3:
     ";
2310 POKE KEYCLR,0: GET K$:K = VAL (K$)
     : IF K < 1 OR K > 3 THEN 2310
2320 PRINT K$;
2330 IF K$ = "1" THEN GOSUB 2370: GOSUB
     1800: GOSUB 1650
2340 IF K$ = "2" THEN GOSUB 2700
2350 IF K$ = "3" THEN GOSUB 3350
2360 GOTO 2240
2370 REM INITIALIZE EDITOR******
2380 IF XE%(1) > 0 AND YE%(1) > 0 THEN
     POKE SWITCH,0: GOSUB 720: GOTO 2420
     : REM IF OBJECT IN MEMORY, SWITCH
     TO EDIT SCREEN WITHOUT HGR ERASE
2390 GOSUB 1560: REM SCREEN INITIALIZ
     E
2400 X = 80:Y = 40:Z = 0
2410 P = 0:N = 0:C = 0:AXIS = 1:TYPE = 0
2420 GOSUB 720: REM DRAW CURSOR
2430 RETURN
2440 REM LOAD FROM DISK*****
2450 ONERR GOTO 3240: REM DISK ERROR
     -HANDLING ROUTINE
2460 MODE = 1
2470 PRINT : PRINT : PRINT "PRESS 'C' TO
     CATALOG, 'A' TO ABORT
2480 PRINT : INPUT "FILENAME TO LOAD: ";
     NAME$
2490 IF NAME$ = "C" THEN PRINT D$;"CATA
     LOG": GOTO 2450
2500 IF NAME$ = "A" THEN RETURN
2510 FLNAME$ = NAME$ + ".ED"
2520 PRINT D$;"LOAD ";FLNAME$: REM FORC
     ES ERROR TO CONFIRM FILE PRESENT ON
     DISK BEFORE OPENING
2530 PRINT D$;"OPEN ";FLNAME$
2540 PRINT D$;"READ ";FLNAME$
2550 INPUT P: INPUT C: INPUT N
2560 FOR J = 1 TO P
2570 FOR K = 1 TO 5
2580 INPUT XYZ%(J,K)
2590 NEXT K
2600 NEXT J
2610 FOR J = 1 TO C
2620 INPUT XE%(J): INPUT YE%(J)
2630 NEXT J
2640 PRINT D$;"CLOSE ";FLNAME$
2650 SFLAG = OFF
2660 GOSUB 190: REM DRAW OBJECT LOADED
2670 X = XYZ%(P,1):Y = XYZ%(P,2):Z = XYZ
     %(P,3)
2680 GOSUB 720: REM CURSOR
2690 GOTO 2210: REM MUST USE GOTO (ONE
     RR POPPED RETURN OFF STACK)
2700 REM SAVE TO DISK******
2710 ONERR GOTO 3240
2720 MODE = 2
2730 PRINT : PRINT : PRINT "PRESS 'C' TO
     CATALOG, 'A' TO ABORT"
2740 PRINT : PRINT : INPUT "FILENAME TO
     SAVE: ";NAME$
2750 IF NAME$ = "C" THEN PRINT D$;"CATA
     LOG": GOTO 2710
2760 IF NAME$ = "A" THEN RETURN
2770 PRINT : PRINT "PRESS 'E' TO SAVE ED
     IT FILE ONLY"
2780 PRINT "ANY OTHER KEY TO SAVE ALL FI
     LES"
2790 POKE KEYCLR,0: GET K$: IF K$ = "E"
     THEN PRINT : PRINT "SAVING EDIT FI
     LE TO DISK": GOTO 3070
2800 PRINT : PRINT "SAVING FILES TO DISK
     "
2810 FLNAME$ = NAME$ + ".XYZ"
2820 PRINT D$;"OPEN ";FLNAME$
2830 PRINT D$;"DELETE";FLNAME$: REM DEL
     ETES PREVIOUS FILE, IF ANY
2840 PRINT D$;"OPEN ";FLNAME$
2850 PRINT D$;"WRITE ";FLNAME$
2860 PRINT N: REM NUMBER OF UNIQUE POI
     NTS
2870 I = 1
```

APPLE II Series

```
2880 FOR J = 1 TO P
2890 IF XYZ%(J,4) < I THEN GOTO 2940: R
     EM SKIP DUPLICATE POINTS
2900 FOR K = 1 TO 3
2910 PRINT XYZ%(J,K)
2920 NEXT K
2930 I = I + 1
2940 NEXT J
2950 PRINT D$;"CLOSE ";FLNAME$
2960 FLNAME$ = NAME$ + ".PC"
2970 PRINT D$;"OPEN ";FLNAME$
2980 PRINT D$;"DELETE";FLNAME$
2990 PRINT D$;"OPEN ";FLNAME$
3000 PRINT D$;"WRITE ";FLNAME$
3010 PRINT C/2: REM LINECOUNT
3020 GOSUB 510: REM FILL PC%() ARRAY
3030 FOR J = 1 TO C
3040 PRINT PC%(J)
3050 NEXT J
3060 PRINT D$;"CLOSE ";FLNAME$
