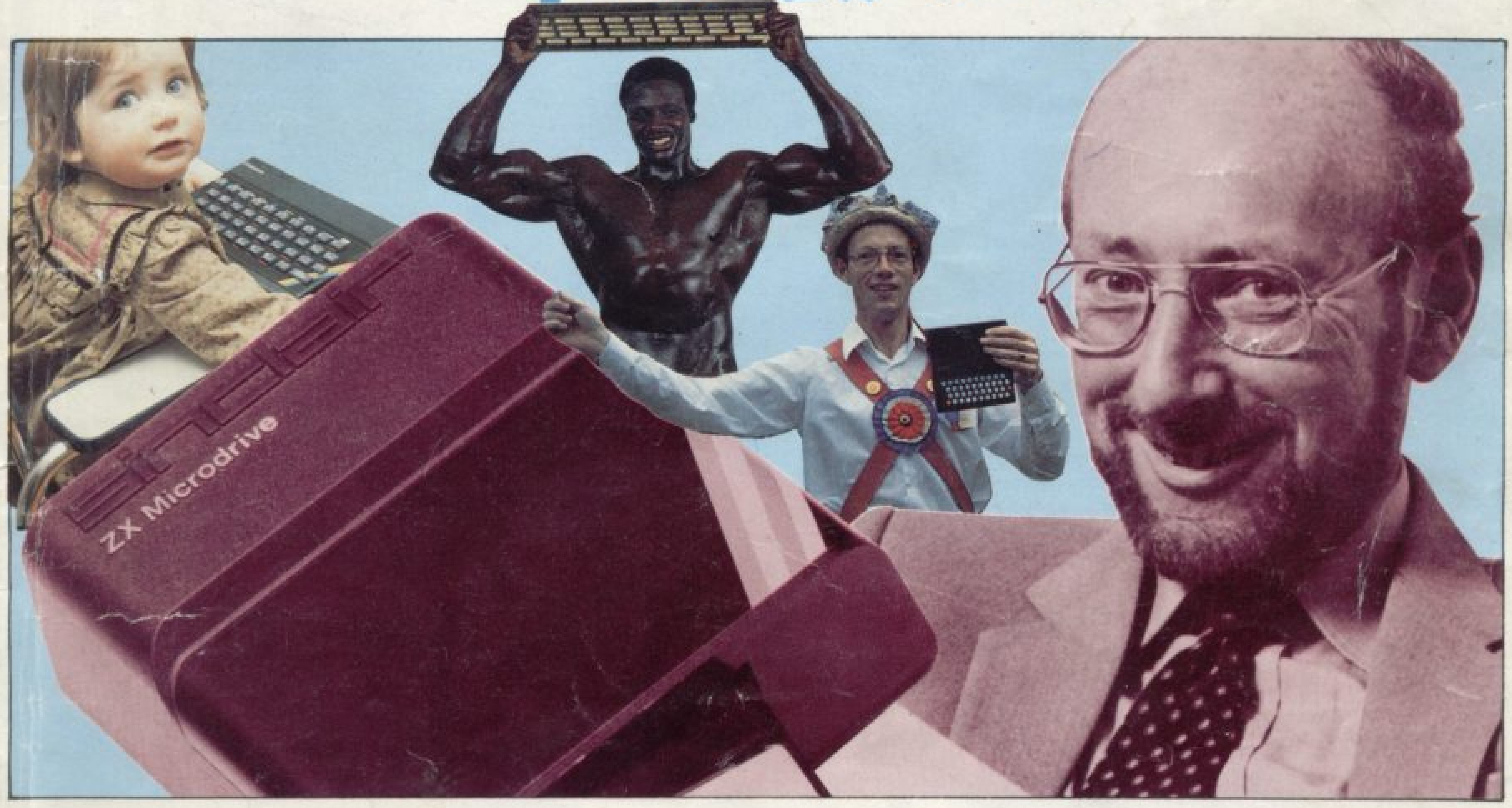
£1.95 1984

THE SECOND sinclair user ANNUAL

The Complete ZX Companion



TOP ADVICE FOR ZX OWNERS.

Buyers' guide to best software and hardware.

Sixteen pages of superb programs.

1983 REVISITED

SPOT THE DIFFERENCE!

Choosing which game to buy from the mountain available is a difficult job, especially when everyone claims to produce the best on the market. But how can you tell the best from the rest?. To help you decide, read on. . . .

What the real critics say. . . .

Very rarely have software titles produced such universal acclaim as 'Halls of the Things' and 'The Dungeon Master'. Now, with three brand new programs, Crystal continues to set the standard of software excellence. The difference is obvious - the choice is yours:

The Best or The Rest.

THE DUNGEON MASTER

Let your Spectrum be your guide in a totally new dimension in adventures in the true spirit of traditional role playing games where YOU design the scenario.

"I have been a Dungeons and Dragons fan for several years....The package provides excellent entertainment for all fans of the cults and should prove a good introduction to the game".
....Sinclair User.

"Highly recommended for its versality, originality and quality. Definately well worth obtaining".
....ZX Computing.

ZX Spectrum 48K **£7.50**

Written by Graham Stafford.

HALLS OF THE THINGS

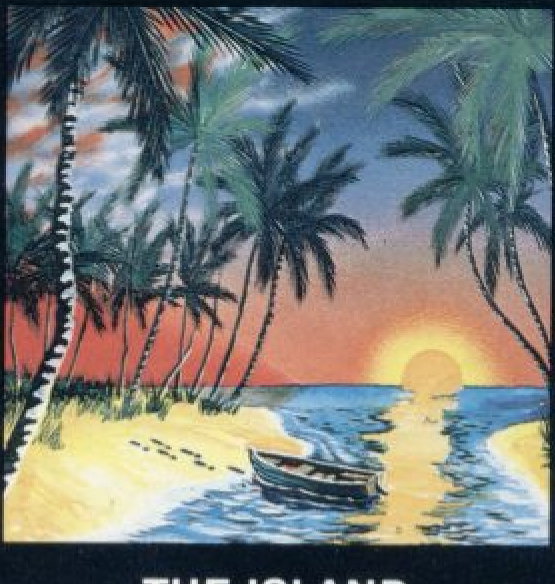
A stunning multi-level maze 'arcade - adventure'.

"Excellent and dangerously addictive - could change the Spectrum games scene overnight".
....Sinclair User.

"Spectacular - One of the best games I've seen, finely balanced between simplicity and addictiveness - superb graphics and colour - I CAN'T RECOMMEND IT HIGHLY ENOUGH".
....Popular Computing Weekly.

ZX Spectrum 48K **£7.50**

Written by Neil Mottershead, Simon Brattel and Martin Horsley.



THE ISLAND

The ultimate test of logic and deduction! Can you solve the hidden mysteries of the South Pacific Island on which you have been stranded - and escape alive! A brilliant classic style adventure game to facinate and frustrate you for months!

ZX Spectrum 48K **£7.50**

Written by Martin H. Smith.

INVASION OF THE BODY SNATCHAS!

At last! a version as fast and furious and as frustratingly addictive as the arcade original. Landers, Mutants, Bombers, Pods, Swarmers and much much more combine to produce the ultimate space game!

ZX Spectrum 48K **£6.50**

Written by Simon Brattel and Neil Mottershead.

ROMMEL'S REVENGE

A brilliant interpretation of the most visually stunning arcade game of all time. Superb high resolution 3D graphics with full perspective plus a host of new and exciting features make Rommel's Revenge the most spectacular game ever produced for your Spectrum!

ZX Spectrum 48K **£6.50**

Written by Martin Horsley.

Please send SAE for our latest catalogue and details of our forthcoming software.
Catalogue FREE with every order. P&P included.
Please add £0.50 per item for overseas orders.

Please make cheques/PO's payable to:

CRYSTAL COMPUTING
Dept SU11
2 ASHTON WAY
EAST HERRINGTON
SUNDERLAND SR3 3RX

Please Supply:
Invasion of the Body Snatchas ☐
Rommel's Revenge ☐ The Island ☐
Halls of the Things ☐ The Dungeon Master ☐
Catalogue (please enclose SAE 6in. x 9in.) ☐

I enclose cheque/PO for...

NAME...

ADDRESS...

SUA

DEALERS! For details of our excellent dealer discounts (including export) ring Chris Clarke on 061-205 6603.

PROGRAMMERS! Written any good software? send it to us for evaluation and details of our excellent royalty scheme.

THE SECOND
sinclair user
ANNUAL
The Complete ZX Companion

Managing editor
Nigel Clark
Deputy editor
Nicole Segre
Consultant editor
Mike Johnston
Managing production editor
Harold Mayes MBE
Software editor
John Gilbert
Program reviewer
Rebecca Ferguson
Illustrator/designer
Brian King
Group advertisement manager
John Ross
Sales executive
Annette Burrows
Production assistant
Dezi Epaminondou
Managing director
Terry Cartwright
Chairman
Richard Hease

Sinclair User Annual is published by ECC Publications Ltd. It is in no way connected with Sinclair Research Ltd.

Telephone
All departments
01-359 3525

If you would like to contribute to any of the Sinclair User group of publications please send programs, articles or ideas for hardware projects to:
Sinclair User and Programs,
ECC Publications,
196-200 Balls Pond,
London N1 4AQ

Programs should be on cassette and articles should be typed. We cannot undertake to return them unless a stamped-addressed envelope is included.
We pay £10 for the copyright of each program published and £50 per 1,000 words for each article used.

ISSN No 0262-5458

Printed and typeset by
Cradley Print PLC,
Warley,
West Midlands

Distributed by
Spotlight Magazine Distribution Ltd,
1 Benwell Road,
Holloway,
London N7
01-607 6411

CONTENTS

JUST AROUND THE CORNER, A NEW R

BLACK CRYSTAL

The Classic, six program adventure game for the 48K Spectrum and 16K ZX-81 computers. No software collection is complete without it.

"BLACK CRYSTAL is an excellent graphics adventure and a well thought out package" (Sinclair User, April '83).

"BLACK CRYSTAL has impressed me by its sheer quantity and generally high quality of presentation. I am afraid I have become an addict" (Home Computing Weekly, April '83).

SPECTRUM 48K 180K of program in six parts . . . only £7.50
ZX-81 16K over 100K of program in seven parts . . . only £7.50

Why pay more for less of an adventure?

ZX-81 COMPENDIUM

Alien Intruder, Wumpus Adventure, Numerology, Hangman, Hieroglyphics, Movie Mogul.

The ideal software package for all 16K ZX-81 owners. Six major programmes on two cassettes for only £6.50.

"Alien Intruder/Hieroglyphics — Both programs make good use of graphics and words to make a very entertaining package." (Sinclair User, Aug '82)

"Alien/Hieroglyphics/Wumpus/Movie — A varied mix from Carnell, all featuring imaginative responses and graphics and all of them good games for all ages (Hieroglyphics is particularly good for children)." (Popular Computing Weekly, Aug '82)

THE ADVENTURES OF ST. BERNARD

An exciting, fast moving, machine code, arcade game where you guide your intrepid St. Bernard through the perils of the icy wastelands to rescue his Mistress from the clutches of the abominable snowman. 48K Spectrum £5.95.

STARFORCE ONE

Take on the robot guardians of the central computer in a superbly stylised three dimensional battle game. 48K Spectrum £5.95.

(100% MACHINE CODE ARCADE ACTION)

THE CRYPT
written by Stephen Renton

Prepare yourself for the many challenges that shall confront you when you dare to enter "THE CRYPT".

You will battle with giant scorpions, Hell spawn, Craners, Pos-Negs and if you are unlucky enough — the Dark Cyclops in this arcade style adventure.

Available now for the 48K Spectrum @ £4.95

THE DEVIL RIDES IN

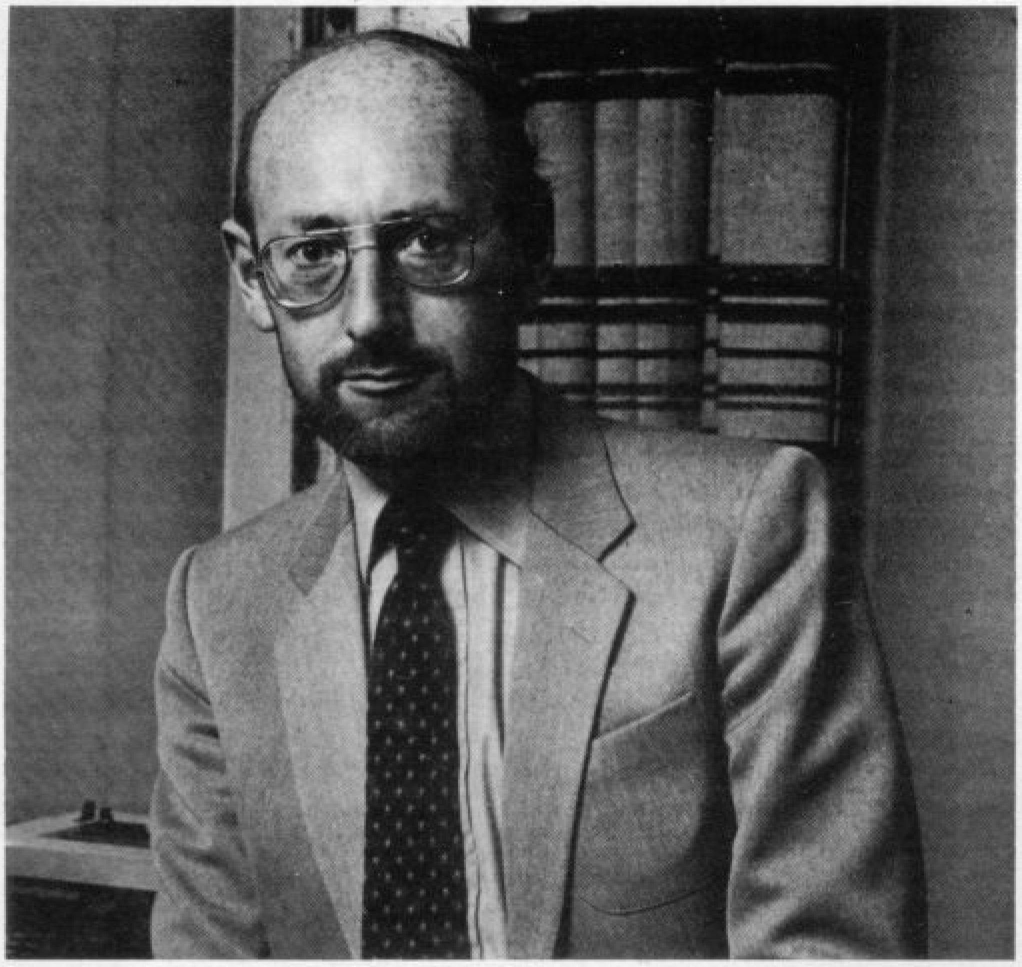
I uttered the last incantations as the clock struck thirteen. All fell silent except for a faint rustling in the corner. From out of the shadows they came, all Hell's fury against me but I was not defenceless until the Angel Of Death, astride a winged horse, joined the battle. Avoiding his bolts of hell fire, I took careful aim. My chances were slim, but if my luck held . . . 48K Spectrum £5.95.

(Fast moving, machine code, all action, Arcade game)


W RANGE FROM CARNELL SOFTWARE
COMING SOON
"THE WRATH OF MAGRA"
The first born has been destroyed. The Black Crystal of Beroth has been banished. The alliance of Evil has been defeated by the armies of Lord Fendal. So ends the Third Age. Now we invite you to write your name in the history of the Fourth Age of the Third Continent.
You will meet friends and enemies, old and new, in the long awaited sequel to Volcanic Dungeon. Using high resolution graphics and combining the best qualities of "Black Crystal" and "Volcanic Dungeon", we will allow you to become part of this tale of revenge.
"The Wrath Of Magra" comes as three, 48K programs on cassette, boxed with instruction manual and book detailing the history of the Third Continent and the many spells you will be using throughout the game.
NOTE: "The Wrath Of Magra" is a complete adventure. You need not buy "Volcanic Dungeon" or "Black Crystal" to play it.
Available through most good computer stores or direct from:
Carnell Software Ltd, North Weylands Industrial Estate, Molesey Road, Hersham, Surrey KT12 3PL.
Dealers: Contact us for your nearest wholesaler.
CARNELL SOFTWARE LTD

News

The last year has been another eventful one in the Sinclair market. The arrival of the long-awaited Microdrive and a knighthood for Clive Sinclair are only two of the items we cover in this assessment of the developments of the 12 months.



Mike Johnston reports on last year

Market reaches critical mass

IF YOU GATHER sufficient radioactive element together, at a certain point it reaches a critical mass and a chain reaction follows. Something similar seems to have happened with the density of home computers in the population early last year. Suddenly everyone seemed to want a home computer.

Computing was scarcely a minority interest even before that, with an estimated half-million home micros in use in the U.K. in 1982 but in 1983 it caught the public imagination. Computing became news.

The Sunday newspaper colour supplements contained features on computers, the daily papers started computer columns, young programmers were hailed as the new pop stars and, perhaps most fittingly, Clive Sinclair, the man who invented the home micro, received a knighthood.

While micros in general began to receive increased media coverage, Sir Clive and his company were rarely out of the news. In December, 1982 a strike at one of the Sinclair subcontracting plants, Timex in Scotland, made national headlines. In January the sale of 10 percent of the company to City investors for £12 million and the news that Sinclair Research was worth £130 million practically guaranteed more coverage.

In March Sir Clive was named Young Businessman of the Year by *The Guardian*. In April there were more industrial relations problems at Timex and the development of the flat-screen TV project was affected. The Spectrum price reduction in May made headlines as the first colour computer for less than £100 — the 16K version sold for £99 — and the June announcement in the Queen's Birthday Honours of a knighthood for Sir Clive made even bigger headlines.

Later that month the Sinclair plan for a high-technology think-tank again attracted the national press. In August, the final release of the Microdrive was just a little of an anti-climax — after 'coming shortly' for 18 months — but that was capped in September by the unveiling of the flat-screen TV, first announced even earlier.

Sales of the Spectrum soared during the period, perhaps helped by all the publicity, from 60,000 in November, 1982 to more than half-a-million by August. The ZX-81 may have done less well, although no figures have been released, despite the considerable price reductions — down to £40 in May and again down, including 16K, to £45 in August. Sinclair is continuing to support the machine but the W H Smith announcement that it would not be stocking any new titles for the ZX-81 is perhaps ominous.

Sinclair has not had the field entirely alone, as he did in the early days, and the less-than-£200 market has become increasingly crowded in the last 12 months, with some fierce price-cutting taking place as companies jockeyed for a place.

The Vic-20 fell from more than £200 to around £140 in some shops before last Christmas and other companies followed suit — Texas, Atari, Lynx, Dragon — either by reducing the price or offering a machine as part of a package including software or other hardware.

After Sinclair reductions in the price of the Spectrum in May, there was a further round of cost-cutting from other manufacturers. All of which has been good news for the consumer, who now has a much wider choice at lower prices. It has not been such good news for some companies forced to reduce their profit margins to compete with the Spectrum without the advantages large-scale production brings.

Despite the competition, the Spectrum has managed to maintain a healthy lead over its rivals and continues to top the best-selling hardware charts. By the end of March, the company was reported to have doubled its turnover from £27 million to £54 million and made a profit of £14 million before tax.

In the U.S., where competition has been even stiffer, a number of large companies, including Texas Instruments, Atari and Mattel (Aquarius) had substantial losses. Timex, which makes Sinclair lookalikes, has not done too well either, despite a promising start. The TS2000 (Spectrum+) was due for release early in the year but was delayed, then re-designed to meet the competition. Timex was forced to make heavy discounts on the TS1000 (ZX-81) to boost flagging sales.

Play with a Griffin

So they take school in their stride

Your children are using a computer at school far more often than you think.

Computers make learning an absorbing game and retain a child's interest and participation.

Griffin Software have selected and adapted for home use, a series of Griffin educational software titles already used in schools.

The first six titles suitable for 4 to 8 year olds, are available now.

If you have a ZX Spectrum 48K or a BBC Model B microcomputer at home, you can so easily keep up the school's good work. And the beauty of it is, the child sees it as a game!

Wordspell helps with spelling; Tablesums unravels the mysteries of tables; Fairshare uses Ollie the Octopus to make sense of division; Numberfun makes addition and subtraction easy; Wordgames helps with spelling and expands vocabulary; and with Getset a child is helped with counting.