3070 FLNAME$ = NAME$ + ".ED"
3080 PRINT D$;"OPEN ";FLNAME$
3090 PRINT D$;"DELETE";FLNAME$
3100 PRINT D$;"OPEN ";FLNAME$
3110 PRINT D$;"WRITE ";FLNAME$
3120 PRINT P: PRINT C: PRINT N
3130 FOR J = 1 TO P
3140 FOR K = 1 TO 5
3150 PRINT XYZ%(J,K)
3160 NEXT K
3170 NEXT J
3180 FOR J = 1 TO C
3190 PRINT XE%(J): PRINT YE%(J)
3200 NEXT J
3210 PRINT D$;"CLOSE ";FLNAME$
3220 IF PEEK (222) = 255 THEN RETURN :
     REM NO DISK ERROR
3230 GOTO 2230: REM MENU2
3240 REM DISK ERROR-HANDLING******
3250 ERR = PEEK (222): POKE 216,0: REM
     DETERMINE ERROR #, RESET ERROR F
     LAG
3260 PRINT
3270 IF ERR = 13 THEN GOTO 2530: REM F
     ILE EXISTS
3280 IF ERR = 6 THEN PRINT NAME$;" NOT
     FOUND ON DISK"
3290 IF ERR = 8 THEN PRINT "NO DISK IN
     DRIVE OR DRIVE DOOR OPEN"
3300 IF ERR = 9 THEN PRINT "DISK FULL"
3310 PRINT : PRINT "<RETURN> TO TRY AGAI
     N>, <ESC> TO ABORT";: POKE KEYCLR,
     0: GET K$: PRINT
3320 IF K$ = CHR$ (27) THEN ON MENU GO
     TO 2070,2230: REM RETURN TO CALLIN
     G MENU
3330 IF K$ = CHR$ (13) THEN ON MODE GO
     TO 2440,2700: REM RETURN TO DISK S
     AVE OR LOAD FOR ANOTHER ATTEMPT
3340 GOTO 3310: REM MUST BE GOTO; BUG I
     N APPLESOFT'S RESUME COMMAND
3350 REM RETURN TO MENU*****
3360 IF XE%(1) = 0 AND YE%(1) = 0 THEN
     GOTO 3420
3370 PRINT : PRINT : PRINT "DO YOU WANT
     TO SAVE OBJECT TO DISK? Y";: PRINT
     CHR$ (8);
3380 POKE KEYCLR,0: GET K$: PRINT K$;
3390 IF K$ = "Y" OR K$ = CHR$ (13) THEN
     GOSUB 2700: IF NAME$ = "A" THEN
     RETURN
3400 IF K$ = "N" THEN GOTO 3420
3410 GOTO 3380
3420 PRINT D$;"RUN MENU"
3430 END
3440 REM CREATE CURSOR SHAPE******
3450 FOR MEM = 3940 TO 3963
3460 READ CS: POKE MEM,CS
3470 NEXT MEM
3480 RESTORE
3490 POKE 232,100: POKE 233,15: REM A
     PPLESOFT'S POINTERS TO START OF SHA
     PE TABLE (ADDRESS 3940)
3500 ROT = 0: SCALE = 1
3510 DATA 1,206,4,0,146,18,36,4,128,128
     ,128,128,128,128,48,182,218,219,45,
     77,77,45,5,0: REM CURSOR SHAPE TAB
     LE
3520 RETURN
3530 REM INITIALIZE******
3540 SIZE = 100
3550 DIM XYZ%(SIZE,5)
3560 DIM PC%(SIZE * 3)
3570 DIM XE%(SIZE * 3),YE%(SIZE * 3)
3580 D$ = CHR$ (4): BEEP$ = CHR$ (7): SC
     NCLR = -958: LNCLR = -868: KEYCLR
     = -16368: SWITCH = -16304
3590 HCOLOR= 3: INC = 10: SET = 1: OFF = 0:
     SFLAG = SET
3600 AXIS = ASC ("X") - 87: REM START
     ON X-AXIS
3610 GOSUB 3440: REM POKE CURSOR SHAPE
     TABLE IN MEMORY
3620 GOTO 2070: REM MENU 1
3630 RETURN
```

HCM

Conversation . . . from p. 155

to participate. So from our standpoint, the feeling was that you had to provide a way for the mother or father to help in an interactive way. And the size was critical. In fact, we wanted to make it even larger, but retail space and other considerations limited us to the current size.

HCM: There are a number of touch-sensitive pads on the market now. How do you like the way yours compares with the others?