All good fun!

Griffin Software titles are at W H Smith, Boots, and other computer shops everywhere.

Only £7.99 each to suit the ZX Spectrum 48K *Only £9.95 each to suit the BBC Model B*

GRIFFIN SOFTWARE

– it's an education.

One advantage the Spectrum has continued to enjoy is a very large and sophisticated software base. Many companies which produced software for the ZX-81 have found it easy to make the transition to Spectrum software. Other companies from the record and publishing world — Thorn-EMI, CBS, Virgin — attracted by the large potential market, have also joined the contest.

The number of new software companies, large and small, to have joined the Sinclair industry in the year is astonishing. They are almost as numerous as the new computer magazines — two new weeklies and innumerable monthlies which appeared during the year. A feature of the last year has been the movement into software. Companies once specialising in hardware, like Quicksilva, dK'Tronics and Kempston, have begun to market programs. Quicksilva ceased production of hardware entirely.

Software has become increasingly big business with the one-man programmer/entrepreneur giving way to the publishing software house which commissions programs from freelance programmers or employs full-time software producers. In recent months computer magazines have been full of advertisements for programmers and/or programs, preferably in machine code.

A number of companies now have large minicomputers which behave like Spectrums — or Orics or BBC machines — but which offer many additional facilities for developing software. Psion, which had a £5 million turnover last year, produced **Flight Simulator** using such a machine; Melbourne House developed **The Hobbit** in that way; and Ultimate devised **PSST.**

More recently, Virgin Games declared its intention to use such a system in preference to commissioning programs. While it is still possible, no doubt, for someone to run a part-time company from a kitchen table, those developments make it more difficult because of the users' demands for well-finished programs and the costs now involved in marketing.

It is an indication of the size and popularity of the computer market that it is beginning to attract crime. Three thousand Spectrums, later recovered, were stolen from Prism Microproducts, the Sinclair retail distributor, in June. It has also become profitable to pirate and mass-produce popular games tapes which are then passed-off as originals from the major software houses.

Another issue which has enraged some software companies is the one of software libraries. In March, Quicksilva applied for an injunction against a company it claimed was effectively lending Quicksilva programs, which might easily be copied.

It is an indication of the seriousness of software companies and the large amounts of money involved in fraud that two organisations have been set up, The Computer Traders' Association in February and the Guild of Software Houses. Both claim to be seeking methods of protecting the customer and policing the industry.

Methods of distribution have changed, too, and have become better-organised. Mail order is still popular but many users prefer to see in advance what they are buying, without having to wait. More and more, software is being distributed through wholesale and retail outlets. Prism Microproducts has reached an agreement with an electrical retail chain to establish software displays in the shops, which will be restocked regularly by Prism. It is hoped to extend the system to other retailers. Prism also hopes to introduce electronic distribution of software via a terminal held by the retailer.

> 'Companies once specialising in hardware have begun to market programs'

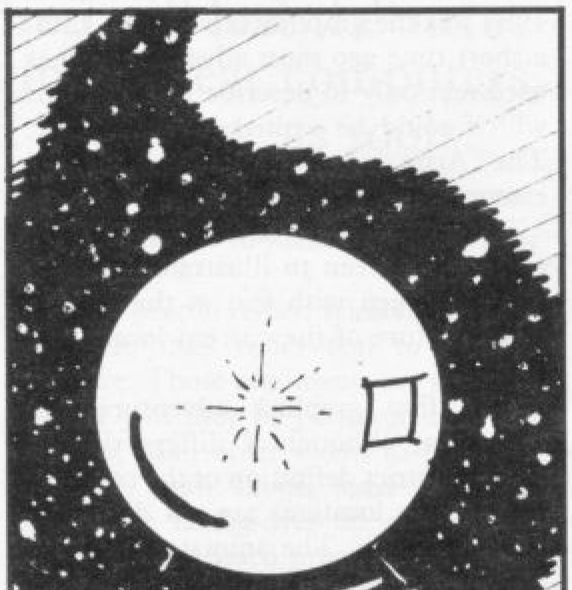
Another new development for the distribution of software appeared this year, after two false starts. Micronet 800 produced an adaptor which allows the Spectrum to be linked via a telephone line to Prestel and to the Micronet programs and information on micros. It also opens the possibility for Sinclair users to contact each other via the system.

Star item this year, though, was the fabled Microdrive and Interface One, which finally surfaced in August after numerous false alarms. It was almost bound to be little disappointing, considering the length of time we had to imagine how wonderful it would be. Having said that, there is little doubt that it will be reasonably high on the shopping lists of most Sinclair users for add-ons; 85K of program loading in a matter of seconds must be worth waiting for and for most of us the wait is likely to be some time.

The only serious reservation is the price of the cartridges. The interface and networking look ideal for use in schools and may go some way to upsetting the near monopoly of BBC machines. That was one area where the Spectrum has not been too successful so far this year.

The other major piece of hardware from Sinclair appeared at the Personal Computer World Show. The Interface Two seems to be designed specially for those who can admit that they did not really buy a computer to work out quadratic equations at all. A joystick controller with plug-in ROM cartridges, it allows instant access to your favourite games programs with a minimum of fuss. The Interface is priced very reasonably at around £20 but the games cartridges are a little more expensive at around £15 each.

Software Scene



An important part of *Sinclair User* is the review of the latest software. Our writers give their opinions on the best available for both the Spectrum and the ZX-81.

John Gilbert's overview

Development rate is 'astonishing'

THIS YEAR the software industry has been very fortunate in having more rises than falls, although in most cases that is due to luck and the market situation rather than good software. The rate of development and achievement by some software houses has been astonishing at times. In the adventure sector **The Hobbit** swept the board for accolades. Arcade software **Time Gate, 3D Combat Zone** and now **Maziacs,** have stood out from a bewildering array of space and maze games.

The Hobbit sent thousands of 48K Spectrum owners mad and floored most critics, who could not get enough of it. Many critics thought, when it was first released, that there would be many imitations. The critics were wrong as there are some companies trying to equal the game but no-one has managed to capture the same atmosphere of madness.

As with adventure games, arcade software has reached a peak. The year started with John Hollis' Time Gate from Quicksilva which showed what could be done with the graphics capability of the Spectrum. Particular attention was paid to the three-dimensional effects used in the program and, as an early effort, Time Gate was stunning.

Another game which shows-off the graphics of the Spectrum is **Halls of the Things** from Crystal Computing. It stretches the Spectrum graphics facilities to the limit and provides an incredible chase round a multi-level maze, populated by strange and deadly creatures.

The game has proved popular with many people although, for some reason, retailers seem reluctant to stock it. From the moment it is LOADed, Halls of the Things displays signs of technical excellence. If you watch closely you will notice something unusual happening when the title screen enters the computer. The display does not appear in blocks on the screen. It appears to move vertically straight down until it hits the last line of main screen and then the game begins. There is no wait between the LOADing of the initial display and the game.

It is surprising that no manufacturers of software have tried to duplicate the efforts of Crystal, although one game, Maziacs from dK'Tronics, is close to it. It also relies on speed and graphics for its appeal. A good deal of thought obviously went into the visual presentation of Maziacs and because of that it is an attractive package.

A new type of game to emerge in 1983 was the graphics adventure. Until a short time ago most adventure games used text only to describe the locations which could be visited by the player. The Artic adventures provide good examples of this type of game.

The new class of adventure uses either full-screen to illustrate locations or split-screen with text at the bottom and a picture of the current location at the top.

The first graphics adventure was **Pimania,** although it differs slightly from the strict definition of the category because the locations are not displayed using graphics. The animated graphics which are used, including the Pi-man singing, give clues to the solution to the puzzles which are posed in the text, or an admonition to the player who is not doing very well. There has been unmistakable reaction from players—you either love it or you hate it.

Apart from Pimania, the first graphics adventure for Sinclair machines was **The Black Crystal** from Carnell Software. It provided the basis for several similar games where the locations were put on to a screen map and the player moved a marker around it.

The next development of this type of format was provided by Doric Computer Services which produced a Spectrum version of **The Oracle's Cave.** The bottom of the display is taken-up with a description of the cave system which you are in and the top contains a silhouette picture of the cave and your player-figure in the cavern.

That figure will walk and climb, depending on what you are asking it to do, and the whole scene is very lifelike. The graphics are smooth, startling and exciting. They show the way for the development of the graphics adventure in the next year.

Unfortunately the business and education market is not much further along the road than it was at the start of 1983. It is too soon to visualise what companies are likely to do with the Microdrive but manufacturers already in the business have made a good, if slow, start.

Most of the products produced in the last year have been for the home finance market and include **Finance Manager** from OCP and **Business Accounts** from Wilsden Computer Services. Little has been produced but the products available are of good quality.

The main contributor to the small business market is Hilderbay, which still produces a **Stock Control Program** and **Payroll** program for the Spectrum and ZX-81. Hilderbay is an exception to the rule, as it also produces software for other more expensive computers, such as the Apple II, and so has the experience to produce the proper type of program.

Another company to edge its way into the small business sector is Kemp. It has provided businessmen with three very usable programs—**Purchase Ledger, Stock Control** and **Stock Ledger.** Unfortunately they are available only for the 48K Spectrum.

No doubt programs of this kind will soon have Microdrive upgrades but until then they are of somewhat limited use for anything but small businesses.

There is little worth noting in the education field this year, except for the MEP programs from Sinclair Research. Sinclair took the unusual step of advertising the programs as ones which will be used in schools and can be bought for home use.

One company moving to the fore this year, not because it retains any really

outstanding programmers but because of the simple brilliance of the teaching technique which is used in the programs, is Widget Software. It has produced several educational programs but one of the most exceptional is **Pathfinder.** The cassette contains four games based on different mazes. The idea is to learn as you play—an important lesson to be learned by all educational manufacturers.

Young people learn more easily by playing games, so Widget has concentrated on combining educational concepts with play. Although the programs are written in Basic, they perform the

to games and utilities as soon as the machine is switched on.

Next year there should also be a radical change in the way software develops and how it is accessed. Software on Microdrive and on ROM cartridge are only two new items. The third is telesoftware which can be accessed from Micronet 800.

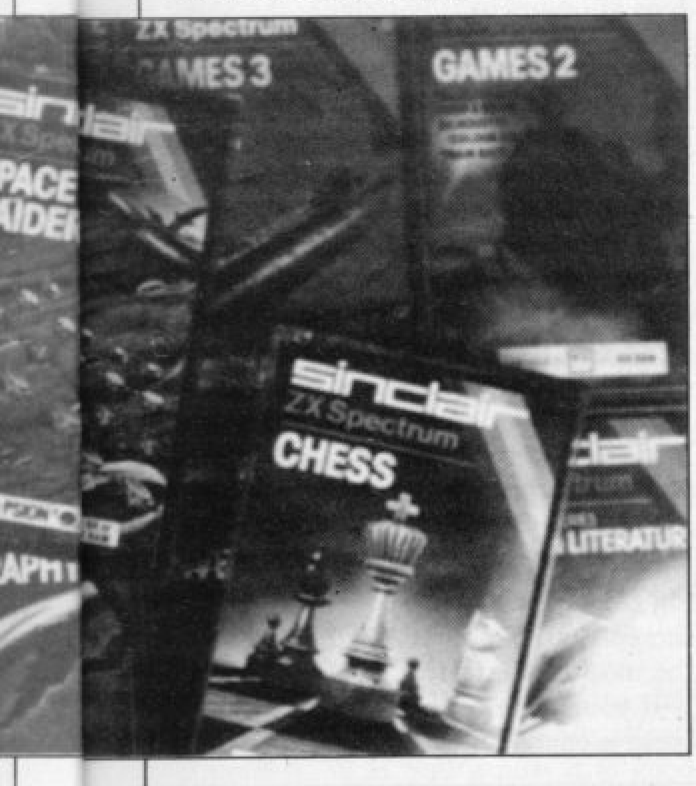
Telesoftware can be accessed from a central database of programs via telephone lines. The programs are transmitted or downloaded from the database and can then be listed on the user's computer. At present the service is available only for the Spectrum but

'Now that 48K Spectrums are beginning to filter into schools, children will lose the attraction for games and begin to treat computers, and the software which they run, as tools'

dual task of teaching and keeping a child interested very well.

Education is another area in which the Microdrive will be useful. It is possible to store information, graphics and questions on Microdrive which can be called up by a teacher or a student. In that way it will be a valuable tool and educationalists would be advised to take advantage of it.

It is too early to see the effects of the Microdrive, Interface One, or even to see what impact Interface Two will have on the Spectrum. Interface Two will contain a ROM cartridge system which will allow users to plug in cartridges similar to those used with Atari machines. That will give instant access



Micronet plans to make it available for the ZX-81.

Some people regard telesoftware as a threat to mail order and to cassette software. Those arguments are not logical nor will they be pertinent in a few months. Mail order seems to be dying and the retail shops, such as W H Smith, are taking-over the sale of cassette and disc-based programs. Because of the easy availability of those programs, cassette software will be with us for a long time. After all, not everyone will not want to buy a Micronet adaptor. It is rather like saying that everybody who wants a video will not want to watch live television.

The Sinclair software scene has undergone a massive change this year but that is nothing compared to what may happen in 1984. Now that 48K Spectrums are beginning to filter into schools, children will lose the attraction for games which they had initially, and begin to treat computers as tools which can be used for a purpose in everyday life. Demands for games software should recede slightly and the education, and eventually business, market should benefit from that.

Children will still have an initial fascination with computers but that should turn to understanding now that many schools are geared to computer science. While 1983 has been a year of consolidation of the market, in 1984 there will be an expansion of the moves which have already begun for practical uses on the Spectrum and still, to some degree, the ZX-81.

Sinclair accounts

John Lambert

THE SPECTRUM and ZX-81 have no reputation for being serious business computers but they are built around the same central processing unit as many grander machines and can perform all the functions required of a business computer system by a small — or not so small — business. The high-quality software necessary for the purpose is becoming available to meet the needs of an ever-expanding market and the recent appearance of the much-vaunted Microdrive for the Spectrum promises to rival some of the bigger disc-based systems. With the addition of a full-size printer and interface, your computer can deal with your accounts, filing system, word processing, business planning and payroll and produce results of a professional standard.

To be fair to Sinclair, its machines were not designed with the business user in mind but rather to be a low-cost introduction to computers in the home. In that it has undoubtedly succeeded but it imposes certain limitations, the most noticeable of which are lack of memory and data storage.

Memory, however, should no longer be a problem, given the large number of add-ons on the market, and the Microdrive provides quick and easy access to large amounts of stored data.

The Sinclair keyboards have attracted criticism but again if the user finds them a problem a number of full-size keyboards are available. The screen display can also pose a problem, 32 characters per line often not being sufficient to produce the required display, but by careful programming the software can either give up to 64 characters or allow the user to output to a full-size, 80-column printer. Ultimately using a Sinclair machine in business will not be the same as using a large, dedicated machine, but different does not necessarily mean worse.

Accounts programs lie at the heart of a business system. They give the business user a powerful tool in the running of a business where at the touch of a button a list of debtors and creditors can be called-up or even a set of accounts produced. In that way it is possible to keep far greater control over affairs. Having decided to operate a computerised accounting system, users must first decide which program to buy.

There, unfortunately, they must make their choice. There are many on the market ranging in price from about £10 upwards and each will perform in slightly different fashion. If possible, the user should take an account to one of the bigger computer shops to try the programs or if that is not possible to contact the software house direct.

Firms such as Transform Ltd or Hestacrest Ltd both supply a range of programs known technically as a suite — for both the ZX-81 and Spectrum — and Gemini, which produces a suite for the Spectrum, will be ready to guide the prospective buyer through the maze of software and hardware available.

Having bought your program you should not have to spend hours learning an alien subject. Your computer program will take you step by step through each stage by a series of prompts and menu selections. There are two important things to remember. First, garbage in, garbage out — i.e., if you enter incorrect data, do not expect the computer to read your mind. That is how you can send bills for £0.00. Second, always make a back-up copy of data entered. Normally that is done by the father-and-son method, where two tapes are used alternately so that, should there be a power cut or some other disaster, you will always have an up-to-date record of your data.

Computer-generated invoices are always impressive and generally less time-consuming to produce. Take for example, the invoicing program from Transform Ltd. Initially you would enter a list of your clients' names and addresses, together with your own, and then whenever you wished to send an invoice/credit note you would have to enter only the first three letters of the name for the computer to find it.

The computer also takes care of all the calculations of VAT, pricing and the various forms of discount, i.e., trade settlement and the like. Having arrived at the total, it will give you the option of printing as many copies of the invoice as you wish, even pausing to allow you to change the paper in the printer when necessary.

At any time you can print a list of outstanding invoices or print your statements. Additionally the program will even print-out your address labels but that facility will be examined later.

Purchase/sales ledger are the main accounting books of any business and as such are catered for by a number of software houses, notably Hestacrest, Transform, Kemp and Gemini. The Cash Book program by Gemini for the 48K Spectrum offers the user some 88 nominal accounts, split evenly between profit and loss and balance sheet which cover all the usual business items, plus memo accounts. It also has the option to make journal entries.

VAT naturally is catered for and the program will provide the figures necessary to complete your returns. The batch system of data entry is used — the purchases or sales are put into batches and the computer checks the batch total against the total individual entries to ensure correct entry. The data files created can be used by the Gemini Final Accounts program to produce a very impressive set of accounts.

Hestacrest also supplies programs for the ZX-81 and Spectrum which supply a set of accounts for either the sole trader or a limited company.

For stock control, there is a wide choice of programs from such firms as Gemini, Kemp — which also produces a Stock Ledger — Transform, Hestacrest and Hilderbay. Using the larger Spectrum, Stock Control from Hilderbay can handle approximately 1,300 lines with details of name, stock level, re-order level, type and supplier code, unit value, total value and a text description. There is also the facility to print-out a list of all items which are below the re-order level or even to print a total stock value.

'Databases is an area where people expect the computer to rule'

Databases is an area where people expect the computer to rule. It is able to store large amounts of data and to manipulate it in any way you wish, whether you want to control stock, store names and addresses or to keep track of stocks and shares.

Of those on the market, probably the best-known are those from Campbell Systems, namely The Fast One for the ZX-81 and Masterfile for the Spectrum 16K or 48K. They must be the fastest and most comprehensive available for the Sinclairs. By using dynamic fields — that is each record and each item in each record can be of different length — maximum use is made of the memory and by writing almost entirely in machine code you have a very powerful program.

Its one disadvantage, if it can be so considered, is its extreme versatility. To use it to the full, a passing acquaintance with computers is a great help, which may deter some would-be purchasers. To the business user a pre-packaged stock control or address program may prove a better choice.

A dedicated mailing list program can prove useful to anyone who wishes to keep a record of clients or perhaps club members. Address Manager by OCP can store up to 400 names and addresses for the purpose and can sort them, if desired, into categories. Mailing list programs are also available from Gemini and Hestacrest.

A bane of an employer's life must surely be the calculation of income tax, especially if more than six weekly-paid workers are employed. Now, rather than doing the job of the Government, you can use your computer. Hilderbay, among others, offers a payroll program which it guarantees to be correct; it also offers a Statutory Sick Pay program.

The purchase of a computer could well be justified even if it was intended only to be used for one of the foregoing programs. There are many other areas where its calculating ability can be put to good use. The best-known utility business program is the spreadsheet type of program. It can best be described as a grid of boxes where each box can be identified by its row and column position, with the contents calculated by referring to the relevant calculation for that particular row and column. By altering the criteria for the calculation it is easy to see the result of a what-if? situation. Of the Spreadsheet type of programs, VisiCalc for the Spectrum or ZX-81 from Sinclair are well-known but others are available from MiCrol, Myrmidon (ZX-81) and Workforce (Spectrum).

Critical Path Analysis from Hilderbay for both the ZX-81 and Spectrum allows the user to enter a network of more than 500 activities, i.e., costs, duration and resources, and the computer will find the critical path.

Optimax from Hilderbay for the ZX-81 is a linear optimisation program where up to 75 variables/constraints are entered and the computer finds the optimum. For example, if a farmer wants to optimise cattle feed, by entering the nutrient value of foodstuffs and the cost, the computer will determine the most cost-effective mix.

If your business is such that all time spent is charged to your clients, as is the case with accountants, advertising agents and the like, Time Ledger should prove useful. It can handle up to 17 employees and 200 clients and is available from Hilderbay for the ZX-81.