RR: We use a digital surface, as opposed to an analog resistance circuit. Analog surfaces are less expensive as long as they remain small. The digital surface permits a larger pad and allows, say in a drawing mode, for 10 fingers to be placed on the surface and read at once by the software. You cannot do that with an analog surface.

HCM: Does the Koala Pad use this technology?

RR: Well, the Koala Pad uses a membrane, but in a slightly different manner. You're dealing essentially with two sheets of aluminum foil that come into contact. With the PowerPad you're dealing with multiple lines drawn on the surfaces, and it's the closure of two lines that indicates position.

HCM: Are the expensive Houston Instruments and Summa Graphics digitizers analog circuits?

RR: No, what they employ is typically a radio transmitter in a stylus, and the pad is an antenna. You can only indicate one point at a time. These pads are built along the lines of the old Apple digitizer tablet.

HCM: How do marketing strategies differ, if they do, between your product and the other digitizing pads?

RR: In terms of last year's strategies, I don't think we differed significantly, but I have the feeling that we're beginning to take a different direction. From what I understand, the rest of the market's orientation is going to be towards computer graphics and higher-end computer products, while initially for 1984, our focus is going to be towards lower-cost computers for children with a much more educational, multi-subject emphasis. Koala, for example, has a nice little product, but their emphasis is on graphics, as is Atari's.

HCM: Do you expect any competition from computer manufacturers themselves?

RR: Commodore may get into the field. They have approached both Koala and us about some sort of deal. We haven't entered into one, and I don't think Koala has. I have heard rumblings about IBM negotiating a deal with a touch-tablet supplier, so there may be an announcement from IBM forthcoming.

HCM: What is your attitude towards third-party software developers, those who would like to write packages for your pad?

RR: We're encouraging them. Two companies are developing packages right now, and we've provided information to at least 12 or 13 other firms on how to develop software for the PowerPad.

HCM: We've looked at four of your software packages—three for the C-64 and one for the Apple. Do you have any more packages coming along?

RR: We have nine on the market for the C-64.

HCM: How about for the IBM PC and PCjr?

RR: They're under release for late April.

HCM: We particularly liked the graphics that the Apple Micro Illustrator package produces. Is it possible to import those graphics from a disk into a BASIC program for title screens?

RR: On the Micro Illustrator package you can't do that, but on our new Super Graphics package, which is directed toward the IBM marketplace and which will be available shortly on the Apple and the C-64, you indeed can do that.

HCM: What about repair service? Will units have to be sent to your headquarters in Georgia?

RR: Distribution is through both mass merchandising channels and through distributors into local computer stores. Busell, MicroD, Keystone, handle the product, but units will have to be returned to Georgia for service.

HCM: The pad overlay seems a bit delicate. Have you done any studies that would indicate how it will stand up to children's use?

RR: There is no question that after a while the overlay frames can be broken. Kids can step on them, despite the best of intentions. The overlay itself, though, which is polycarb, is very tough and almost impossible to rip or puncture, so even if it loses its frame it can still be placed on the pad and can perform its function. We will also replace them for a nominal charge.

HCM: Are any steps being contemplated to strengthen the cord connection to the Apple? I have visions of a child jerking motherboard components right out of the machine.

RR: There will be a different type of connector offered in the spring.

HCM: The manual says that inserting a telephone connector into the pad may permanently damage it. I wonder if an inquisitive child might do this.

RR: The connectors are of the same type, but inserting the telephone connector won't damage the pad. There's not enough current in a telephone line to do anything, and the wires in each connector don't match up in any case.

HCM: You mention that the board is strong enough to withstand a child's weight. Does that mean a child can stand on the sensor surface without damaging it?

RR: Yes. My daughter, in fact, has a propensity for jumping up and down on it, despite my protestations. It still does work.

PowerPad . . . from p. 155

much to his playing partners' glee. He settled down, though, for a one-stroke victory.

This package is a lot of fun, in spite of some awkward moments in the software. *Leo's Links* and *Pinball Construction Set* (also reviewed in this issue) are part of an interesting trend in software: You build your game as well as play it.

The object of *LogicMaster*, the third C-64 package, is to deduce the shape, position, and color of the symbols in a hidden line of code. The symbols may be clubs, spades, diamonds, and hearts, and the positions and colors vary. Clues given after each guess tell you how many items in your guess are the correct color, how many items are the correct color and in the correct position, how many items in your guess are the correct shape, or how many items are the correct shape and in the right position, and then you're on your own. Pumping symbols probably builds up one's deductors, but the package seems a little severe and narrowly focused. It does, however, work well.

The software packages on the whole are attractive although they do have the look of being rushed a bit to market. The manuals have a few typographical errors, and the software differs from the manual descriptions in various minor ways. The software is also just a bit clumsy in spots, as if hurriedly written. Chalk Board predicts that the home computer peripheral market will expand to around $6 billion by 1986, and perhaps the thought of $6 billion is enough to make anyone hurry a bit.

Still, this is an innovative product. A computer writer predicted last year that in the future, people are going to be less interested in "computers" and more interested in "machines that do things." Maybe even now the word "computer" is too narrow because for most people it's associated with abstraction, numbers, screen displays. . . Perhaps "robot" is a better all-around description.

We may have only just begun to tap the potential of these machines to see, move, speak, hear, and touch. And touch-sensitive pads are a move in an interesting direction. HCM

Index to Advertisers

Abbey Tape Duplicators....52
Allman Systems....55
American Software Design & Distribution Co....104
Apropos Technology....80
Atarisoft....3
Axiom Corporation....156
Bach Co., The....56
Basic Home Computing....54
Bel Ayr Software....94
Ben Hur Software....54
Best Software....61
Bizware, Inc....62
Borland International....50
Bradbury, Peter....105
CMS Systems....118
CR Distributing....49
Casas Adobes Software....94
Centroplex Computer....23
Commodore Computer Systems Division....164
CompuAdd Corp....48
CompuWare Inc....61
Cumberland Technology....92
D & D Publishing Co....145
Dean Software....68
Dennison Computer Supplies, Inc....2
Dow, John T....77
EASy....37, 70
Educational Software Products....134
El Dorado Trading Group....38
Emerald Valley Publishing Co....124
Euroware....54
Extended Software Company....73
Federal Hill Software....92
Gary Phillips and Associates....96
Genesis Star Software....136
Harris House Software....86
Hayden Book Company....13
Heritage Computer Services....135
Home Computer Magazine....106, 145, 147, 162
Information Associates....120
Information Research Association, Inc. (Infra)....79
J & K H Software....45
KCR Corporation....136
K. R. Rullman Company....134
Kaleidoscope Programs....67
KIDware....53
Konrad Komputerware....136
Letendre, Bruce....104
Magnum Programs....144
Micromagic Software....72
Mikel Laboratories, Inc....64
Millers Graphics....95
Mok, Berniel....104
Morning Star Software....131
MULTICOM, Inc....68
Myarc, Inc....70
Navarone Industries....69, 88
Newport Controls....93
Nebula Software....53
Not Polyoptics....77
Parallel Systems....79
Parker Brothers....9
Platinum City Software....55
RB's Software....55
RC Distributing Company....16
Realization Software....45
Ro-Cal Associates....60
Rocketman....52
SST Software, Inc....52
Scott CompWare....60
Siren Software....72
Smith-Ware....72
Soft Art, Inc....35
Softext, Inc....18, 19
Softies, The....95
Software Advances....65
Software Support, Inc....128
Sprite-O-Lite....77
Star Micronics....163
Sun City Software....92
SunWare, Ltd....22
Supply Co....135
Symbiotech, Inc....49
Systems Interface....134
TBR, Inc....95
TSS Software....53
Tenex Software....25, 76
Tex-Comp....29, 46, 140, 150
TImagination....144
TImore Products....45
TI Soft....66
Unisource Electronics, Inc....85
User-Happy Simulations....94
VMC Software....105
Whiplash Software....120
World Class Software....89
Woodinville Computer Center Inc....133
York 10....119
99/4(A) Program Exchange....87
99'er-WARE....125

Home Computer Magazine is not responsible for errors or omissions in this *Index to Advertisers*. The listing is merely provided as a convenience for our readers.

HCM: Has the development of the PowerPad and related software been a major project? What sort of time has it taken?

RR: I almost hesitate to tell you how quickly the hardware was developed. The first software packages took us about seven weeks from inception to out-the-door. There are some aspects other than the initial software and hardware that took a lot longer. For instance, after a period of time, usually between 1 million and 10 million lives, any touch sensitive membrane will begin to wear. The dielectric spacers on any membrane, whether digital or analog, will begin to wear down. When it wears down and collapses on an analog membrane like a Koala, you're dead in the water, but we have a digital surface, and we've built a fairly extensive error checking and error correcting routine into our software that can keep the surface functioning even after wear has begun.

HCM: Are there any new software developments you see yourself utilizing?

RR: Oh yes, there will be additional enhancements both to the hardware and software announced in the spring.

HCM: Are there any new directions or products you see yourself moving to?

RR: There are several areas that we are actively investigating. One is what other things we need to do to make it easier for the user to interface with the computer. We've thought of using an infrared link like the PCjr's keyboard has. We will be considering other ways of getting information into the computer. Obviously Koala was also thinking the same way when they went to the light pen. We have a whole bunch of stuff floating around the back room that we're playing with. We will be developing other devices for getting information into the computer.