Should you need labels printed, Hilderbay can supply a program for the Spectrum 48K to print them. The label can have the date and nine calculable fields of information and a range of sizes. If necessary, the company supplies the labels.

Budget, also from Hilderbay for the ZX-81, will help the business user keep track of expenses through the year. Comp-U-Share for the ZX-81 or Spectrum 48K allows the user to keep track of various shares and investments and can be obtained from Software Workshop.

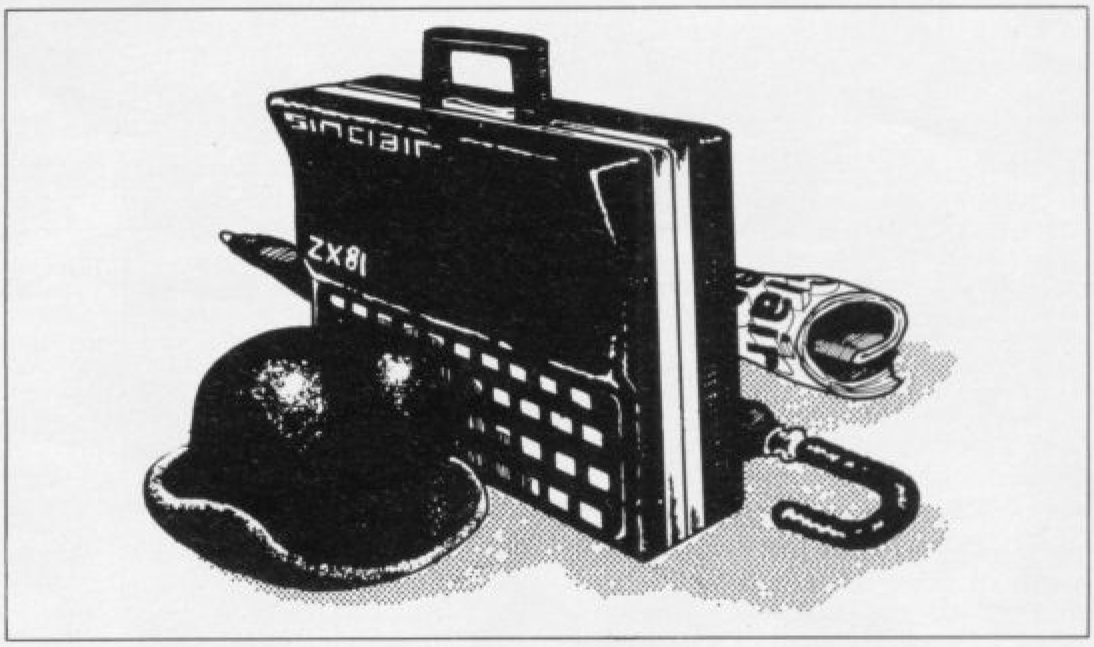
Having produced your figures, Graph Plot by Gemini will do that, by means of pie charts, histograms or graphs. Those graphs aid the user in the comparison of figures or as a selling tool. They are available for both the ZX-81 and Spectrum.

One of the most frequent uses of a computer after databases must surely be word processing. With the addition of a full-size keyboard a Sinclair computer can become a very effective word processor.

Of the programs available, Tasword by Tasman Software for the ZX-81 and Tasword Two for the Spectrum stand out. Tasword Two offers facilities which would not be out of place on a Wang system. With 64 characters to the line, the ability to justify text and generally to move it around at will — either as blocks, lines or single words — and to be able to replace or alter words in the middle of a piece of text make it a very powerful system.

Technical or specialist programs are an area which show how cost-effective a Sinclair machine can be. Similar programs for larger machines cost in the region of £500 for the software alone and yet in most cases the program requires the computer to act only as a calculator on a set number of variables. Any one of those programs, therefore, represents a remarkable saving, in that in a big company each engineer could have his own computer and even in a smaller one the time saved would pay for the machine in its first day of use. One such is Beamscan by G A Rooker for the ZX-81 or Spectrum, of which a full review is given elsewhere in this section.

It is well-known that a moving display in a shop window will attract customers. W H Smith puts it to good use when it started selling the Spectrum in its stores by using the Spectrum to sell itself. The most recent program aimed at the display market is Dlan by Campbell Systems. Using a range of built-in type styles, colour commands and scrolling techniques with a built-in timer, a display can be built and made to alternate displays in a set timed sequence.



What happens to software will depend on the advances in hardware. Most of the Spectrum business programs are being updated to take advantage of the Microdrive, even though its reliability has yet to be proved. The new Interface Two with ROM cartridge slot will also add a new edge.

Perhaps the most important additions to the range of add-ons are the Micronet 800 adaptor and the RS232 interface which can be connected easily to a modem. The possibility of, say, two computers which cost less than £100 communicating with each other anywhere in the world where there is a telephone offers untold possibilities.

'The ZX-81 also has its place in business in the future'

The ZX-81 also has its place in business in the future. Fitted with suitable sensors it must be the cheapest way on the market to control equipment, from factory machines to a programmable office burglar alarm.

Software continues to improve but the problem is one of communication. Ten years ago businessmen were buying £10,000 computers which could do less than a ZX-81 today. Today the same businessmen still spend £10,000 on computers when for £5,000 they could have a complete word processing/accounting package, including the price of the printer. For software writers it is unfortunately true that business software is not profitable on anything but the smallest scale. Until businessmen realise that high prices and high-pressure salesmanship do not necessarily mean high quality, business use on Sinclair machines will remain little more than a sideline.

USEFUL ADDRESSES

Blandfold Ltd, Rydings, Gallows Green, Alton, Stoke on Trent.

Campbell Systems, 15 Rous Road, Buckhurst Hill, Essex IG9 6BL. Tel: 01-504 0589.

Gemini Marketing Ltd, 18a Littleham Road, Exmouth, Devon EX8 2OG. Tel: 0395-265165/265832.

Hestacrest Ltd, PO Box 19, Leighton Buzzard, Beds LU7 ODG. Tel: 052-523 785.

Hilderbay Ltd, 8-10 Parkway, Regents Park, London NW1 7AA. Tel: 01-485 1059.

Kemp Ltd, 43 Muswell Hill, London N10 3PN. Tel: 01-444 5499.

MiCrol, Freepost, 38 Burleigh Street, Cambridge CB1 1BR.

Myrmidon Software, PO Box 2, Tadworth, Surrey KT20 7LU.

Oxford Computer Publishing, PO Box 99, Oxford.

G A Rooker, 20 Vaughan Avenue, London NW4.

Sinclair Research Ltd, Stanhope Road, Camberley, Surrey GU15 3PS.

The Software Workshop, Yew Tree, Selbourne, Hampshire GU34 3JP.

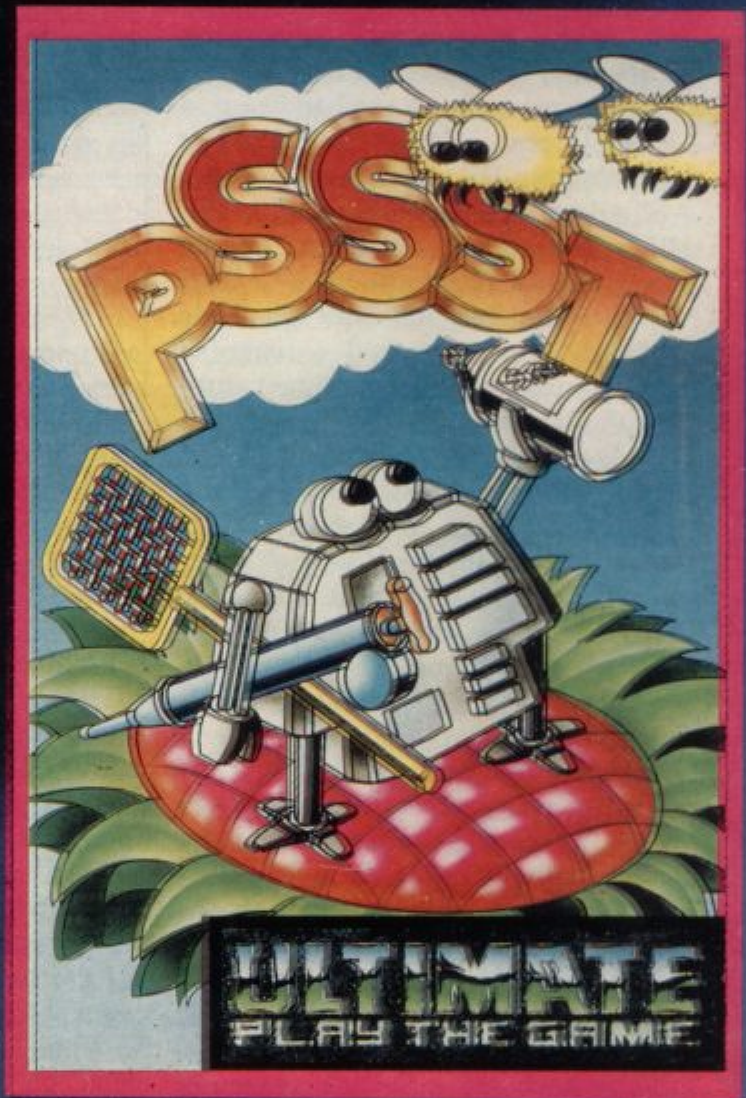
Tasman Software, 17 Hartley Crescent, Leeds LS6 2LL.

Transform Ltd, 41 Keats House, Porchester Mead, Beckenham, Kent. Tel: 01-658 6350.

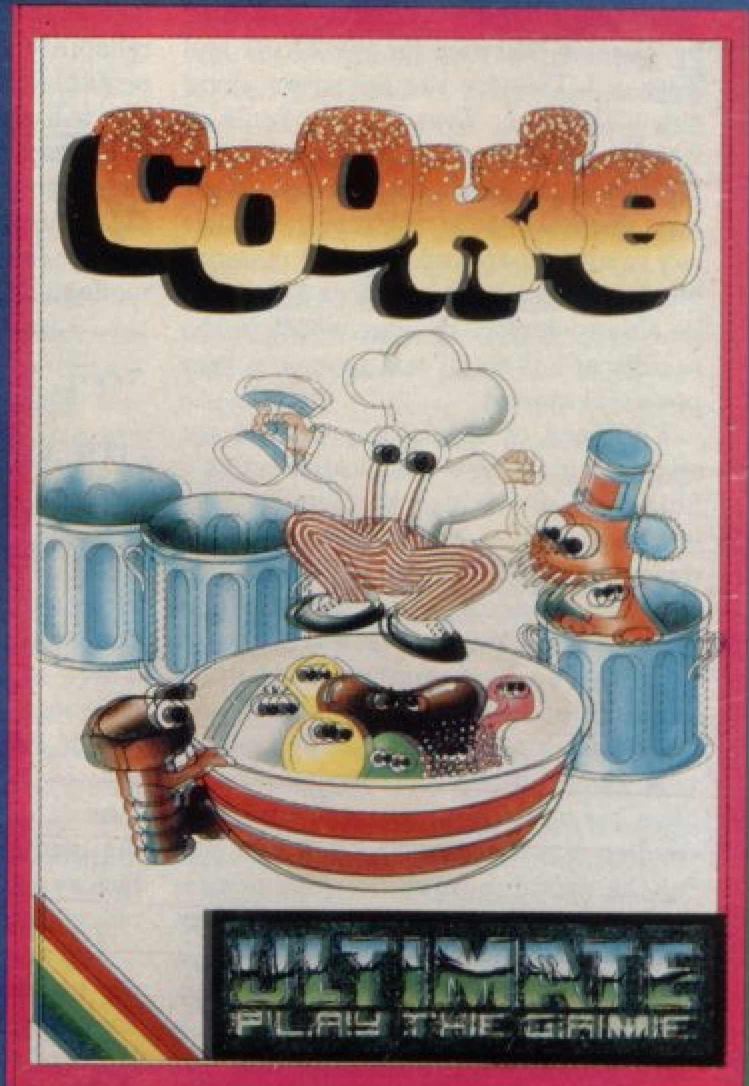
Work Force, 140 Wilsden Avenue, Luton, Beds.

JET PAC –
16/48K ZX Spectrum or 8K Expanded VIC 20

PSSST –
16/48K ZX Spectrum

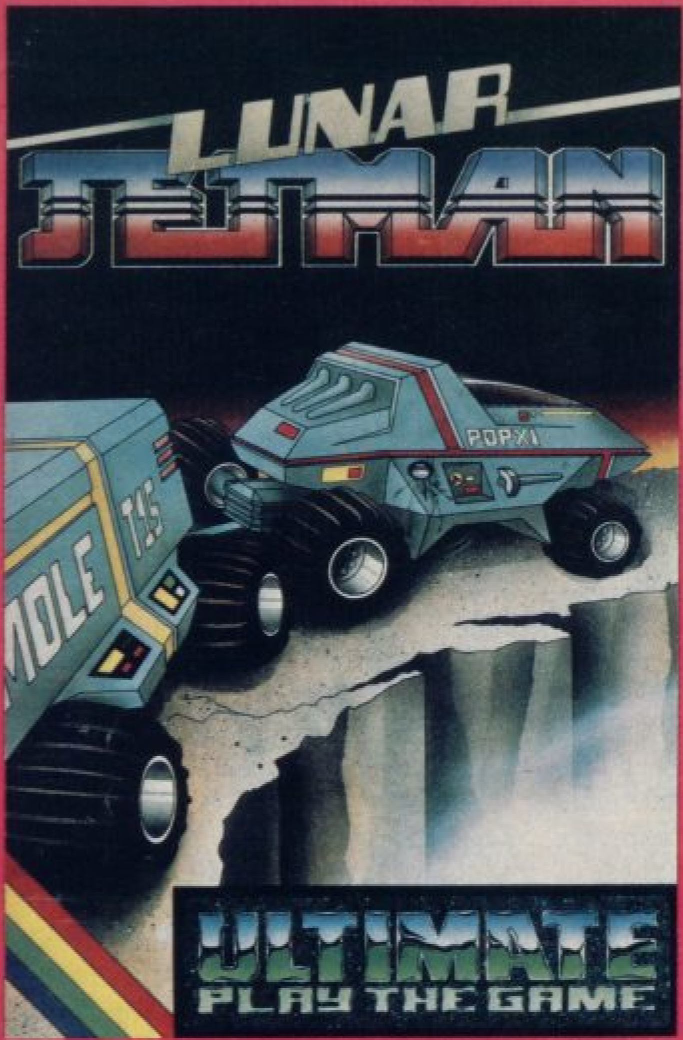


TRANZ AM –
16/48L ZX Spectrum



COOKIE –
16/48K ZX Spectrum

LUNAR JETMAN –
48K ZX Spectrum



LUNAR JETMAN – For the **48K Sinclair ZX Spectrum.**

LUNAR JETMAN – The **ULTIMATE** Intergalactic **G.A.S. (Graphic Arcade Simulation)** Adventure Space Battle.

LUNAR JETMAN – Arcade standard, 100% machine code, incredible sound effects, amazing smooth high resolution graphics, the totally new addictive concept and all those extra features you expect from the **ULTIMATE** games people.

Design – The **ULTIMATE PLAY THE GAME** design team.

ATIC ATAC – For the **48K Sinclair ZX Spectrum**

ATIC ATAC – The super spooky 3D horror **G.A.S. (Graphic Arcade Simulation)** Adventure game.

ATIC ATAC – Arcade standard, 100% machine code, incredible sound effects, amazing smooth high resolution graphics, the totally new addictive concept and all those extra features you expect from the **ULTIMATE** games people.

Design – The **ULTIMATE PLAY THE GAME** design team.

£5.50 each, including VAT, first class postage and packing within UK.

These games should be available from **W. H. SMITHS, BOOTS, JOHN MENZIES, LASKYS, SPECTRUM CENTRES,** other **large department stores** and all **good major software retailers.** Alternatively, send the coupon to **ULTIMATE PLAY THE GAME** for immediate dispatch by return, subject to availability.

Dealer enquiries welcome, Phone (0530) 411485

ULTIMATE PLAY THE GAME is a Trade name of Ashby Computers & Graphics Ltd., The Green, Ashby de la Zouch, Leics. LE6 5JU.

Post this coupon to:
ULTIMATE PLAY THE GAME, The Green, Ashby de la Zouch, Leicestershire LE6 5JU.
Please rush me the following:

- ☐ LUNAR JETMAN
- ☐ TRANZ AM
- ☐ COOKIE
- ☐ JET PAC
- ☐ ATIC ATAC
- ☐ PSSST
- ☐ JET PAC (8K Expanded VIC 20)

I enclose cheque/PO for £ ....

Name..............................

Address..............................

....

Post Code..............................

ATIC ATAC –
48K ZX Spectrum

ZX learning is much cheaper

Eric Deeson considers uses in the school

MANY YEARS AGO, in the 1970s, the use of computers in the home was an oddity. Now something like five percent of British families have computers among their prize possessions. That is higher than anywhere else in the world and Sinclair Research must take a great deal of the credit.

In the 1970s the use of computers in schools was also an oddity. Now almost every British school has one or more among its resources for learning. That is higher than anywhere else in the world, too, and again Sinclair must take a great deal of the credit.

In the former case, Sinclair credit is due to the fact that the machines head the list of sales to homes. That is not the case in schools, where teachers' wishes have so often differed from the ideas of central Government and local advisers. Credit, however, must still go to Sinclair, in that its activities increased computer awareness in the country in general and that increased awareness led to the decisions to spend money widely on educational computing.

It is difficult to employ statistics alone. It is certainly clear that the numbers of schools offering computer activities, and the numbers of candidates in associated public examinations, have risen at the same explosive rate as general sales of micros for less than the £500 mark. Data from other countries is difficult to obtain and far less instructive. Even in the U.S., Japan and Australia, all reasonably well along our road, home computing is still for the tiny minority of middle-class folk and few schools possess, let alone use, micros in the classroom.

A primary school head I know, on a recent visit to Japan, asked to see something of educational computing. His request was received impassively but several days later he was taken to a central Tokyo school. It was a pleasant place, with all the signs of great expenditure, and it had a "computer room". That room contained 12 posh micros, all new, and a pile of boxes hidden in a corner. No children, no teacher in charge. Was it a put-up job?

How do ZX micros compare to others in educational contexts? First, I should note what the others are. There are two computers claimed by their manufacturer as the only machines designed specifically for education. It is certainly the case that very few, if any, homes have them. The manufacturer is Research Machines Ltd of Oxford and the micros are the five-year-old 380-Z and its one-year-old sibling, the 480-Z Link. Both are massive and costly. They will probably always suffer from inadequate software support because the numbers in use are so small compared to those of micros which are also marketed to the general public.

The 380-Z and 480-Z are certainly pleasant machines but their size and cost and lack of support mean they will never take computing to the masses of pupils in a school. All the same, some local education authorities continue to restrict schools to RML machines for various and, in my view, inadequate reasons. I fear that in the areas of the country concerned, pupils and teachers will not progress happily towards computer confidence.

> 'Perhaps no local authority knows the facilities in its schools'

The Acorn BBC micro is growing rapidly in importance as a school machine, just as it is taking a good share of the richer home market. Although there are too many versions for comfort — different models, operating systems and Basics — it has much more potential for education than the RML competitors. It is cheaper and much easier to move around and there is an impressive volume of reasonably good educational software available for it. Even so, I am less than happy at the implications in Acorn advertising that 70 percent of micros in schools are BBCs.

There are other reasonably popular educational machines — popular but in an overall small minority. They include the Apple — especially in Scotland — the Tandy TRS-80 and its clones, and the Pet. The Commodore 64 has made surprisingly little impact on schools as yet and the Vic-20 none, but it is likely that the Electron will make inroads in due course.

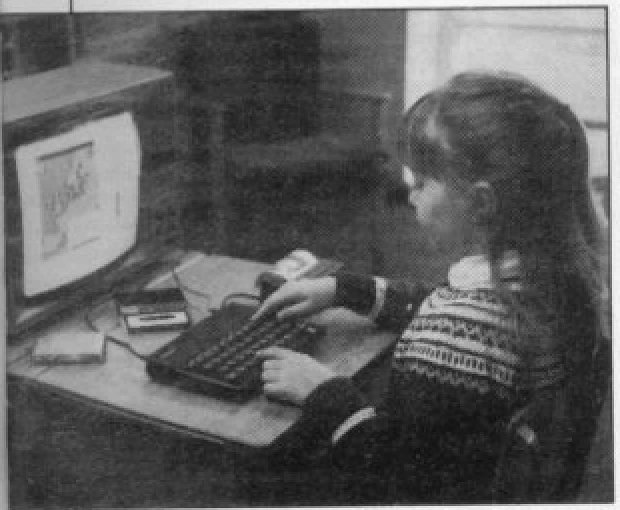
Having made a few vague statements about the popularity of different micros in the educational scene, I wish I could be less vague. There was a flurry of small and large surveys two years ago. That was before the BBC had arrived and before the ZX-81 had a chance to make much impact, so the results are almost meaningless. Since then little hard information can be added to reinforce impressions.