HCM: Do you see any software applications outside the entertainment-education venue?

RR: Yes, for both the Micro Illustrator graphics package and Super Graphics. Super Graphics is being used by several commercial artists here in Atlanta on a test basis for doing computer art and for package work. It looks highly successful, especially when coupled with a color output plotter or a printer. And at the CES we demonstrated a graphics package coupled with an ink-jet printer that has some business applications. Though our thrust is primarily toward the home and educational market, we will be producing a combination of things for the professional as well. HCM

HCM Program Bug

DeBUGS on DisPlay

Bugs—problems that cause programs not to function properly—are an inevitable consequence of programming. Programs are complex systems that are difficult to test exhaustively. And occasionally bugs slip through even the best testing procedures. When such a bug in one of our programs comes to our attention, we print a correction in this column so that our readers can correct their programs as soon as possible.

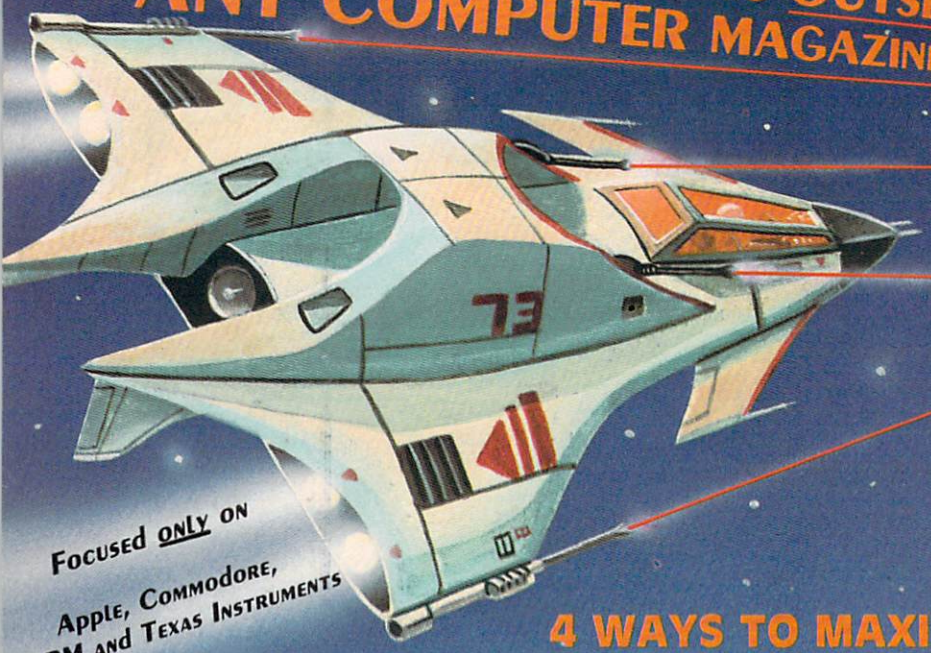
In the article *Movable Feast* from the last issue, a bug was found in the *Menu Planner* program at line **170**. Change the last portion of the line which reads:

:IF B>A THEN B=A−1 to read as follows: **:IF B>A THEN B=A:A=A+1**

This will allow the program to print out the matching recipes under all conditions of selection.

Readers have informed us that the article in last issue entitled *Challenging the Tower of Hanoi* lacked the instructions on how to move a ring from one post to another via the computer keyboard. The answer is quite simple: To select the post to move the selected ring from, enter the post number (1, 2, or 3). Following that single key entry, the post to move to is entered as a single key stroke too (1, 2, or 3). See, as easy as 1,2 or 3,1 or 1,3. . .

LASER-TARGETED TO OUTSELL ANY COMPUTER MAGAZINE...



FOCUSED ONLY ON APPLE, COMMODORE, IBM AND TEXAS INSTRUMENTS

4 WAYS TO MAXIMIZE PROFITS FROM COMPUTER MAGAZINE SALES

ATTENTION: READERS WE NEED YOUR HELP!
Please show this page to any magazine seller who doesn't carry HCM

1 **Display Magazines Demanded By The High-Growth Segment Of The Market.**

The home computer explosion of the 80's focuses on 4 brands: Apple, Commodore, IBM, and Texas Instruments. These systems have already attracted over six million users—a figure that will quadruple by 1986. Vendors who supply magazines to this group will make the most money.

2 **Feature Magazines That Elicit Repeat Sales.**

The right selection and the right balance of magazines is a powerful draw. Not only will it attract buyers and add to your profits, it will also create one of the most important elements for success—store loyalty.

3 **Promote Magazines That Ensure Buyer Satisfaction.**

Readers want facts—not fluff! They want magazines that do more than just talk about computing. . . they want publications filled with in-depth "How-To" articles, serialized features, reliable product reviews, and plenty of free software ready to type-in-and-run.

4 **Start Selling *Home Computer Magazine* Today!**

No other magazine focuses exclusively on the leading home computer brands—Apple, Commodore, IBM, and Texas Instruments. It's the magazine designed to appeal to millions of current users as well as millions of future home computer buyers. *Home Computer Magazine's* popular programs, colorful graphics, valuable reference articles, and in-depth product reviews will keep your customers coming back again and again. We invite you to examine a copy of *Home Computer Magazine*, and measure it against any contender.

ONCE YOU COMPARE — THERE'S NO COMPARISON.℠

HOME COMPUTER™ magazine

Contact our Dealer Sales Department: 1500 Valley River Drive, Suite 250 • Eugene, OR 97401 • Tel. (503) 485-8796

One-Of-A-Kind

Page for page, *Home Computer Magazine* provides the unique combination of resources perfectly suited to your style of home computing.

- Dozens of top quality key-in-and-run programs for Apple, Commodore, IBM, and TI home computers appear in each issue.
- Self-paced computer literacy lessons expand your opportunities for education, recreation, and home productivity.
- A nationally-acclaimed review department keeps you informed of the latest exciting products.
- Arcade-quality games, ready to key-in-and-play!

FREE SOFTWARE
with your subscription

Subscribe today, and with your paid subscription you will receive FREE software—ON TAPE™ or ON DISK™—the same high-quality programs published monthly in the magazine. This cassette tape or floppy disk program service—normally a $3.95 per month extra cost—is the convenient, accurate and affordable way to save hundreds of typing hours.

With a 1-year payment, you'll get 2 months of programs ON TAPE or ON DISK. Subscribe for 2 years and you'll receive 4 months. And with a 3-year subscription, we'll give you 6 months of this useful and valuable software.

SUBSCRIBE TODAY

Please Print NEW SUBSCRIBER OFFER

Name _____

Address _____

City _____ State _____ Zip _____

Select your choice of media: ☐ ON TAPE ☐ ON DISK

Indicate which machine the media is for: (Pick one)

☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A

Choose desired subscription term:

1-yr ☐ $25—12 issues + 2 months ON TAPE / ON DISK

2-yr ☐ $45—24 issues + 4 months ON TAPE / ON DISK

3-yr ☐ $63—36 issues + 6 months ON TAPE / ON DISK

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription only. Free software not available outside USA.

☐ Check or Money Order Enclosed. Must be in US funds drawn on a US bank.

Bill my ☐ VISA ☐ MasterCard

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Allow 6-8 weeks for your first issue.

Account No.

Signature _____ Exp. Date _____

SATISFACTION GUARANTEED—or the unfilled portion of your subscription will be refunded, less the cost of any premium items you have received. MSO

SAVE OVER 30%
Off The Single-Copy Price And Receive
FREE SOFTWARE TO BOOT!

ORDER TOLL-FREE
1-800-828-2212
VISA / MasterCard Orders Only

In Oregon, Alaska, Hawaii-Tel. (503)485-8796

SUBSCRIBE TODAY

Please Print NEW SUBSCRIBER OFFER

Name _____

Address _____

City _____ State _____ Zip _____

Select your choice of media: ☐ ON TAPE ☐ ON DISK

Indicate which machine the media is for: (Pick one)

☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A

Choose desired subscription term:

1-yr ☐ $25—12 issues + 2 months ON TAPE / ON DISK

2-yr ☐ $45—24 issues + 4 months ON TAPE / ON DISK

3-yr ☐ $63—36 issues + 6 months ON TAPE / ON DISK

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription only. Free software not available outside USA.

☐ Check or Money Order Enclosed. Must be in US funds drawn on a US bank.

Bill my ☐ VISA ☐ MasterCard

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Allow 6-8 weeks for your first issue.

Account No.

Signature _____ Exp. Date _____

SATISFACTION GUARANTEED—or the unfilled portion of your subscription will be refunded, less the cost of any premium items you have received. MSO

PLACE FIRST CLASS POSTAGE STAMP (CANADIANS PLEASE AFFIX 37¢ POSTAGE

HOME COMPUTER™ magazine
P.O. Box 5537
Eugene, OR 97405

Subscribe Today

SAVE OVER 30%

Off The Single-Copy Price

AND RECEIVE

FREE SOFTWARE TO BOOT!