Data from two of my recent small surveys of school machines perhaps can add a little. The figures are 35 percent ZX-81/Spectrum, 29 percent 380-Z/480-Z, 17 percent BBC, 19 percent others — crude figures from crude surveys but perhaps indicating at least something about BBC advertising.

There are various reasons for the lack of adequate data. The first is that perhaps no local authority knows the computing facilities in its schools. That is partly because computing advisers, if they exist, have vast amounts of work to do. Probably even most head teachers, at least at secondary level, could not say what equipment is available in their schools. I have to think hard what we have in my department. If you want to know, it is seven Caltexts, two Cromemcos, two BBCs, two ZX-81s, a 380-Z and a terminal. No-one else knows that, apart from the computing teachers.

The second reason for the lack of data is the multiplicity of sources. Many schools receive at least their first machine through parent-teacher associations, as gifts from local benefactors, or as prizes in competitions. Many others have bought equipment from science, mathematics or technology budgets. In all those cases, I am sure, Sinclairs have scored more heavily than other machines, being so inexpensive and so well-known outside the staff room.

The likelihood of such unconventional acquisitions of micros is particularly high in areas where an authority has told its schools to buy only RML equipment. Teachers see a different need for computers than many advisers and they have often been determined to have a ZX, whatever official policy may be. The main reasons for that determina-



tion are, of course, price and availability of software and other back-up resources.

What then are the needs for computers in schools? I shall not say much about that as the uses are fairly obvious and well-documented in magazines and books. There is my *Spectrum in Education,* for instance. Broadly speaking, computers are used in schools for these main purposes:

To give pupils and staff experience in using them, an aspect of computer awareness or information technology courses.

To extend the possible approaches to teaching specific topics — computer-assisted learning we call it in the trade.

To investigate control/data capture applications.

To provide the means of practical work in formal examination courses.

To assist with administrative tasks.

For the first three purposes the ZX machines win hands down. They are inexpensive. You can equip a room if you wish with 12 Spectrums in a network, plus Microdrives, two ZX printers and a better-quality printer for less than £5,000. That sum would buy you a system based on only two 380-Zs or, if using BBCs, five or six machines. Pass over the permanent network idea and the ZX way gives you sufficient power for every teacher to have good access, even in a large school.

Those teachers would welcome the second huge Sinclair advantage, the mass of cheap, fairly good learning software available for home and schools use. Not many of us have the time to develop many good programs for the purposes of our colleagues. There is also in the Sinclair case a wealth of magazine articles and books, again saving the individual a vast amount of time and frustration.

It is only when one reaches the last two educational applications in my list that the anti-Sinclair case becomes significant. Yet, even there, we find ZXs in positions of strength around the country. Many schools base even advanced level programming courses on ZX-81s and Spectrums and many teachers have at least some administrative packages up and running.

I am preaching to the converted, of course. If you are reading this publication you will already believe that Sinclair micros are the greatest. Be aware, then, that not everyone agrees with you, and in the corridors of political power fierce battles will still have to be fought on behalf of Britain's youth.

What does Britain's youth need? Why did I begin by saying a short prayer of thanksgiving to Sir Clive Sinclair? In my opinion the pupils in our schools need and want to be able to acquire a high level of confidence and familiarity with computers. Computers will offer more and more threats to society — to privacy, employment, freedom, peace — and it is essential that every citizen knows that the machines are no more than friendly slaves.

By the end of the decade it is likely that powerful, fully-portable micros will cost something like £10 and that the present problems of inadequate support will be on the way out. When that day arrives I would hope that as many members of the world community as possible will be able to have such a machine and be able to enjoy, welcome and benefit from its use.

The only way we can approach even feebly that state of readiness is to maximise the use of computers in schools, colleges and homes. They must therefore be as cheap, portable and powerful as possible. They must be supported by plenty of software and print resources. I believe that only the Sinclair range meets that description.

Although many teachers feel that way, educational computing remains much too fossilised in the approach of the early 1970s. Too many schools have massive, costly, unsupported machines with which the word user-friendliness has no meaning. Few schools can offer more than one micro per 100 pupils and more than two teachers able to use the equipment with ease.

The appearance of the BBC machine is helping significantly but ZX-81s and Spectrums remain the only sign of salvation. One large authority in Britain is going the other way. It is introducing an unknown business machine costing £2,000 as the standard. That machine has no colour nor sound and only block graphics. I understand it does not even have a manual.

While that kind of approach continues, educational computing will be for only a small minority. There is no way that all pupils will treat micros as friends — no way will many teachers even find the machines of use in their work. The computing teacher will reign as a high priest over a mystical temple and a small band of unintelligible worshippers. That is an abhorrent picture.

Three-and-a-half years ago I founded the Educational ZX Users' Group within MUSE, the British educational computing association. EZUG grew out of all recognition, far beyond my ability to cope. It showed me that thousands of teachers and parents were convinced of the value of the Sinclair approach to computing in schools. Now EZUG is re-absorbed within MUSE, its function as a banner-waver fulfilled.

All the same, though the banner-waving is over, the battle for recognition won, the war is still a long way from a conclusion. I hope that the conclusion will be computers accessible and valuable to all in our schools but there are many high priests in the corridors of power who continue to fight for computing for the few.

Alas, I fear that even if the war goes our way, there still will be no time to rest. There are few countries in the world where the war has even been declared. Sinclair is leading Britain towards Computer-Assisted Freedom; Britain will then have to do some leading, to give all human beings a brave new world rather than a big brother society.

> **'Educational computing remains fossilised in the approach of the early 1970s. Too many schools have massive unsupported machines with which user-friendliness has no meaning'**

Eric Deeson, who teaches in Birmingham, organised the world-wide Educational ZX Users' Group and is honorary editor of the MUSE magazine, Computers in Schools.

John Gilbert reviews more complex games

Arcades brought into the home

ALMOST every major type of arcade game has been simulated on Sinclair computers. There are also new games which have an arcade format but have evolved on a microcomputer. The Spectrum is an ideal machine on which to play arcade games. The quality of them has improved substantially since the launch of the machine. Some of the first arcade games to be produced for the Spectrum were versions of Space Invaders, the game which started the arcade craze.

Spectral Invaders was the first to be announced by Bug-Byte, which was already renowned for its arcade and adventure games on the ZX-81. So far it is the game which most closely resembles the original arcade version. Although it is slow it is difficult to score points, as you can fire only one laser blast at a time.

Another invaders game, **Space Intruders,** was launched at about the same time as Spectral Invaders. The game is much faster and is recommended for those who like to keep their fingers on the fire button and amass a big score. The only criticism is that the aliens and mothercraft are very small and the mothership is blue on a black background and so is very difficult to hit. Apart from that Space Intruders from Quicksilva is good value at £5.95.

Namtir Raiders, for the ZX-81, is a space invaders game with a difference from Artic Computing. The player still has to face the hordes of aliens which come down the screen but the laser base can be moved up and down as well as left and right.

The player has five laser bases during the game and they can take only five hits from alien bombs. There are three levels of difficulty—easy to impossible. The game, costing £3.95, is addictive and the graphics are neither awkward in design nor jerky in movement.

The game which is gaining popularity with 48K Spectrum owners is **Time Gate,** from Quicksilva. The authors claim that it is the ultimate in 3D arcade space action. The game is loaded in two parts, the first being a lengthy instruction manual. After the manual has finished you can load the game. The object is to destroy a race of aliens called the Squarm who are trying to colonise Earth. The enemy fighters are also seen in 3D and so are the planets on which the player can land to refuel and repair the ship. It is by far the best arcade action game so far for the Spectrum and costs only £6.95.

Asteroids is also becoming popular on Sinclair machines. One of the first companies to launch a version for the ZX-81 was Quicksilva. It was a good version on such a small machine and is still proving popular.

Quicksilva has also introduced a Spectrum version called **Meteor Storm.** It has the added attraction of speech before play. It is difficult to hear the words but we are informed by the authors that it says "Meteor Alert . . . Meteor Alert . . .". Meteor Storm is a novel version of Asteroids and costs £5.95.

Sinclair Research has a good game of asteroids, called **Planetoids,** in its new Spectrum Software library. The asteroids are in 3D and much careful design work on both the asteroids and the player's ship seems to have gone into the game. Planetoids costs £5.95.

Artic Computing seems to be the only company to have produced a version of Galaxians for the ZX-81. **ZX-Galaxians** looks like Space Invaders but the invaders are 'V'-shaped and are supposed to be inter-galactic birds. The birds swoop from formation and bomb the player's laser base. ZX-Galaxian is slow in action but can still be a very addictive game. It costs £4.95.

Defender is still a much-sought-after game in the arcades and Artic Computing took advantage of that early by producing a version for the ZX-81. The graphics are not particularly interesting and the spaceship which the player flies across the landscape is made up of a series of blocks which look only slightly like a ship.

Despite those criticisms, the original idea behind the game is still there and the Artic version can be exciting, as you see the enemy ships rushing at you from the other side of the screen.

With the arrival of the Spectrum, many manufacturers found an interest in the arcade game **Scramble.** The best and fastest version so far is from Mikro-Gen. In the game you have to go through caves which become smaller and smaller as it progresses.

The game becomes progressively more difficult and can be run in slow, normal and fast modes. It costs £3.95.

Silversoft has a Scramble-type game called **Ground Attack.** It works on the same principles as the Mikro-Gen game but is much slower. There is a good deal of blank screen between game rounds and the average waiting time between rounds is 15 seconds. Ground Attack costs £5.95.

The range of arcade-type games on Sinclair machines is always increasing. Manufacturers seem to feel safe in producing standard arcade games such as Space Invaders and Scramble. Those games, especially for the Spectrum, are becoming more imaginative and the graphics and sound more impressive.

Manufacturers have to be careful about copying ideas from other games but with the imagination of some of the firms in the Sinclair market, children and many adults will be kept happy with arcade-type games on the ZX-81 and Spectrum for a long time.

Space Invaders games have now been overtaken by **Pacman** in popularity.

Bug-Byte, 98–100 The Albany, Old Hall Street, Liverpool L3 9EP.

Quicksilva, 92 Northam Road, Southampton SO2 0PB.

Artic Compting, 396 James Reckitt Avenue, Hull, North Humberside.

Sinclair Research, Stanhope Road, Camberley, Surrey GU15 3PS.

Mikro-Gen, 24 Agar Crescent, Bracknell, Berkshire RG12 2BK.

Silversoft, 20 Orange Street, London WC2H 7ED.

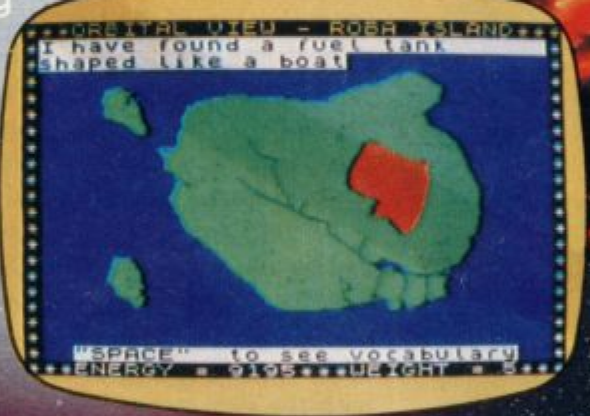
TERMINAL SOFTWARE

VAMPIRE VILLAGE

In a ruined castle near the village of Vladsdorf lurks a vampire who roams by night. As mayor you must organise volunteers to save the villagers. This is a REAL-TIME graphic adventure and every game is different from the one before.

SPACE ISLAND

Reports of a time distortion around a distant planet lead you into the strange world of an ancient civilisation that although long dead, now threatens the present. This is a REAL-TIME graphic adventure and every game is different from the one before.



CITY

A unique REAL-TIME better-than-a-board game for 1-4 players that combines simplicity of use with complexity of form as the city you build takes on a life of its own. A functioning economic model for you to play while the Spectrum keeps track of the time, the money, the properties and the people.

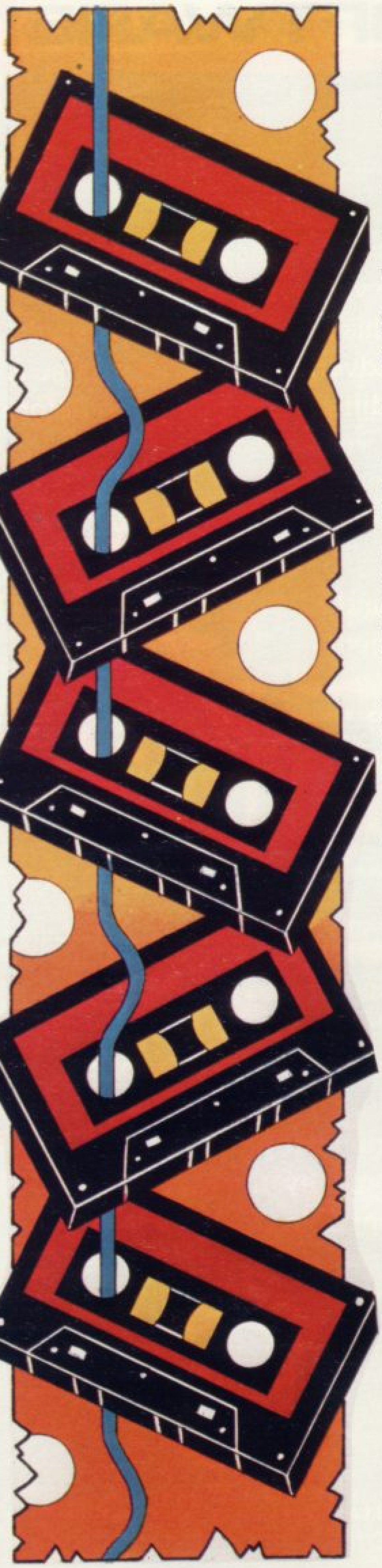
For the

sinclair Spectrum 48k

All Super Value at Rec. Price £6.95

TERMINAL

TERMINAL SOFTWARE
28 CHURCH LANE, PRESTWICH, MANCHESTER M25 5AJ.
TEL: 061-773 9313



Applications for the Sinclair at home and the office

Why not stop playing games and do something interesting with your computer? John Gilbert assesses the software

BOTH SINCLAIR machines can be used for storing data of any kind, such as names and addresses, telephone numbers and even an ever-changing record of appointments. The ZX-81 needs the 16K RAM pack for any kind of data storage and both information and program have to be SAVEd together. That operation can take up to six minutes and is not very reliable.

The **Business and Household** cassette was one of the first packages available from Sinclair for the ZX-81. It contains three programs. One will keep a record of names and addresses, the next will keep a diary of events and the final program will handle all your financial transactions.

The first two programs worked well but the **Bank Account** program on side two took six minutes to load and SAVEing the program back on to tape with the data proved very difficult.

The Business and Household cassette may not be very reliable but it is good value at £3.95.

One of the best data management systems available for the ZX-81 is **The Fast One,** from Campbell Systems. It allows the user to set up files of information in any way which suits him. The program will sort and search for specific bits of data and if numbers are being used it is possible to total them. The program is a step forward for the ZX-81 and is very flexible. It will do any kind of filing job, given the limitations of the machine. The Fast One costs £15 and has a comprehensive manual.

Spreadsheet programs are an easy way to store numerical data in a format in which it can be used for calculations. The spreadsheet is a matrix, or table, on the screen and any box, or cell, in the table can be addressed by using the letters and numbers which run horizontally and vertically at the sides of the sheet. This type of program can be used to plan the family budget and calculate automatically running totals of family expenditure. That is only one of the many applications for which it can be used in the home.

MiCROL produces a spreadsheet program called **Matrix Planner.** It is easy to use and has a spreadsheet of eight rows by 30 columns. That configuration can be changed by the user through the program variables. Approximately 300 cells can be created in the matrix before all the 16K of memory is used.

Sinclair Research markets two programs similar to the Matrix Planner. **Vu-Calc** is a program which uses the spreadsheet. It has limitless possibilities and can be used for financial modelling, keeping track of bank accounts and even setting-up scientific experiments which rely on number-crunching for their outcome.

The second is **Vu-file.** It is like Vu-Calc but the user can only store information and not perform calculations on data. Both programs are available for the ZX-81 and Spectrum. ZX-81 versions cost £7.95 and Spectrum versions £8.95.

> 'There are programs for data storage on the Spectrum but most of them can be used only on the 48K version'

The arrival of the Spectrum set software houses the task of writing programs which can use data files separate from the programs. It has opened the way to storing large amounts of data on cassette and, with the arrival of the

Microdrive on floppy tape. There are several good programs for data storage on the Spectrum but most of them can be used only on the 48K version.

The Database from MiCROL is one such program. The files can be split into documents. Those documents are useful in splitting-up topics within the machine. You can give each document a heading, such as tax, income or budget, and you can have several of them in memory at one time.

Documents are split further into records, with one record corresponding to each datum. With that system it is possible to do your tax and budgets at the same time, without having to load the computer twice with information. The program can store up to 999 record lines in memory. The Database costs £9.95 and is complete with handbook.

The **Masterfile** program from Campbell Systems is the most comprehensive of the databases available. It is

> 'Most data processing programs can already deal with more information than the ordinary user needs'

the successor to The Fast One for the ZX-81 and provides fast access to large amounts of information. The user can also model the program to meet specific requirements. Data can be sorted and searched and reports can be compiled using the system. Masterfile costs £15 for the 48K version and £12 for 16K.

The spreadsheets which proved so popular with the ZX-81 are starting to creep on to the Spectrum market. The best, so far, are from MiCROL and Microsphere. Both are remarkably similar. The MiCROL version costs £9.95 and provides the basic calculating power of most spreadsheet programs. It is easy to use and can help the business or home user with complicated calculations.

Omnicalc is the spreadsheet from Microsphere. It is ideal for someone who has just found the spreadsheet concept but it is also a very powerful tool for anyone who has used one previously. The program seems to work faster than the MiCROL spreadsheet and information can be accessed almost immediately.

The screen format is easily understandable and very clear for the first-time user. The program contains a help option which lists the commands available through the spreadsheet. Omnicalc costs £9.95. It is complete with a user manual.

All-Sort is an interesting utility program for the 48K Spectrum. It enables a user to sort data which has been set up within a home-built program. The data is stored initially in an array and All-Sort can sort up to four of them at once. It can be obtained from Alan Firminger. The program is useful and very fast but at £18 exclusive of VAT it is expensive.

Listfile is a program which does exactly what its name suggests. The program allows a user to store lists of data, such as names and addresses, and to access that information very quickly. Data is entered in blocks which can be up to eight lines of 26 characters long. An extra line, called the info line, can be used to index information but that is not printed-out when the printer is used to list the information.

Listfile is available for the 16K and 48K Spectrum and can be obtained from G and J Bobker. It costs £10 and has full documentation.

Now that the Spectrum has arrived, software manufacturers are beginning to think about software uses other than games on Sinclair machines. The data processing programs could handle many tasks which are centred on the home. Databases, such as the one from MiCROL, are useful for storing textual information, such as a list of favourite records or even knitting patterns.

The Microdrive can expand the data processing capabilities of the Spectrum. Information can be accessed more quickly and as a result bigger programs can be stored in memory and data can be fed in bit by bit.

The capabilities of the Spectrum can be extended in this way but soon we will have to decide whether it is necessary. Most data processing programs can already deal with more information than the ordinary user needs. It may be proved that that type of application for the Microdrive is a waste of time.

Sinclair Research, Camberley, Surrey GU15 3BR.
MiCROL, 31 Burleigh Street, Cambridge CB1 1BR.
Campbell Systems, 15 Rous Road, Buckhurst Hill, Essex IG9 6BL.
Microsphere Computer Services Ltd, 72 Rosebery Road, London N10 2LA.
Alan Firminger, 171 Herne Hill, London SE24 9LR.
G and J Bobker, 29 Chadderton Drive, Unsworth, Bury, Lancs.

New and varied life-forms now inhabit the Pacman maze. John Gilbert investigates.

Leading the Pac

PACMAN is part of what the video historians like to call the arcade maze craze. The game is set in a maze where a hungry little round creature eats dots and power pills scattered through the corridors. The monsters are ghosts which hunt the little man and will eat him if he is not agile enough to escape.

If, however, the Pacman eats a power pill it can chase and eat the ghosts. The original game was introduced to the home computer market by Atari, primarily on its VCS video system and then on the 400 and 800 computers.

The first versions for other computers, variously called Puckman, Gobbleman and Scoffer, arrived shortly afterwards and since then new versions have been released almost every month.

Not surprisingly, the ZX-81 did not escape the craze. The authors of **Zuckman,** from DJL Software, claimed that it as the first version of Pacman for the ZX-81. The game runs in 16K and is written in machine code, which gives the scrolling routines and Pacman a smooth movement. That is something unusual on the ZX-81, as most games flicker slightly.

The game adheres as closely as possible to the original and the limited graphics of the ZX-81 do not matter much. One snag with playing the game on the ZX-81 is the membrane keyboard. Moving a Pacman round the screen using it can be difficult but not impossible.

At the beginning of Zuckman the ghosts speed on their victim and if you panic trying to find the keys and do not press them properly you are liable to lose a Pacman or crash the machine. Once you have a fair idea of the game, however, it is surprising how quickly you can move the Pacman around.

Zuckman is available for £5.95 on the 16K ZX-81 and Spectrum. **Super Glooper** is a version of Pacman, also for the 16K ZX-81, which is retailed by Sinclair Research. Glooper's task is slightly different from normal. Instead of gorging himself with power pills he has to paint the maze. Obviously the ghosts will not tolerate Glooper's antics — perhaps they do not like the smell of paint. The ghosts will try to kill Glooper as usual but if he can get round the maze and paint all the walls you have won.

The game is very amusing and is well worth £4.95. The program will load in less than two minutes, so you will not have to wait six minutes to play the game.

The launch of the Spectrum provided software houses with an excuse to try to produce the perfect Pacman which simulated the Atari version as closely as possible but Atari guards its rights over products jealously and investigated the products of several firms in the ZX industry.

The Abbex **Spookyman** game is the most famous of the Spectrum versions. It also looks most like the original. Most games, until then, included only dots and power pills in the mazes. Abbex included dots, power blobs, diamonds, hearts, clubs, mean-looking ghosts and a cute little Pacman which looks like a diamond turned on its side.

The keyboard is divided into three sections with left control on the left, right on the right, and up and down in the centre. The controls are very difficult to master but, once you have done so, usually with the aid of both hands, you will be surprised at the agility you can attain.

At times the game is almost impossibly fast but Abbex estimates that the highest score possible after every screen has been cleared is 22,400. If you manage to reach 10,000 points you will receive a bonus life. We managed to go that far only once.

Spookyman can be played by one or two players. If two play they take turns to control the Pacman. Their individual scores and the highest score are included at the top of the screen.

There are two unusual features in Spookyman. The first is that you can re-set the high score at any time between games; the second is that you can use a joystick. The game is compatible with the Kempston joystick and once you have seen it in action it is almost a necessity. Spookyman is available from Abbex and costs £4.95.

Gulpman is another variation on the Pacman theme. The round Pacman is replaced by a little man running round the maze and the ghosts become frowning faces which smile only when they have caught Gulpman.

The game is very complex and you can switch to any of 15 mazes in which to play. It is also possible to change the tempo. At tempo one the speed is bearable but at tempo 10 life is not worth living.

The little man can fight back slightly more easily than in other games as he approaches with a fully-loaded laser gun. If the ghosts get too close you can blast them away but only until your energy runs out. It seems as if the space age is over-running everything. Gulpman is available from Campbell Systems and costs £5.95.

Hungry Horace, from Sinclair Research, has developed a reputation as a fun game; in fact Horace is almost a legend. The game is a great improvement on the original and remained at the top of the software top ten for some time.

Horace is a large purple blob which has sprouted arms and legs. He wanders round the maze which looks like a park, eating everything in his path and trying to avoid the guards who act like ghosts and try to capture him. He can scare away the guards by using the alarm bell situated somewhere in the maze.

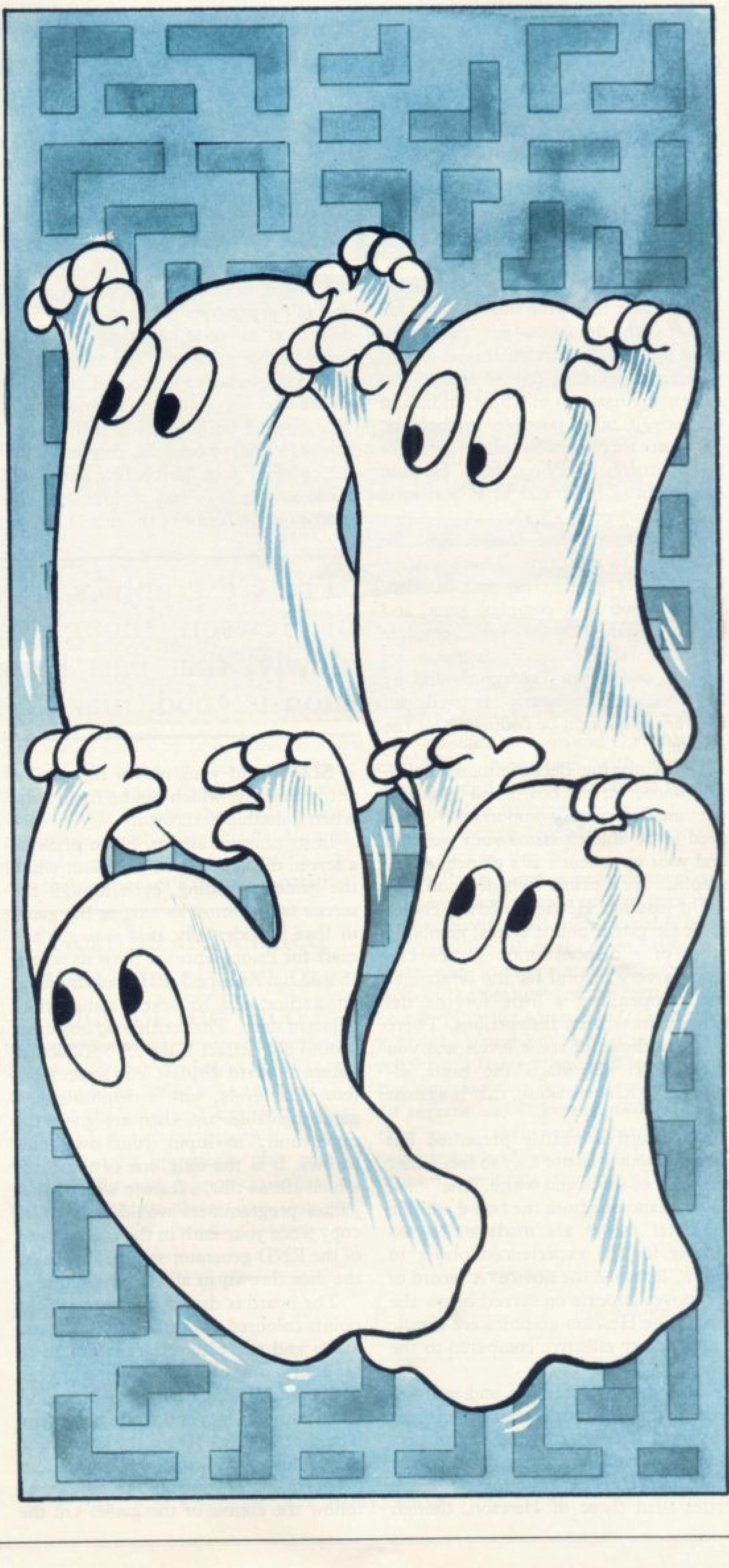
If Horace reaches the exit of one of the mazes he can enter the next maze and continue to the next exit. The mazes become more difficult as Horace proceeds further in his adventure and we managed to reach only the third maze. With a large amount of skill, however, it should be possible to go further.

Hungry Horace, for the 16K Spectrum, is available from Sinclair Research and costs £5.95.

Although the arcade industry in the States, and now in Britain, is declining it is good to see that games concepts like Pacman are being transferred to micros.

Some of the Sinclair versions of Pacman seem as good as, if not better than, the original Atari version. Games such as Hungry Horace are setting-up an interesting mutation in the Pacman concept. They also seem more interesting than the original version because they have added to the idea of Pacman. So far as the consumer is concerned it is to be hoped that concepts such as Pacman will not be destroyed within the industry.

DJL Software, 9 Tweed Close, Swindon, Wilts, SN2 3PU.
Sinclair Research, Stanhope Road, Camberley, Surrey, GU15 3PS.
Abbex Electronics Ltd, 20 Ashley Court, Great North Way, London, NW5.
Campbell Systems, (Dept. SU), 15 Rous Road, Buckhurst Hill, Essex, 1G9 6BL.



Backgammon has long been a popular game which requires a mixture of luck and skill. John Lambert reports on three versions

Ancient game has varied success on the Spectrum

BACKGAMMON is an ancient game involving much more skill than draughts, yet dependent more on luck than chess. It is as old, or possibly older than any of them. The ancient civilisations of China, India and Greece all offer possible birthplaces. There are three versions of the game for the Spectrum by Psion, 16K; Hewson Consultants, 16K; and C P Software, 48K; all priced at £5.95.

Each cassette has instructions for those new to the game. Those written for C P are good, clear and detailed. Backgammon is a complex game and the notes make play easy for a complete beginner. The Psion instructions are equally useful but those provided by Hewson are not nearly as well put together and might be confusing for the novice.

When playing Backgammon, the visual impact of the board and layout of the 'men' is vitally important — you need to be able to assess your position and your opponent's at a glance, so the graphics are a prime consideration.

On loading, Hewson offers a choice of single game, points series, gambling series or a demonstration game. The latter is very helpful for the newcomer and compensates a little for the deficiency in written instructions. There is also a choice of static levels and you can choose who starts the game, although, strictly speaking, that is against the rules.

The board is swiftly presented but unfortunately it is not easy to see, either in colour or black and white. The 'men' do not stand out from the board and the computer moves are made much too quickly for the experienced player to follow, let alone the novice. A record of the moves appears on screen below the table. The Hewson graphics are simple and not very effective compared to the others.

Load the C P version and you are presented with brief instructions for play, which neither of the others provides on-screen, but there is no choice of skill level. The graphics are much better than those of Hewson, though the board is drawn very slowly, that part of the program being in Basic. The definition is good, making the men easily visible in colour, and only a little less so in black and white, but since the points are not coloured alternately as they should be it is often difficult to calculate your moves. In this program the chosen pieces flash before a move is made so that it is easy to follow and a record is kept below of the moves, but it is SLOW and your moves have to be entered singly, which can be frustrating when a double is thrown.

'The CP graphics are much better than those of Hewson, though the board is drawn very slowly, that part being in Basic. The definition is good, making the men easily visible'

In its normal fashion, Psion presents a screen display for you to look at while the game is loading, even though the screen takes almost as long as the game to load. Incidentally that was a black mark for Psion; whoever drew its screen should have realised that opposite faces on a dice add to seven rather than adjacent ones. That criticism, however, should not detract from the spectacular nature of board display. You select from four skill levels, with a demonstration game available, and then are given the opportunity to input your own dice throws. It is the only one of the three which allows this, a feature which other games programmers would be wise to copy since your faith in the randomness of the RND generator will be shaken by the dice thrown in all the programs.

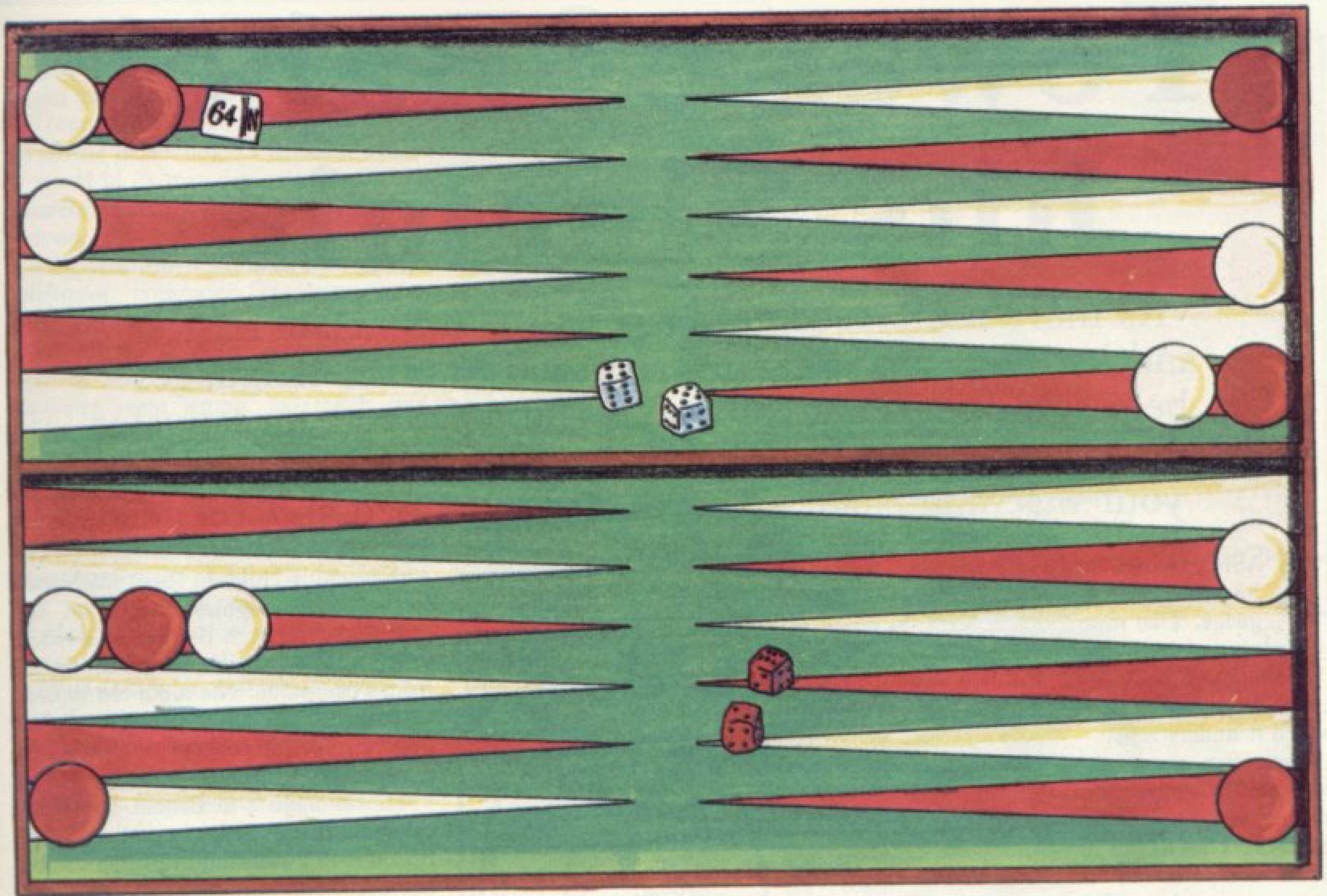
The board is drawn quickly with the points coloured alternately in black and white and the pieces, large enough to see easily, four character squares, in red and cyan. The definition is not lost when using a black and white television. The dice 'roll' in 3D up the screen and the pieces move across the board from point to point, making it simple to follow the course of the game. On the points with more than five men, the pieces appear to stand on their edges to make space, whereas the other two games resort to using numbers in that situation. When blots are hit, they travel gracefully to the bar, where a maximum of two men of any one player are shown at a time.

In the middle of the bar is the doubling cube, which moves from player to player in use. Hewson is the only other game to offer doubles but only in its gambling series.

Moves can be changed after they have been made by use of the DELETE key, the men re-tracing their steps across the screen. EDIT elicits suggested moves to help the novice player throughout the game. The graphic display is well-designed and effective.

All the games use the conventional rules of play, as published by Hoyle, but for scoring C P has no doubling option, an integral part of the modern game. Hewson uses its own method of calculating points instead of the accepted one. Only Psion scores correctly.

Hewson plays erratically, sometimes being very conservative and at other times taking wild risks. Moreover, by moving about frequently within its own inner table it is unable to take full advantage of the dice. When playing a back game it does not persevere long enough and on one occasion when one of its men was on the bar and most of its opponent pieces had been borne off leaving a blot on the three, Hewson threw five/three and came in on the five, thus losing a gammon. Apart from that instance it usually 'hits' at almost every opportunity and so it can be trapped by a skilful opponent. On the whole the level of play, even at its



highest, is moderate and does not provide a stimulating challenge to an experienced player.

It is interesting to note that M Male, the author, also wrote the excellent air traffic control simulation, Heathrow, for Hewson.

C P is another fanatical taker, but rarely takes the conventional precaution of building houses in its inner table. On the rest of the board its moves are generally conservative but its defeats of Hewson, as indeed when Hewson beat it, depended on some very lucky dice throws towards the end of the game. The two programs are well matched, their skill levels being about the same and their strategies very similar.

Psion plays a much more sensible game and provides more of a challenge. It makes better and more frequent use of the standard openings and its strategy throughout the game is more consistent. It protects its inner table and leaves few unnecessary blots but once again when playing a back game it tends to lack conviction and runs for home too soon.

To test the abilities of the games a 'tournament' was arranged. Each program played five games against each of the others. The results, shown in the table, were surprisingly even.

It was expected, on the basis of playing the game individually, that the result would be Psion first, Hewson and then C P. None of those programs, however, can assess the play of its opponent, which is why they fail to take advantage of each other's faults. Human players would assess and eventually predict their opponent's moves, frustrating a back game by refusing to hit blots, or avoiding blots left as obvious traps.

'Results depended often merely on the luck of the dice'

Since the programs cannot do that, the Psion game, for example, fails to realise that its opponents play consistently badly and cannot capitalise on that as a human player does. For the same reasons, Hewson and C P opposed each other three times with identical strategies and neither was able to realise that and alter its play accordingly. The results therefore depended often merely on the luck of the dice.

The Psion game is programmed entirely in machine code and so uses the comparatively small space available on a 16K machine efficently, even using the spare space in the printer buffer for the table of the positions of the men on the boards. Now the Microdrive is available it may be a problem to fit it in. On the other hand Hewson and C P are written predominantly in Basic, Hewson about 70 percent and C P nearly 90 percent; that makes them somewhat cumbersome and would, particularly in the case of C P, welcome the use of a good compiler.

RESULTS

	1	2	3	4	5	
Psion v.		W	B	W		Psion wins 5/3
Hewson	G				W	
Hewson v.		W				CP wins 8/1
CP	G		G	W	W	
CP v.	G				G	Draw
Psion		W	W	G		

W = Win G = Gammon B = Backgammon

Tower of Babel hits Sinclair

As more users become accustomed to Sinclair Basic and its limitations they have turned to other languages to help solve particular problems. John Gilbert investigates the growing demand for new ways of conversing with your machine, such as Forth, Pascal, Logo and Lisp.

BASIC, as used on the ZX-81 and Spectrum, is a high-level language. That means that it is easily understood by human operators but that the computer has to translate any entry made by a user into the code which it understands.

Basic was designed to help people become accustomed to a computer and was not written to perform any particular task. That is the problem with it when compared to other languages. Languages such as Forth and Fortran were written to perform specific tasks.

Forth has already been implemented on the ZX-81 and some versions are available for the 48K Spectrum. The language was designed originally for engineers who needed to process formulae. It runs at many times the speed of Basic and, with graphics added to the list of commands available, you can play a good game of space invaders by using it.

The most interesting aspect of Forth is that you can define your own commands using words which already exist as part of the standard system. A dictionary of those words usually is accessible on the system using a list command. For instance, if you want to find the square of a number you would use the standard word DUP. If you want to find the square of a number squared again you could define your own word, for instance DUPD, by defining it as DUP*DUP. All that seems very complicated if you are used to Basic.

Some languages require even more discipline on the part of the programmer.

Pascal, for instance, is a language which requires the programmer to define all variables and functions used throughout the program in the first section of the code. A program written in Pascal must then be structured as a series of routines, each routine being relatively independent of the others.