(See reverse side for details)

PLACE FIRST CLASS POSTAGE STAMP (CANADIANS PLEASE AFFIX 37¢ POSTAGE

HOME COMPUTER™ magazine
P.O. Box 5537
Eugene, OR 97405

Come Discover A Truly Unique Magazine

Inside You will find:

- EDUCATION
- RECREATION
- INFORMATION
- ILLUMINATION

AND ABOVE ALL

- COMPU-PRESTIDIGITATION*

***compu-prestidigitation**
(kŏm·pū·prĕs·teh·dĭ·jeh·tā·shŭn)—*n.* **1.** The magical quality of unexpected comprehension that results from presenting technical information about computers in a lively, entertaining, visually attractive and easy-to-understand format. **2.** The magical tricks that make a computer sing, dance, and do all sorts of wonderfully useful things.

HOME COMPUTER MAGAZINE'S SENSATIONAL SOFTWARE GIVEAWAY

YOUR CHOICE OF THE PROGRAMS FROM THIS ISSUE AT THE UNBELIEVABLE LOW PRICE OF ONLY $3.95!

PROOF OF PURCHASE NECESSARY

To participate in our monthly Software Giveaway, you need to be a bona-fide purchaser of *Home Computer Magazine* and follow the directions on this form.

PLEASE PRINT

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

Have you taken advantage of our Software Giveaway before? ☐ Yes ☐ No

Indicate your choice of media: ☐ ON TAPE™ ☐ ON DISK™

Indicate which machine the media is for: (Pick one)
☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A

Fill in this magazine's *Volume No.* ___ and *Issue No.* ___. You will receive all programs (which have versions for your selected machine) whose listings appear in this issue. Please also list, in order of preference, these same programs—helping us determine future offerings.

1. _____ 4. _____
2. _____ 5. _____
3. _____ 6. _____

IMPORTANT: Complete the questionnaire on the reverse side. Tear out this complete page and enclose in an envelope along with $3.95. Payment must be made by check, money order or VISA/MasterCard. Proof of purchase (subscriber label/number, sales receipt or reasonable facsimile thereof) must also accompany this form.

☐ Questionnaire is completed.
☐ Enclosed is check or money order for $3.95.
☐ Enclosed is my proof of purchase.
☐ Bill my ☐ VISA ☐ MasterCard

Account No.

OFFER GOOD ONLY IN USA
(Canada & Mexico please inquire)

Signature _____ Exp. Date _____

Mail this page to:
Software Giveaway
Home Computer Magazine
P.O. Box 5537
Eugene, OR 97405

TO PROPERLY HANDLE THE VOLUME OF SOFTWARE REQUESTS, ORDERS MUST BE SENT IN ON THIS FORM, NO PHOTO COPIES. SORRY, WE CANNOT ACCEPT TELEPHONE ORDERS FOR THIS SERVICE.

SG

ON TAPE / ON DISK—$3.95 each issue.

ABSOLUTELY FREE!
2-6 MONTHS OF MAGAZINE SOFTWARE ON TAPE™ OR ON DISK™ ARE FREE WITH YOUR PAID SUBSCRIPTION

With a 1-year subscription to *Home Computer Magazine* you get 2 FREE months of ON TAPE or ON DISK. Subscribe for 2 years and receive 4 months. And with a 3-year subscription we'll give you 6 full months of this convenient software on cassette tape or floppy disk.

NEW SUBSCRIBER OFFER
(Valid in USA only)

PLEASE PRINT

Name _____

Address _____

City _____ State _____ Zip _____

☐ Check or Money Order Enclosed
MUST BE US FUNDS DRAWN ON A US BANK

Bill my ☐ VISA ☐ MasterCard

Account No.

Signature _____ Exp. Date _____

SATISFACTION GUARANTEED—or the unfilled portion of your subscription will be refunded, less the cost of any premium items you have received.

Indicate your choice of media: ☐ ON TAPE ☐ ON DISK
Indicate which computer media is for: (Pick one)
☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A
Choose desired subscription term:
1-yr ☐ $25—12 issues + 2 months ON TAPE / ON DISK
2-yr ☐ $45—24 issues + 4 months ON TAPE / ON DISK
3-yr ☐ $63—36 issues + 6 months ON TAPE / ON DISK

Enclose payment or credit card information and mail with completed form to:
Home Computer Magazine
P.O. Box 5537 • Eugene, OR 97405
Or use our TOLL-FREE ORDER LINE*
FOR VISA/MasterCard orders only: 1-800-828-2212
In Oregon, Alaska, Hawaii Tel. (503)485-8796
*Please make reference to this ad when placing phone orders.
Allow 6-8 weeks for your first issue.