The language is slightly more difficult to use than Basic as it forces the programmer to think about how the code is to be put into the computer, which operations are to be performed first, and where routines are to be placed in a program, instead of deciding what code you need for a specific job.

"Hmm . . . it's not so much BASIC as EARTHY."

Children and schoolteachers are becoming interested in a language called Logo, which has been imported from the U.S. Its inventor is Seymour Papert and embodied in it are several controversial concepts which many educationalists are beginning to question.

The emphasis of the language lays with teaching children about mathematics through graphics shapes on the screen and through the use of a robot, called a turtle, which can be controlled from a computer. The robot can be used to draw shapes on the floor and puts into practice the concept of learning by experience.

Educationalists in Britain are sceptical about using turtles in that way. The reason is that learning in British schools has been, and still is, by the repetitive rote learning method.

Some schools are experimenting with turtles but the lessons taught using them have been structured so that children are still learning by rote and not by experiment or experience.

At present, Sinclair Research is designing a version of Logo to work on the Spectrum. Edinburgh University is also working hard to produce a floor turtle for the machine. The language will be supplied free to schools which choose the Spectrum under the Government Microcomputers in Schools scheme.

The artificial intelligence language Lisp has already been released for the Spectrum by Serious Software. Lisp is used for list processing and programs can be written which will act like intelligent databases.

Serious Software has provided a database written using the Lisp package. It includes the names of a family group with relations which they form to each other. Using the program you can discover which members of the family have children, which have brothers, and whether those brothers are married. The database seems intelligent because you can ask it questions which are limited only by your imagination.

Normal databases would allow information to be accessed only using such techniques as entering a keyword to find data. You can also use Lisp to write programs which will seem to give meaningful answers to entries you type into the computer. The computer will select an answer depending on the data which is stored in its memory and how it has answered questions similar to the current one.

Programs which simulate this type of intelligent computer response have been written in Basic. One such program, Eliza, acts like a human psychiatrist and many of the replies it generates seem almost too much like human response.

The Eliza program in Basic is amusing but with large amounts of data in memory the program will respond very slowly, sometimes taking two minutes to analyse entries. Lisp was devised for that kind of work and so it has the advantage of speed over Basic in this area.

Most computer languages were designed with a purpose. Unfortunately, languages like Basic have been corrupted and are now used for different purposes from those which their designers had intended.

Basic was not intended as a commercially-used programming language for the design of space invader games. Unfortunately, programmers can use only the langauge available to them and on the Spectrum that is either Basic or machine code.

Sinclair is now making it possible to use other languages but Basic is still the most popular, not because people like it better than other languages but because very few people have had the opportunity to use any of the others.

Only one small microcomputer has been produced which will run a language other than Basic, the Jupiter Ace. Sinclair is definitely an innovator in the market, so perhaps a micro with a new dedicated business language may be the next step.

FORTH
Artic Computing, 396 James Reckitt Avenue, Hull, North Humberside. (48K Spectrum/ZX-81).
Abersoft, 7 Maes Afallen, Bow Street, Dyfed SY24 5BA. (Spectrum 48K).
PASCAL
Interface, Dept. SU, 44–46 Earls Court Road, London W8 6EJ. (Book with Basic compiler).
LOGO
Sinclair Research, Freepost, Camberley, Surrey GU15 3BR. (Spectrum — to be released).
LISP
Serious Software, 7 Woodside Road, Bickley, Kent BR1 2ES. (Spectrum 48K).

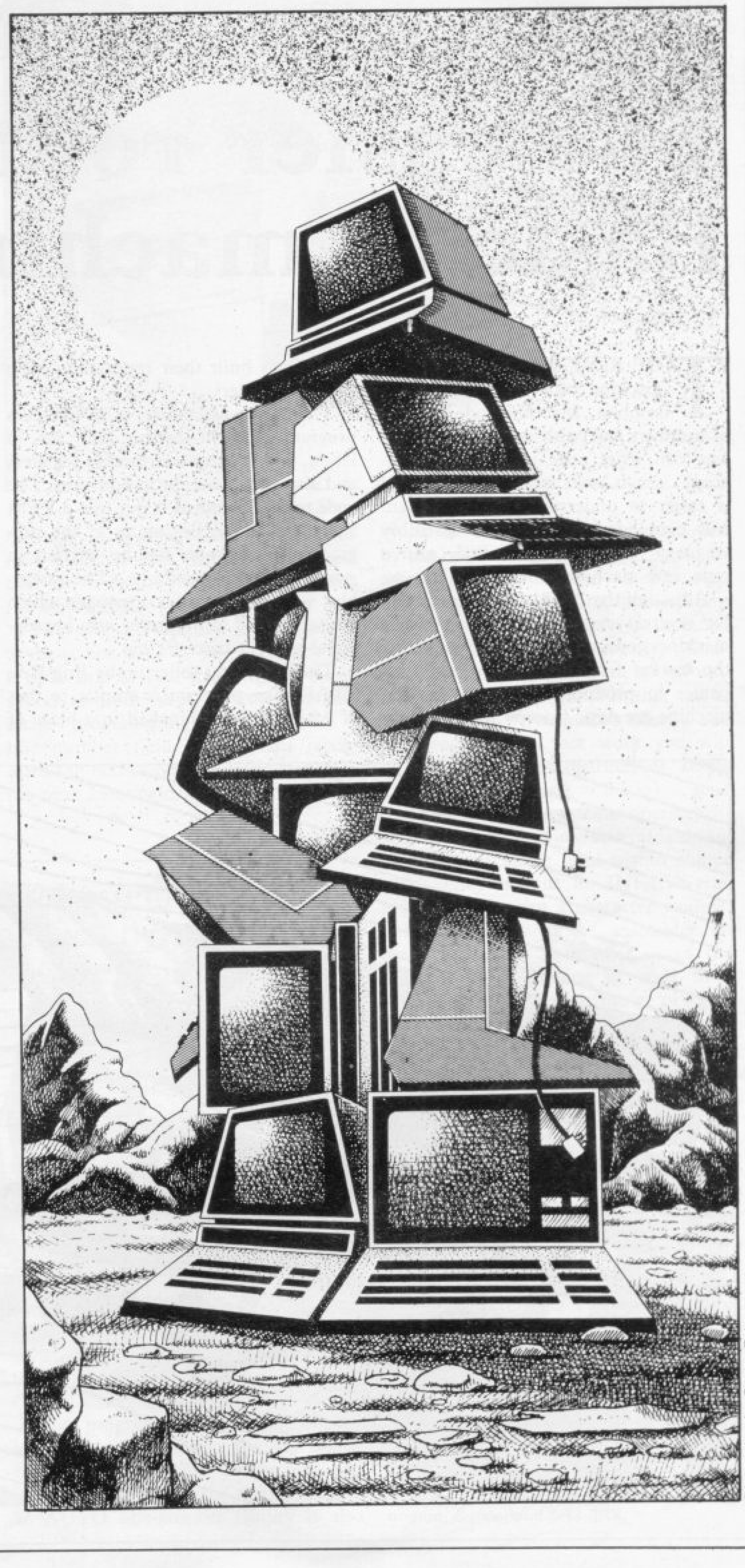


Illustration by Stuart Briers

Writing programs is time-consuming. John Gilbert examines a collection of cassettes which help to make the job much simpler

The easier route from Basic to machine code

THE BEST WAY to learn about machine code is to use an assembler. Machine code consists of numbers but with an assembler the user can write code in assembler language, which looks more like Basic and is easier to understand. The program will then convert the user's assembly language instructions, called source code, into machine code.

Although the ZX-81 has been around for several years there are only a few machine code assemblers available on the market for it. The only big companies to produce assemblers for the machine are Artic Computing and Bug-Byte. Both built their reputations with these assemblers.

The Artic assembler is a two-pass program. It will permit full use of labels, will inspect and modify registers and also allows output to a printer. The code to be assembled is put into a REM statement at the beginning of the program and all code can be written in standard Z-80 mnemonics. The assembler will also assemble messages which are to be used in programs into hexadecimal code. It costs £9.95.

The other best-seller is the Bug-Byte ZXAS. The program is similar to that of Artic but was launched in a blaze of publicity as being the first machine code assembler for the ZX-81.

Bug-Byte also wanted to be the company which produced the first assembler for the Spectrum but it was to be disappointed. Wrangling within the company between its programmers put the release date further and further back until the package became available early in 1983.

The program is for the 16K and 48K Spectrum. It is very comprehensive in its options and very easy to use. As well as assembling user machine code, it has a full editor facility with which the user

can view assembly code, delete and insert, search for specific strings of text within machine code, and list all the labels which have been specified by a user in a program within a cross-referenced table.

The editor will also reverse 16-bit values, such as memory addresses, if the user specifies that option. That facility is useful when dealing with a large number of 16-bit addresses in a long program. The use of 16-bit values can be a problem for beginners, who often do not know whether or not to reverse a number.

The program also has a good cassette interface. Both the source code — the user's — and the object code — assembled-code can be filed on to cassette. That means that source code can be saved and re-edited when the user needs it. The saved source code could also be useful if the programmer wanted to upgrade a program.

Unfortunately the manual, or lack of it, provides points against Aspect. Instructions are written on a piece of paper. They are just about adequate and contain no examples. Aspect costs £9 and is available from shops such as W H Smith.

Picturesque slipped its Editor/Assembler on to the market very quietly. The program is for the 16K and 48K Spectrum and is very powerful. It is complete with a comprehensive user manual which a complete beginner can understand.

The Editor is the part of the program which enters the source code. It is possible to enter code in the same type of format as a Basic program, as each line is given a line number. Unlike the Basic system on the Spectrum, line numbers can be generated by the program automatically with the use of the AUTO command. The use of line numbers means that the source file can be edited quickly and easily.

When the source code has been entered correctly and there are no bugs in the text, the assembler can be called into operation. The usual ORG instruction is included as part of the instruction set to define the address at which the assembled code should be put.

The program display is interesting, as the screen has a 40-column width and is split into several fields which correspond to those used in assembly language programming, together with a line number field. The cursor recognises the end of one field and jumps to the next automatically. That makes the entered source code easy to understand.

The Editor/Assembler is ideal for the beginner and could also be a powerful tool in the hands of a professional programmer. It costs £8.50.

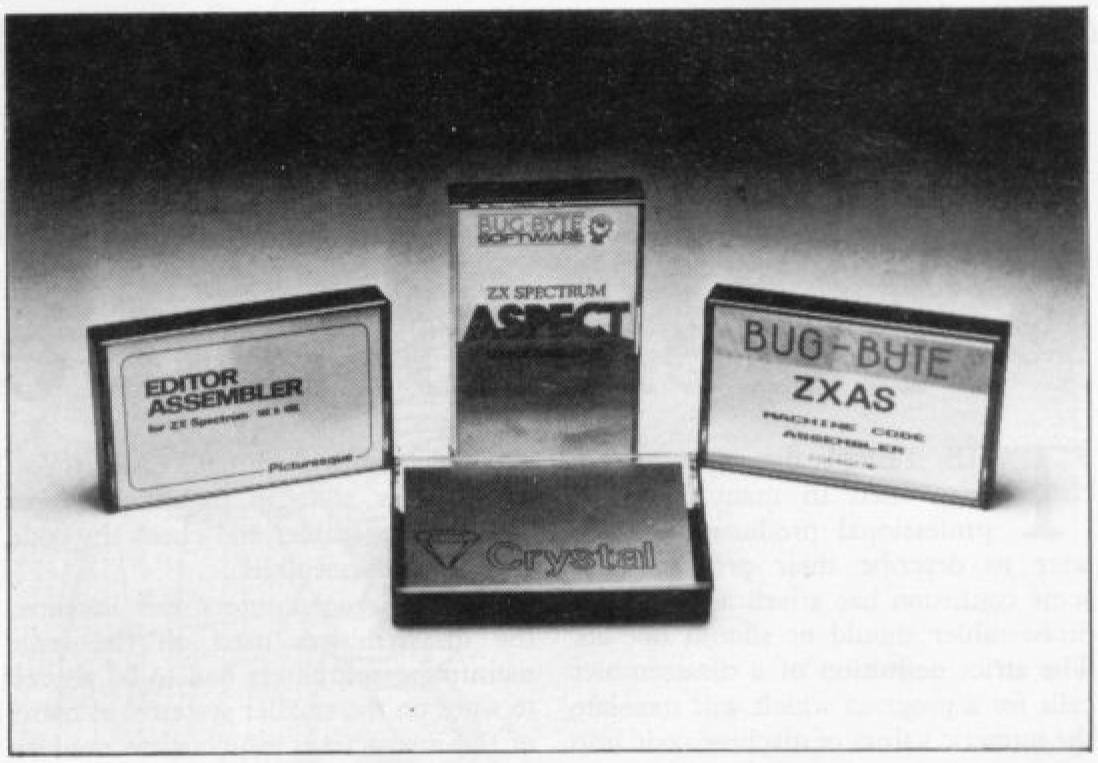
The program which caused a buzz of excitement in the *Sinclair User* offices is probably one of the most powerful assemblers which we have reviewed. It is all the more remarkable as it is produced by a small and, until now, largely unrecognised software house, Crystal Computing.

The program, Zeus, is a two-pass assembler which allows the use of the full Z-80 mnemonic instruction set. Source code can be line-numbered and an AUTO line-number facility is also available. The program is accompanied by the best manual we have seen for an assembler. It contains step-by-step instructions for entering and editing source code. An example is included which will, if entered correctly, colour the screen white, the current ink colour.

The use of an example in that way is good, because if you make a mistake and the program does not work you will have to re-learn the instructions. If the example works, users will have a good understanding of how Zeus operates.

Zeus also contains several subroutines which can be used within source code. They include an INKEY$-type function and print a character routine. Other functions in the assembler include automatic re-numbering of the source file, outputting of code to a printer, and the reclaiming of 'old' source files for further work. Zeus has been aptly-named by Crystal Computing. It costs £8.95.

There are very few good assemblers on the market although the big software houses all claim to have the best available. It is, therefore, surprising that a small company like Crystal should produce such an excellent assembler as Zeus. The reason may be that while large companies spend their money on colourful advertising, smaller companies need to rely on very good-quality products.



'One of the most powerful assemblers which we have reviewed is produced by a small, and largely unrecognised software house, Crystal Computing'.

Artic Computing, 396 James Reckitt Avenue, Hull, North Humberside HU8 0JA.
Picturesque, 6 Corkscrew Hill, West Wickham, Kent BR4 9BB.
Crystal Computing, 2 Aston Way, East Herrington, Sunderland SR3 3RX.

After dealing with assemblers, John Gilbert turns his attention to their counterparts, the disassemblers.

Making machine code easier to understand

THE TERM disassembler has been used in many ways by professional producers of software to describe their products and some confusion has arisen as to what a disassembler should or should not do. The strict definition of a disassembler calls for a program which will translate the numeric values of machine code into the mnemonics of assembly language. There are other packages which do all kinds of things with machine code but do not fulfil that definition. The correct term for those packages is toolkit.

The mnemonics of assembly language look like shortened versions of Basic keywords. They are used to represent the machine code numbers which computers can understand but which human programmers still find difficult to cope with in great quantities. It is much easier to understand a mnemonic such as RET, which means Return to Basic, than its machine code counterpart, which in this case is 201.

Most disassembler packages include a monitor program which allows the user to change machine code stored in RAM which has been disassembled. The disassembler makes the editing process easier, as mistakes can be spotted quickly if the programmer has a disassembly and not just a list of numbers which have to be sorted.

When computers had just been invented there was little software and languages such as Basic and Pascal did not exist. Programmers had to use programs which consisted of long lists of numbers. You can imagine the trouble if those programs did not work. Every number in the program would have to be checked to see if it was correct.

The invention of assemblers and disassemblers made things easier. You could enter code in mnemonic form using the assembler and check the code using the disassembler.

When microcomputers were invented the disassemblers used on the large mainframe computers had to be altered to work on the smaller systems, as many of the instructions which were used on mainframes could not be used on micros.

The first disassembler to appear commercially for the ZX-81 was ZXDB, from Bug-Byte. The company has built its reputation with that package along with its ZXAS assembler and the reason was because it was cheap and ran on an expensive computer.

The Bug-Byte disassembler was fairly standard in its performance. It allowed you to specify the address, or position in memory, at which you wanted the disassembly to start. The program would disassemble one line at a time and you had to press NEWLINE for the next instruction in memory to be disassembled.

The ZXDB for the ZX-81 used two fields, or areas, on the screen in which to display information. The first showed the address at which an instruction was located and the second showed the disassembly of the machine code instruction.

Other disassemblers, such as the Crystal Computing Monitor and Disassembler for the Spectrum, use three fields. The extra area on the screen is used to display the numeric machine code on which the disassembler is working. It is a good idea as you can check the mnemonics and machine code against each other to see if the disassembler is working correctly.

'The strict definition of a disassembler calls for a program which will translate the numeric values of machine code into the mnemonics of assembly language.'

If a company produces both a disassembler and an assembler it usually makes both programs compatible. That means that both programs can be put into the computer to work together and dispenses with the need to load one program to enter code and then load the next to check that the code is correct.

Having both programs in RAM at the same time will make programming in machine code quicker and easier but the amount of memory left for machine code programs will be reduced drastically. The advantages outweigh the disadvantages, though, so it is a good idea to buy a disassembler which can be used together with an assembler.

Two problems occur when a disassembler starts to produce question marks because it has not been programmed to recognise a machine code instruction. One disassembler which does not follow the Z-80 instruction set is ZXDB from Bug-Byte. The program uses 8080 code instead of Z-80. Instructions which are common on the Z-80 are named differently on 8080. For instance, LD in Z-80 is called M in 8080 code. Those two names mean different things. The LD instruction means load a register with a value, whereas M means move.

Trying to use a disassembler which does not understand the full Z-80 instruction set is like trying to use BBC Basic on a Spectrum or ZX-81. As BBC Basic is not the same as Sinclair Basic, the machine would not recognise the extra commands of the other language. As a result, when buying a disassembler, make sure that the program understands the complete Z-80 instruction set and not just a subset.

Disassemblers can serve two purposes. They can be used to view code which you have stored in RAM or they could be used to look at the ROM of the machine

With a good disassembler you could obtain a listing of the complete Sinclair ROM for either the ZX-81 or Spectrum. If you knew sufficient about machine code you might also be able to tell

how the ROM works. To investigate the ROM, or go bug-hunting in it, you need the proper kind of disassembler. Picturesque produces a monitor and disassembler package which suits the purpose. The disassembler has to provide facilities for output to a printer, as the Basic interpreter and operating system within the ROM are very lengthy.

A true analysis would have you jumping to different parts of the ROM when JP, or JumP, instructions are indicated. JumP is similar to the Basic GOTO instruction but there are many options available with that command and the disassembler will help you to spot them.

Your disassembler should be capable of handling the RST instruction as it is used many times in the ROM. It means ReStarT and the instruction provides a quick entry point into the ROM for programmers using ROM routines in their machine code programs.

The RST 10 instruction, for instance, would send the computer control to the part of the ROM which deals with the printing process. The routine sets up the machine to display one character on the screen.

Most disassembler packages on the market for both Sinclair machines are being sold mainly as an afterthought to assembler packages. The disassembler has a very important role in computing

> 'Many programs on the market could be streamlined.'

and the second-class sticker with which it has been labelled is unfair.

A disassembler should be a necessity when you are writing machine code programs.

If you are to buy one, take as much time about choosing it as you would when buying an assembler. In many cases it will be the disassembler which will disappoint you with its performance and not the assembler.

For instance, the Bug-Byte ZXDB was a breakthrough when it was launched but with hindsight it is somewhat mediocre. The standard of the documentation with ZXDB was far from satisfactory; as a result, the program was difficult to use.

Disassemblers are becoming more complex all the time and there is not much which can be done to improve them. Many of the programs on the market, however, could be streamlined to fit into less RAM than is now the case.

dK'tronics, Unit 2, Shire Hill Industrial Estate, Saffron Walden, Essex, CB11 3AQ.

Crystal Computing, 2 Ashton Way, East Herrington, Sunderland, SR3 3RX.

Picturesque, 6 Corkscrew Hill, West Wickham, Kent, BR4 9BB.

New ground broken by powerful architects' aid

Sinclair machines are regarded still as little more than toys. David Marsh disagrees in this review of a new cassette

APART FROM generalised spreadsheet programs of the Vu-calc and Vu-file variety, there has been little or no Spectrum software written for serious commercial applications. Something of a new departure in that direction is a specialised program, Beamscan, which is used to calculate the sizes of steel beams used in building construction.

The program is used interactively, with the screen prompting the user at all stages and asking for details of the loads. A diagram of the beam is displayed on-screen, which makes it clear exactly what information is being requested by the computer. The program seems well error-trapped and user-friendly. When all the data is in, there is a wait of about one minute while the numbers are crunched and then diagrams are displayed giving shear force and bending moment along the beam.

From its library of standard steelwork sizes, the program recommends a choice of up to eight suitable sections with stress and deflection for each. All the regular rolled steel joists, universal beams, universal columns and rolled steel channels in grade 43 steel are featured and a moment of inertia can be obtained for timber beams, from which it is a simple matter to choose a suitable section.

The beam must be simply-supported and single span. Cantilevers are not within the scope. That is perhaps the only limitation worth mentioning. Any combination of point loads, distributed loads and uniformly-tapered loads in any number up to a total of 99 can be specified.

What is more, the distributed loads do not have to extend to the end of the beam. Within the designated span of 0.3 to 20 metres, it is difficult to visualise any beam which could not be analysed by the program.

Point loads frequently consist of the end reactions of other beams. Both end reactions are given, which covers that point and also is a great help in finding the stress in any supporting brickwork.

Also given are the maximum bending moment and shear force, the deflection co-efficient, the permitted deflection and optimum moment of inertia.

That would mean that other types of beams, for example round or rectangular hollow sections, could then be chosen using the data given in BS4.

For each beam size chosen, the L/ry and D/T ratios are given, together with permissible and actual stresses, actual shear stress and deflection.

All that information can be put into the form of a calculation sheet using the ZX printer. That can then be submitted to the client or to the local authority responsible for checking the design. Although full data is given on the results and the presentation is clear and concise, perhaps some local authorities may consider it a little too concise, in that virtually no details are given on how the answers are obtained. It is therefore difficult to check the accuracy.

It is clear, however, from the printout whether or not the correct data has been typed-in and, of course, that is half the battle. The fact that a computer printout is being submitted rather than the more usual written calculations should lend a reassuring air of professionalism.

It is also probable that in the perhaps not-too-distant future when most calculations are made in this fashion, various programs will become widely-known and generally accepted in the profession as being accurate. Some kind of type-approval system might even be possible so that checking would be limited to the data output.

That would be in line with the procedure followed in other areas, where certain materials, for example building blocks, are given a certificate to indicate that they comply with building regulations. If that make is used, then no further proof is required that they meet the requirements. Extending the concept to software seems logical and almost inevitable.

Beamscan is supplied recorded on both sides of the cassette and is suitable for a 48K Spectrum, being about 35K. It is supplied with a well-written manual which guides the user through a worked example and describes the limitations of the program as 999KN for each load to a maximum total load of 10,000,000KN from the 99 loads mentioned. They scarcely seem like limitations when there are other costlier beam design programs for bigger computers which cater for a maximum of eight point loads while others cannot calculate the shear force or bending moment.

At £25, it appears costly compared to the usual programs in *Sinclair User* but it is a more specialised item with a lower volume sale than games or spreadsheets and is much less expensive than anything comparable.

It should be ideal for small architectural practices or the many one-man firms involved in smaller-scale building works. Neither is it necessary to pay a four-figure sum for the computer. The whole system — 48K Spectrum, ZX printer, TV, tape recorder and software — can be up and running for slightly less than £300 and will soon pay for itself in time saved.

Beamscan is available from Beamscan, 20 Vaughan Avenue, London, NW4 4HU. Tel: 01-202 8656.

The technique of 3D has been growing in popularity. John Gilbert finds it is not always used in the best ways

Extra dimension adds excitement to games

THREE-DIMENSIONAL games are increasing in popularity and software houses are beginning to see that 3D techniques have great potential in a market where Space Invaders and Pac-man are rapidly becoming outdated.

Programmers are starting to use sophisticated techniques to achieve three-dimensional effects on the computer screen which seem more life-like than the two-dimensional space-battle games.

The effects are difficult to create, as the programmer is working in three planes, or directions, instead of the usual two. Most shapes are produced on the screen using X and Y co-ordinates which correspond to the flat horizontal and vertical dimensions. Three-dimensional shapes are constructed using an extra axis which, in theory, moves away from or towards the computer screen.

The new axis is called Z and it is the inclusion of that dimension which gives the three-dimensional figure its depth. When a 3D shape moves up, down and sideways it uses the X and Y dimensions. When it seems to move towards you on the screen it is using the Z axis, which exists only in theory, as the computer screen is flat and two-dimensional in shape.

J K Greye was the first company to produce games using three-dimensional effects on the ZX-81. The game was called **3D Monster Maze** and the player had to move around a three-dimensional maze to find the exit and also to evade the deadly jaws of the dinosaur which prowled around the corridors.

Even those critics who dismissed the ZX-81 as a child's toy had to admit that the program was innovative and well-presented. Monster Maze marked the rise of the use of 3D techniques on microcomputers as small as the ZX-81 and Spectrum.

After the release of 3D Monster Maze, everyone could see the potential of three-dimensional games and utility packages on small machines. Using 3D techniques programmers can portray events which seem real to computer users when they are playing a game.

Three-dimensional effects also hide the inadequacies of the computers on which games are played. The dinosaur which chased the player around a maze in the New Generation game was created using the standard ZX-81 character set. That is not noticeable when the game is being played and you could be forgiven for thinking that it used high-resolution graphics.

The use of 3D gives a game added depth but at the same time it can also be used to disguise a poor plot.

The game from New Generation Software, **3D Knot,** is an example of that. While the game has a basic plot it is not deep enough when you strip away the 3D effects. That does not make it a complete failure in this case but it points to the fact that three-dimensional graphics are a means and not an end.

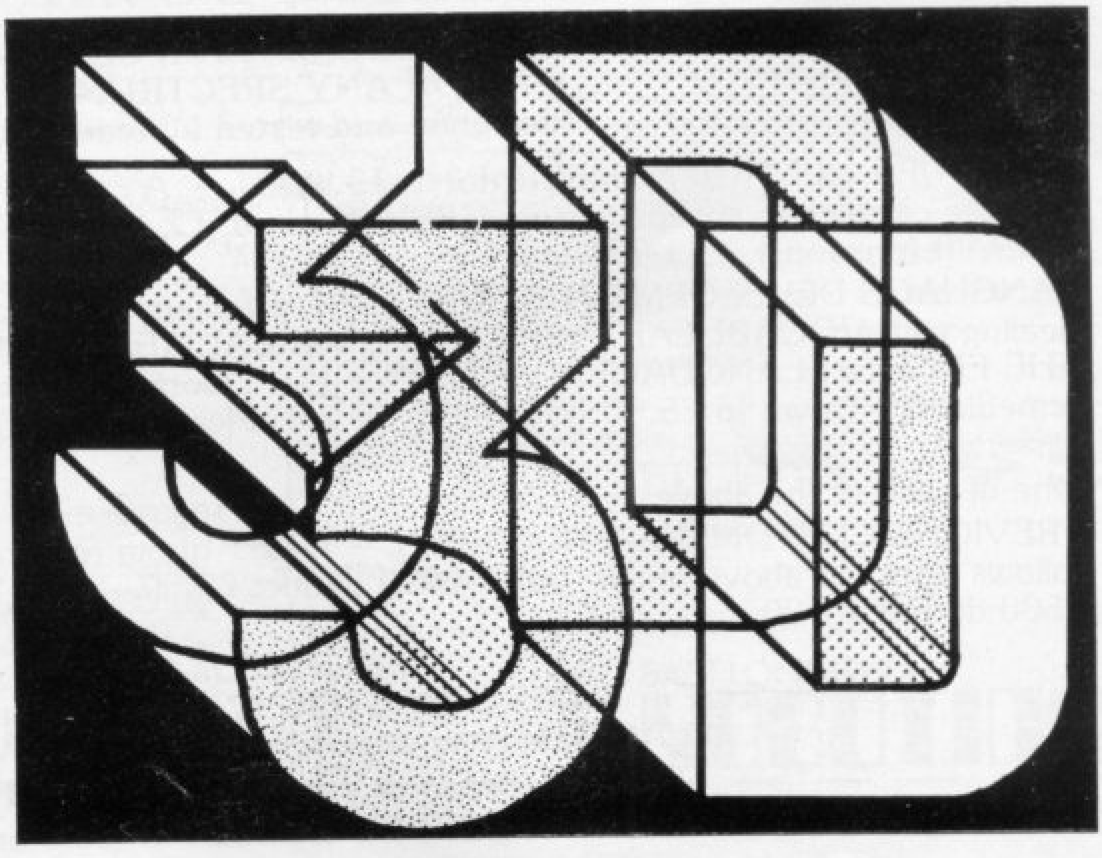
There are two types of three-dimensional effect. The first, and simplest to produce, is shown in the Artic Computing game for the 48K Spectrum, **Combat Zone.** The three-dimensional shapes are shown as line drawings with no shading. That means you can see all the lines of the shapes, including those which would normally be invisible if colouring and shading had been added to the figures.

Combat Zone, like so many other pieces of software for the Spectrum, is not new in concept. It is a version of the arcade game of the same name. Although the graphics are reasonable the animation of the line-drawn shapes is very jerky.

The plot involves the player as the last of a race of tank commanders. Enemy tanks and diamond-shaped spaceships are dotted round the landscape and it is the commander's job to destroy them.

The game involves plenty of action and credit must go to the programmers who have managed to produce images which do not bend out of perspective as they move. Artic seems to have taken the easy way out, however, as the program is so slow that it must use those notorious Sinclair line-drawing routines which are in the Spectrum ROM operating system.

The Sinclair graphics routines are not known for their speed, so it would



have been better for Artic to write new graphics routines into the main body of the Combat Zone program.

The second type of three-dimensional image is produced reasonably well in the Quicksilva 48K Spectrum game, **Time Gate.** The graphics for that type of image are more difficult to produce, as the programmer has to shade and colour the shapes to produce a picture which looks three-dimensional.

If the shading or the shape of the image is even slightly incorrect the picture will appear to be distorted and the effect will ruin the playability of the game.

The three-dimensional effects created in Time Gate show a slight distortion of image, which can be noted when an enemy ship approaches closer to the viewing screen of the player's ship. The enemy seems to unfold its wings as it gets closer and in some cases it appears as if the fixed wings materialise from nowhere.

No doubt Quicksilva would explain that as a feature of the game but all too often features such as that are errors and are explained away too easily.

The ending of Time Gate is disappointing, as the three-dimensional technique seems to have been thrown out of the window. When you have destroyed the enemy you must approach its base planet. The planet becomes larger as if an approach is being made but the technique being used is so obvious that it is embarrassing to watch.

The program uses what again appears to be the Sinclair high-resolution routines to draw circles which start small and continue to grow bigger. As they increase in size the drawing process slows and the technique becomes even more obvious.

It would have been better to do what New Generation did with **3D Tunnel** on the Spectrum and create several separate pictures in memory to switch on to the screen one at a time in sequence. That will produce an animated effect.

If Quicksilva used that technique the planet could have been produced in high resolution at several stages of approach and would have looked like a real planet and not a rope mat.

Time Gate has its technical faults where graphics are concerned but it is playable. The 3D Tunnel from New Generation, on the other hand, has what can only be described as brilliant and imaginative graphics but it is almost impossible to play to the end.

The game takes you and your laser base through an underground tunnel inhabited by rats, spiders, toads and a very impressive tube train. Unfortunately so much memory seems to have been used to create the three-dimensional effects that the movement of the player's laser base is awkward.

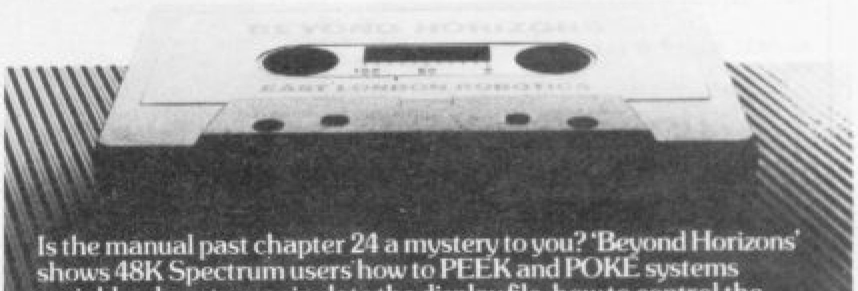
More attention could have been paid to that area of the game, as movement of the base is not smooth or quick enough. That is understandable, however, as a great deal of memory has been used to produce the displays. Apart from that problem 3D Tunnel has the best graphics for a Spectrum game.

There are still very few 3D games for Sinclair machines, although those available give confidence that this sphere is developing in the proper way.

In the next few months we could see a move away from 3D arcade games and into 3D adventure games. There are already a few graphics adventures on the market but they do not use 3D.

As 3D techniques are developed, computers will be better able to produce simulations of the real world. Adventures will become more exciting and arcade games more dynamic. The 3D field is, however, just starting to develop but it has a long way to go before perfect holographic images can be produced on a small computer.

Extend your 16k Spectrum to 48k for just £23·00. And get a Free program worth £4·50 into the bargain.



Is the manual past chapter 24 a mystery to you? 'Beyond Horizons' shows 48K Spectrum users how to PEEK and POKE systems variables, how to manipulate the display file, how to control the attributes, how to find out how much spare memory is left and much, much more. With the SP48 this program is free. On its own, £4.50.

SP48, 32K Memory Extension with Program — £23.00

Now, our SP48 offers even better value.

Because now, we're not only offering you the facility to up-grade your 16K Spectrum to 48K, we're also offering you the opportunity to be able to utilise this vastly extended memory quickly and easily.

For £23 all you need to do is plug the chip-set into the sockets provided by Sinclair on your issue 2 (or £35 for issue 1) and you have a standard 48K Spectrum fully compatible with all Sinclair add-ons and very low in power consumption.

There is no soldering required.

Fitting and removal are easy.

And the SP48 carries our full warranty and is upgradable, on a part exchange basis, to SP80.

Then all you need to do is LOAD our specially written 48K guide program, 'Beyond Horizons', and your 48K computer will guide you through its memory.

With no need to labour through manuals.

This will save you time, because we all know how easy it is to get lost with memory maps.

SP80 Paged 64K memory extension — £46

The fitting, power consumption and add-on compatibility are identical to the SP48 (Issue 1 Version £50).

It can be used as a standard 48K, but software instructions can switch to a second page of 32K.

But this is not recommended for the complete beginner.

Yet it is of tremendous benefit to the serious user. And an SP80 FORTH compiler will be available in late September.

For a fit and test service at our premises £3, and by registered post £7.

MAXIMEM ZX81. 32K — £39 (PLUS ONE USED 1K ZX81) 64K — £59 (PLUS ONE USED 1K ZX81)

The MAXIMEM 32 and 64K is a modified ZX81 and not just a RAM pack. It fits inside the case.

We also give a six-month guarantee on the computer, regardless of the age of your trade-in ZX81.

The extra memory uses only a quarter of the power consumption of a 16K plug-in memory, and does not interfere with any standard add-ons, such as printers RS232 or other interfaces.

ZX Slowloader — £10

The ZX Slowloader is a software masterpiece that will help you to LOAD ZX81 programs onto a Spectrum, saving hours of typing time.

The Next Robotics Exclusive. The Trickstick

This new joystick is a revolution in computer games control. With our Trickstick from one to eight players will be able to play simultaneously.

It will be on sale soon — see our next advertisement.

All products on this order form are in stock at the time of going to press. Same day dispatch for phoned Access and Visa orders.

To East London Robotics Ltd: Please send by return (tick items required)

- ☐ MAXIMEM 32K (£39) 64K (£59) + 1kZX81 ____
- ☐ SP48 (Issue 1 £35) (Issue 2 £23) ____
- ☐ SP80 (Issue 1 £50) (Issue 2 £46) ____
- ☐ SLOWLOADER £10 ____
- ☐ "BEYOND HORIZONS" program £4.50 ____
- ☐ Free catalogue (SAE only) ____

Postage 0.65p. I enclose a cheque/PO for ____ My Access/Visa No. is

Name ____

Address ____

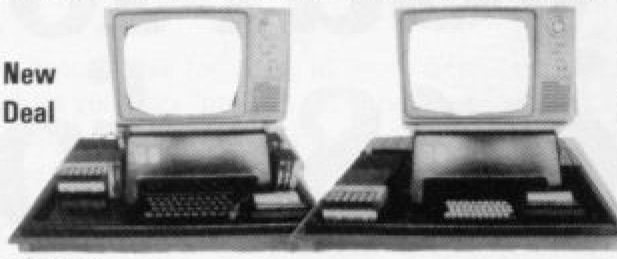
SUA

East London Robotics Ltd., Gate 11, Royal Albert Dock, London E16.
24 Hour Information Service
Tel: 01-471 3308
24 Hour Access and Visa ordering
Tel: 01-474 4715
Special Enquiries: Tel: 01-474 4430
Telex: 8813271

EAST LONDON ROBOTICS

DESKTOP CONSOLE FOR ZX SPECTRUM/ZX-81, SMALL BUSINESS SOFTWARE AND SPECIAL DEALS*

New Deal



Lower Prices

CONSOLE DETAILS:
TV/MONITOR STAND REMOVABLE FOR ACCESS TO ADD-ONS, WIRING ETC. TV/Monitor tilted to minimise eyestrain. Space for ZX Printer, Ram Packs (and other add-ons), Power Supply, Tape Recorder or ZX Microdrive and Storage Space for Cassettes. Hides most of the wiring. Console complete with two built-in illuminated power switches (one for ZX Spectrum/ZX-81 and the other for the Tape Recorder). Console colour finish: Matt Black.

Join COMPUTERLOCK SOFTWARE CLUB. Up to 20 per cent reduction on most software exclusively for Members. Members FREE Software Catalogue and FREE Post & Packing on all software.

Date ....

YOUR NAME AND ADDRESS (In Capital Letters Please) ....

Qty	Description	Item Price	Total
	Console for ZX Spectrum/ZX-81	£28.00	
	ZX Spectrum/ZX-81 Sales/Expenses Ledger for 16K/48K Spectrum or 16K ZX-81 (Free Post & Packing).	£5.50	
	*1 Console + 1 ZX-81	£60.00	
	*1 Console + 1 ZX Spectrum 16K	£120.00	
	*1 Console + 1 ZX Spectrum 48K	£150.00	
		Sub Total	
	ALL PRICES INCLUSIVE OF V.A.T.	Post & Packing	£4.00
		Total Due	

I wish to join the COMPUTERLOCK SOFTWARE CLUB and enclose my annual subscription of £10. ☐ (Tick Box).

Cheques/P.O. made payable to:

COMPUTERLOCK, 2 Wychperry Road, Haywards Heath, West Sussex RH16 1HJ Telephone: (0444) 451986.

ZX REPAIR SERVICE

At last no need to wait for weeks

Send your faulty ZX81 for fast reliable repair. We will repair any ZX81 KIT or READY BUILT UNIT.
Only **£15.95** fully guaranteed + p&p **£2.00**. Please state the nature of problem. Send cheque or postal order to:

NEXT COMPUTER SYSTEMS
Dept SUA (ZX81)
88 Harvest Road, Englefield Green, Surrey TW20 0QR

Spectrum compatible printer

- ★ Obeys standard Spectrum commands.
- ★ Printing speed 120 lines per minute maximum.
- ★ Low cost paper supply.
- ★ Price includes power supply and interfacing.

This thermal printer, now available in the U.K., is the standard printer used with the American version of the ZX81. The printer plugs direct into the Spectrum but has a through port allowing the use of additional hardware through the same connector.

The size of the printer is 7¼ x 5¼ x 3½ ins. attractively packaged in a moulded matt black plastic case.

A separate power pack is supplied in a 2½ x 3½ x 2¼ ins. case.

Two switches mounted on the printer allow for on/off switching and also access to the test mode and paper advance commands.

The printing speed in the self test mode is 120 lines per minute.

The paper rolls are 4¼ ins. wide and 25 metres in length.

£59.95 Price includes - Printer, power supply, full interfacing, one roll of paper, VAT

Please send me .... Printer(s) at £59.95 each.
.... boxes of Paper (10 rolls per box) at £12 per box (incl. VAT, p & p).
Total £ ....
Cheque/PO made payable to: Dean Electronics Ltd.
Delivery 14 days allow 28 days max.