SGSO

HOME COMPUTER™ magazine

QUESTIONNAIRE

Complete and mail to: Home Computer Magazine • P.O. Box 5537 • Eugene, Oregon 97405

FOR ALL READERS

1. **Where did you obtain this copy of Home Computer Magazine?** ☐Subscriber ☐Supermarket ☐Bookstore ☐Users group ☐Newsstand ☐Computer Store ☐Friend ☐Library ☐Other _____
2. **What types of software are you most interested in?** ☐Educational ☐Entertainment ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____
3. **Are you** ☐Male ☐Female ☐14 or younger ☐15-24 ☐25-34 ☐35-44 ☐45-54 ☐55+
4. **Annual Household Income?** ☐Under $10,000 ☐$10,000-$14,999 ☐$15,000-$19,999 ☐$20,000-$24,999 ☐$25,000-$29,999 ☐$30,000-$39,999 ☐$40,000-$49,999 ☐$50,000+
5. **Occupation?** ☐Professional ☐Management ☐Teacher ☐Student ☐Other _____
6. **What is your ZIP code?** ☐☐☐☐☐
7. **What is the current month and year?** _____
8. **Do you presently own a Home Computer?** ☐No ☐Yes. It is a ☐TI-99/4A ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐Other _____

FOR READERS WHO PLAN TO BUY A HOME COMPUTER

9. **Which model do you think you'll purchase?** ☐Apple IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
10. **When do you expect that purchase to be?** ☐less than 3 months ☐3-6 months ☐7-12 months ☐at least 1 year
11. **What do you anticipate your primary use of a home computer will be?** ☐Entertainment ☐Education ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____

FOR PRESENT HOME COMPUTER USERS

12. **Which home computer(s) do you currently own?** ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
13. **What is the primary use of your home computer?** ☐Entertainment ☐Education ☐Computer Literacy ☐Business ☐Job-Related Applications ☐Household Management ☐Other _____
14. **How often is your computer in use?** ☐Less than 1 hour per week ☐1-4 hours ☐5-10 hours ☐11-15 hours ☐16-20 hours ☐over 20 hours
15. **On the average, about how many program listings in each issue of HCM do you key into your computer and use?** ☐None ☐1 ☐2 or 3 ☐4 or more
16. **What peripherals do you currently use?** ☐Disk System ☐Printer ☐Modem ☐Monochrome/Color Monitor ☐Other _____
17. **What do you expect to buy within the next year?** ☐Software ☐Disk system ☐Printer ☐Modem ☐Books ☐Magnetic Media ☐Monochrome/Color Monitor ☐Furniture & Accessories
18. **How much do you expect to spend on computer-related products during the next year?** ☐Less than $25 ☐$25-$49 ☐$50-$99 ☐$100-$249 ☐$250-$490 ☐$500-$999 ☐$1000-$2499 ☐$2500 or more

OPTIONAL: If you would like to help us by participating in a telephone interview, please include your telephone number (____) -_____ here and the most convenient time you can be reached _____:_____ ☐AM ☐PM

Be confident because your PowerType Daisywheel correspondence becomes you.

When you can't personally be there, only the clearest typed correspondence should be your substitute. With a PowerType Daisywheel printer your documents look highly professional. And so do you!

PowerType. It's "typewriter friendly." Using a simple drop-in ribbon cassette, it bi-directionally types executive quality correspondence at 18 cps with a print wheel that holds 96 flawless characters.

Designed for personal or business applications, PowerType's carriage accepts paper that ranges from letter to legal size, from fanfold to roll to cut sheet. You can set right and left margins, vertical and horizontal tabs.

Plus, of course, PowerType has both serial and parallel interfaces to enable it to connect to just about any personal or business computer.

So the next time you're going face to face through the mail, rely on PowerType. It will help you make a professional impression. And that's always very becoming.

star™
micronics·inc

THE POWER BEHIND THE PRINTED WORD.

Computer Peripherals Division
P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261 (214) 456-0052

Commodore Software– The Best Game in Town.

...Take on the world, toughen up your trigger finger and fire away...

Commodore is the best computer value in town...at home, at school and at work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

We're fast becoming the best game in town when it comes to entertainment for the whole family...and at affordable prices.

THE BEST ARCADE IN TOWN can be in your own home with our exciting, faithful reproductions of the best of Bally Midway arcade games. Our **Kickman,** (which just received a coveted "Electronic Games" award for an arcade translation) lets you steer the unicycle to catch the falling objects, as they fall quicker and quicker!!

Gorf, Lazarian, and Omega Race give you the best in classic space action against the one-eyed leviathon, the droids or the evil Empire.

In **The Wizard of Wor** you attempt to defeat the Wizard and the Warriors, fighting your way through to the end. With the new Commodore "MAGIC VOICE"... It talks back to you too!!

You commandeer the fleet at sea with our version of **Seawolf,** and become the master tactician as you battle "it out" with enemy fleet.

Clowns and **Blueprint** round out our arcade entertainment package to keep your fingers nimble and your mind in gear.

commodore COMPUTERS

First In Quality Software

See your local dealer now... He's got the best game in town... just for you.