Credit card facilities for telephone orders.

DEAN ELECTRONICS LIMITED
Glendale Park Fernbank Road Ascot Berkshire England
Telephone Winkfield Row (0344) 885661 Telex 849242

DEALER ENQUIRIES INVITED.

Hardware World

The ingenuity of the peripherals manufacturers has continued and Sinclair Research finally launched the Microdrive. Stephen Adams reviews another interesting year for Sinclair users.

Stephen Adams summarises the add-ons

Ingenuity beats Spectrum limits

THE BIGGEST blow to users of non-Sinclair-produced equipment was the announcement of the Spectrum. It knocked most memory-mapped add-ons on the head, as no provision for those devices had been made. That was because, unlike the ZX-81, there was no way of turning-off the internal RAM from the expansion port. The only thing left was for producers to put things in the input/output map or to supply RAM packs or RAM upgrade kits.

The I/O map was already full of Sinclair devices, allowing the use of only eight separate devices, if they wanted them to be compatible with the Microdrive. Nevertheless, ingenuity brought forward a large number of new devices after the initial shock had worn off.

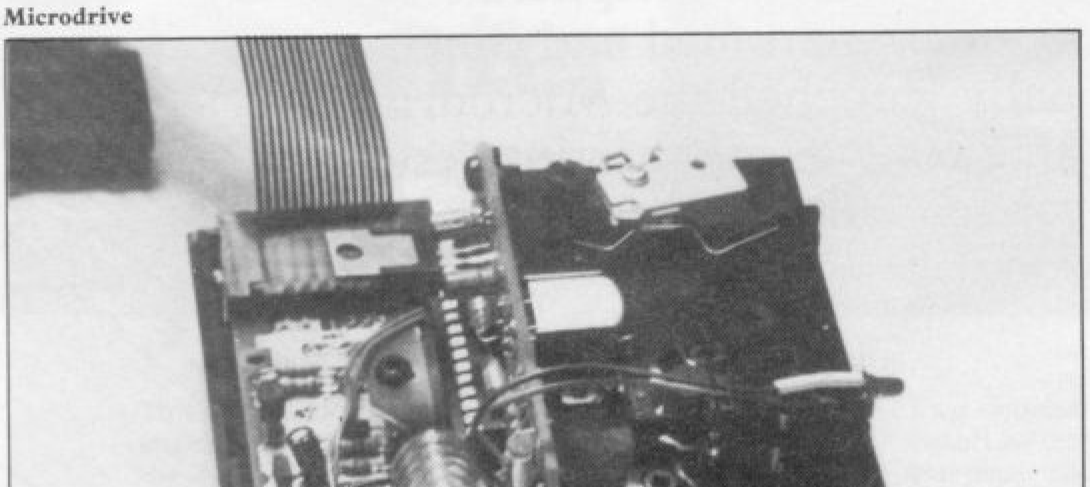
RAM kits for upgrading the 16K Spectrum to 32K are now readily available for about £21. Fox Electronics, for example, supplies a kit for £20.99 and it has instructions on how to improve your TV display. They are easy to fit and now have simple instructions on what to do, if you can overcome the fear of opening the case. The kits are usually identical to the proposed Sinclair upgrade but will fit only on model 2-type Spectrums. East London Robotics can also supply an extra 64K kit which is paged.

For those with a model 1 Spectrum, identified easily by the grey keys, there is the choice of an add-on board made by Downsway Electronics or the Spectrum 32K RAM pack by Cheetah Marketing. The Spectrum RAM pack will also fit on to any other Spectrum, as it plugs into the expansion port.

Another memory-mapped device available this year was in a surprising place, inside the Spectrum 16K ROM space. It was the Orme Electronics ROM containing RENUMBER, block delete of Basic lines and many other useful routines in a 2K ROM. That can be very useful, as it is immediately available on power-up.

> 'Printers and other hardware devices make the software come alive'

Microdrive



Sinclair has also provided two new devices for use in the ROM space, which should please hardware and software users. One is the Microdrive Interface One which not only contains the controlling ULA and hardware to run the RS232/network/highspeed cassette Microdrive but an extra ROM which can be used to write your own Basic commands. The other is the ROM cartridge system available with Interface Two.

The RS232 can handle a printer easily with its limited handshaking ability but still is limited for input from an RS232 device, as all that is under software control and not a hardware chip. The network is a very simple arrangement which allows you to talk between 64 Spectrums but I have no doubt that someone will write similar software to use it with the ZX-81 through the cassette sockets. Two books which should help in this respect are those by Andrew Pennel and Dr Ian Logan.

The Pennel one also contains an ON ERROR GOTO machine code routine which should trap 90 percent of Basic errors and send them to a program line to deal with them.

Sinclair has also launched the Interface Two, containing two joystick sockets which will operate the first or last set of five number keys. That will mean that software will have to be re-written to use Sinclair joysticks, as the accepted standard at the moment is the one set earlier in the year by the Kempston joystick interface which operates as an I/O device, depending for its decoding on A6 only being low.

Much software has already been converted to use it and I cannot see software writers wanting to do it all again. Programmable joy-sticks are just starting to reach the market, which will eliminate the need to re-write the software.

For the Spectrum and soon the ZX-81, the Stonechip one seems to be best. For ZX-81 and Spectrum users, a cheaper hardware-based version, the Pickard controller, can be used as it fits both computers by plugging into the keyboard sockets. The AGF version uses crocodile clips to set up the five keys to be used; it is a little unstable but it plugs into the edge connector and does not require entering the machine.

The Interface Two also has a single ROM socket which allows you to use Sinclair-designed cartridges. It is not known whether software suppliers will want to try to fit normal ROMs to a port like this. That, and the fact that recording Microdrive cartridges is a slow process, will limit the amount of software available from software manufacturers on Sinclair devices. EPROM software for ZX-81s is available from Eprom Services and Audio Computers.

Both, however, concentrate on the machine code user and not the games player. The fall in price of the ZX-81 and the amount of hardware available for it has started to attract the business user as a control processor for a robot or controlling some industrial process. That is because it contains all the requirements of the development system on one board — Z-80-A, one of the most popular processors, working at 3.25MHz, TV interface, cassette interface for program storage, alphanumeric keyboard, Basic as well as machine code monitor for designing programs, and an expandable memory map and I/O map.

There is also a wide range of machine code assemblers, dissassemblers, EPROM cards and blowers, RAM, battery-backed memory already available, costing just a few pounds. That to the industrial process manufacturer is peanuts, as a development kit from a chip manufacturer would cost more than £200 with far fewer facilities. The ZX-81 costs only £45 with a massive 16K memory included.

Plain-paper printer interfaces for the Spectrum abound and Tasword, the word processing program, can work with most of them. Hilderbay and Kempston provide software-driven Centronics versions, while Morex provides both RS232 and Centronics outputs. Deans also introduced its version of the Timex printer, which uses a much better paper than the Sinclair, gives a clearer print in both black and blue ink, and will use Sinclair commands to control it, so there is no need for extra software.

Printers and other hardware devices make the software "come alive" and that combination makes the Spectrum a very powerful business computer.

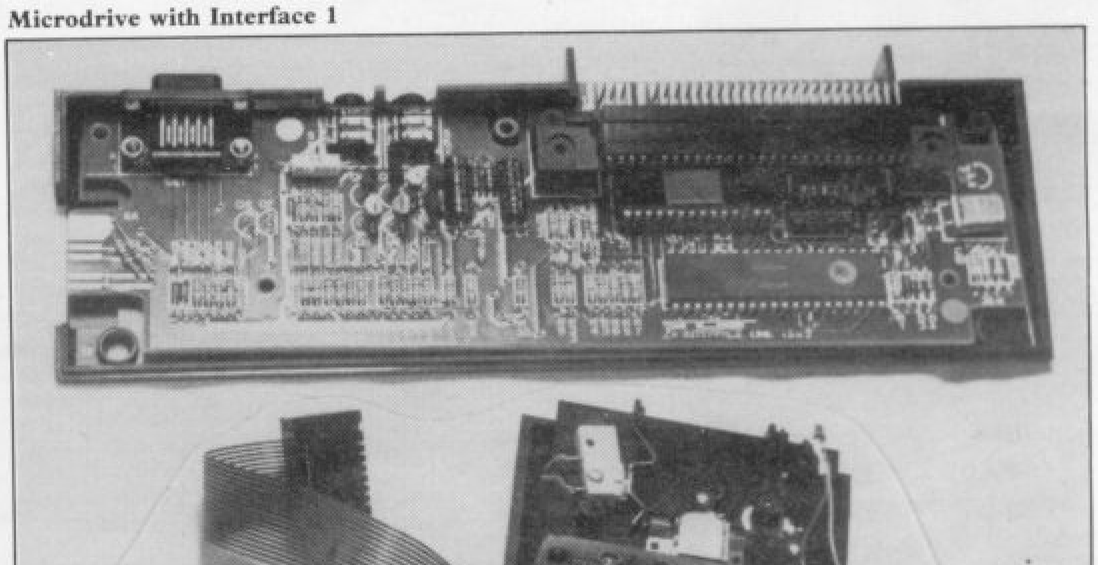
That is the most important event of the year, as the more software which is written to use the large amount of hardware, the more that hardware will be used. Voice output units, modems — to talk to other users or databases over the telephone — RS232 interfaces and light pens all rely on good software to make use of them.

The Cheetah Marketing Sweet Talker, for instance, works better because it is accompanied by an instruction tape which not only demonstrates how to use the unit but also shows the user how to structure programs to make it easier to use in their programs.

Modems from Maplin, Ambit and Micronet — for Prestel — will allow users of the Spectrum and, in the first two cases, the ZX-81, to talk to many other computers. Some of them will be other types of computers, like the BBC and the Commodore 64. The others will be maintained by public and private companies which maintain large amounts of information and programs on their computers.

Micronet 800 is a database maintained on Prestel computers as a 24-hour-a-day, seven-day-a-week computer club. It has hints and tips on Spectrums — ZX-81s at the moment cannot use the system — as well as news, free programs

Microdrive with Interface 1



AGF programmable joystick

and a mail box facility. Keyboards and consoles also have been making their presence felt as ZX-81 and Spectrum users want to upgrade their machines. Plastic cases from W H Smith are about the cheapest containers at £3.99 and will take a ZX-81 or Spectrum, as well as a few add-ons. The d'Ktronics keyboard and case or that from Fullers appear to be the most popular but neither will allow the use of the Microdrive interface without taking it out of its case.

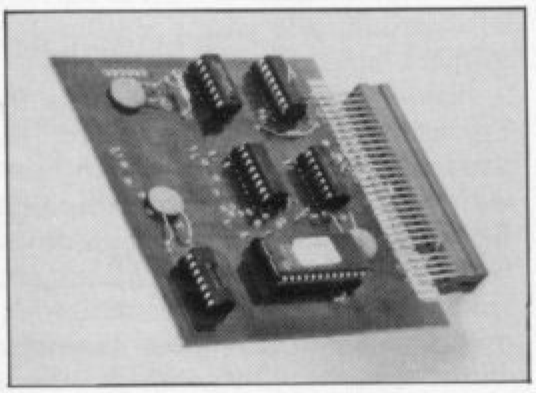
The Filesixty button set is a cheap alternative for ZX-81 users which gives the advantage of spring-loaded keyboard the same size as that of Sinclair, but without costing more than £10.

Colour for the ZX-81 is now available in a simple form for all PAL television users — that is, most of Europe. The black box requires only two wires to insert between the modulator and the breaking of tracks and can be re-connected if required. The rest of the box plugs into the back of the ZX-81 to give black characters on a choice of 16 coloured backgrounds or coloured characters on a black background from DDC. It is also more stable than the Spectrum and requires no extra memory to use it. It will work even on a 1K machine.

One device which requires extra memory is the excellent High-res screen — 192 by 256 pixels — and user-definable graphics package made by Nottingdale Technology Centre. It uses the ZX-81 internal 1K RAM for its system variables and 6K of memory for the storage of the screen in the program. No internal wiring is required, as the unit plugs directly on to the back of the ZX-81.

For the Spectrum, a unit has appeared which will be of great delight to children and disabled users. It is the Currah microSpeech unit which plugs in to the back of the Spectrum and, on command, will speak the key pressed. That happens during program input as well as INPUT and INKEY$.

It requires no programming of the



Orme Electronics EPROM read card

speech by the user. Speech output can also be programmed by using S$ to contain the alphones — sounds which make up words — which are then spoken immediately. Thus input and output can be spoken rather than read.

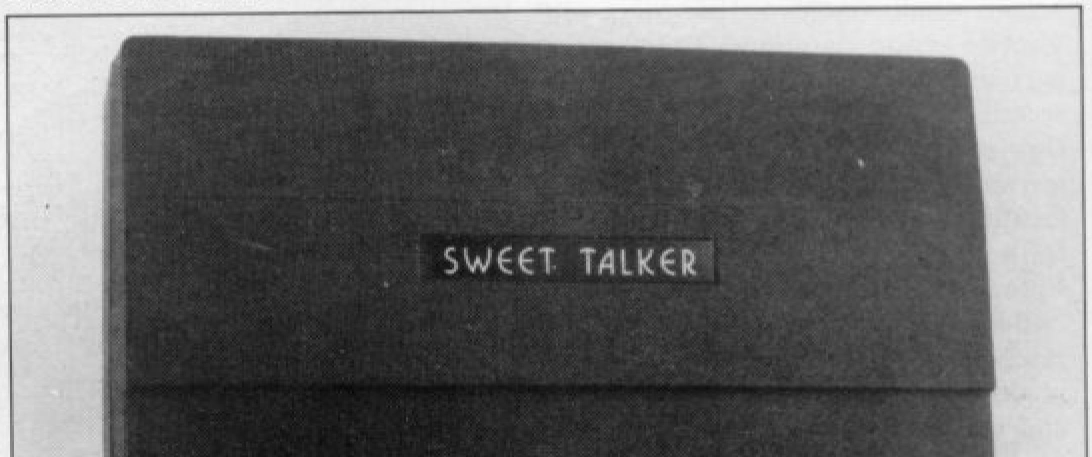
Tapes have been causing problems ever since the ZX computers came into existence; the Spectrum is better than the ZX-81 but still can be improved with some extra hardware.

Tape filters and switch-controlled SAVE and LOAD devices are available from several firms, like Abacus and Elinca. There have now been three Spectrums produced by Sinclair. The model 1s, which can be identified by the grey keys or by looking through the expansion interface and seeing an IC socket on the left-hand side; the model 2s have the large, black, ROM chip there. Those Spectrums need their extra 32K of RAM mounted on a printed circuit board before they can be put into the computer. There are no Sinclair RAM boards available for the machine.

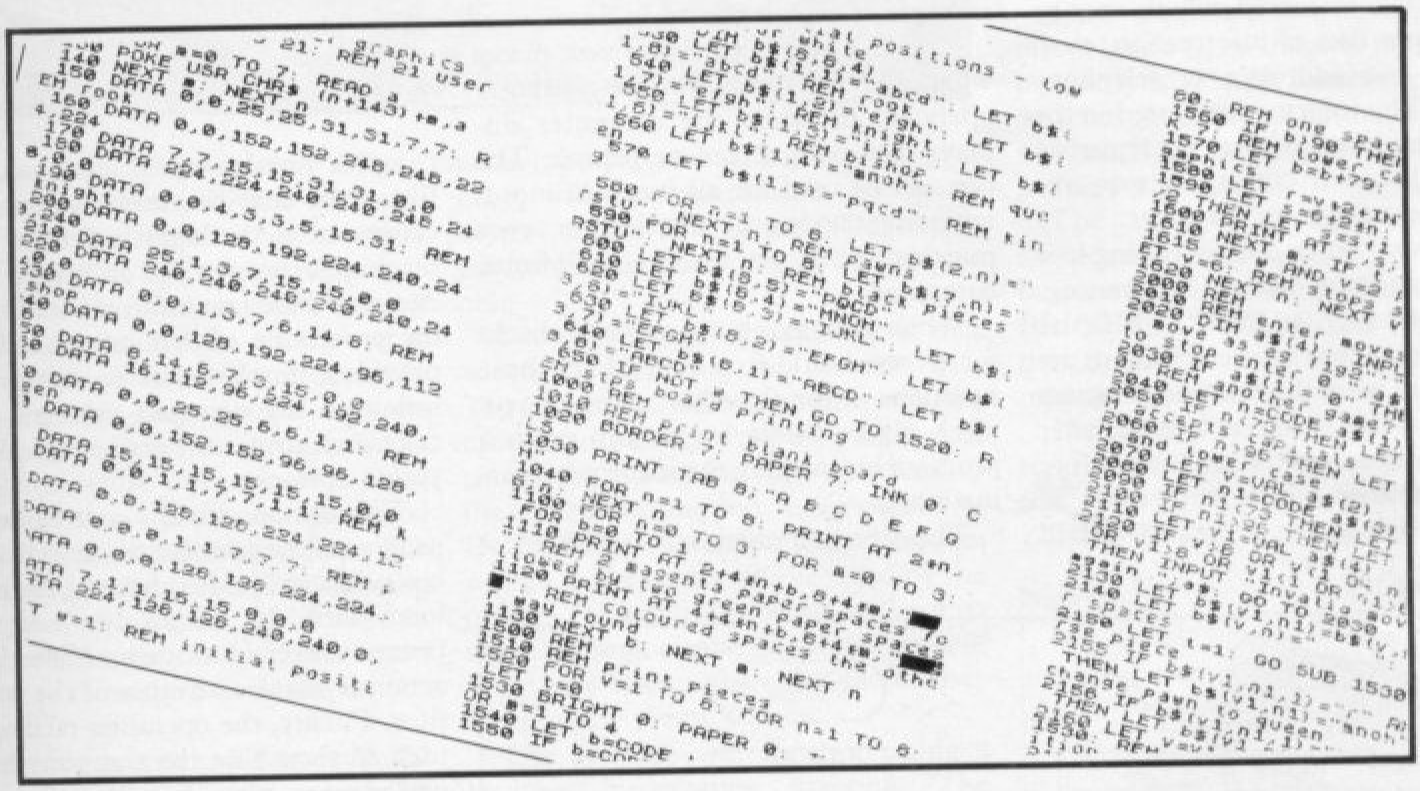
The model 2s were re-designed completely by a computer and the ULA was changed to get rid of an extra IC which had to be inserted in the model 1s, due to a design error. The extra 32K of RAM which can be added to 16K machine now requires only chips to be plugged into sockets on the board. Model 1s and model 2s have had to have an extra transistor fitted to prevent a clash between the keyboard and the ULA TV interface.

The latest model 3s have also had their internal circuitry re-arranged and the ULA updated to give a wider tuning range on TV sets. It has also caused some software problems, as the keyboard inputs are no longer held to binary 1 — +5 volts — when not in use. That was done to reduce the power requirements of the ULA.

Cheetah Sweet Talker



Programming



During the year we published a series to help you learn how to program and occasional articles on how to add gloss to your listings. We now collect them to make a complete programming course.

Follow route to better programs

Basic Sinclair coding with John Gilbert

MANY PROGRAMS written by beginners to computing show a lack of what professionals call structure. The structure of a program is the way in which it is put together and the order in which the code—the language in which it is written—is put down in the finished product.

To help the beginner with how to structure programs so that they will work faster and occupy less memory, a database is constructed which can be used to store lists of information, such as names and addresses or telephone numbers. The information stored in the database can be called back by typing-in a keyword which corresponds to the information sought by the user.

Before we start to do any coding it is important to know about flowcharting. A flowchart, such as the one in figure one, is constructed before coding to act as a guide to what the finished program will look like. If it is written after coding has been completed and before the programmer starts to search for errors—to debug the program—it will be an aid in finding redundant code or code which inhibits the flow of a program.

The program flow is the way in which the program will be executed. It is important to have that correct or errors will continue to occur in the program and the speed of the program will be slower.

The basic flowcharting symbols are the Input/Output box, operation rectangle and decision diamond and an example of each is shown in figure one.

The I/O box is used to mark places where an entry is made by the computer operator, or when the computer displays data on a screen or printer. The box can be used for all forms of input, including keyboard, joystick, or even punched cards on a large mainframe computer.

When debugging a program, checks for errors should first be made at those junctions in the flowchart, as the boxes mark places where a user can crash a program by entering the incorrect information.

The decision diamond is the most

complex operation box in a computer flowchart—and the most necessary. A computer is distinguished from other machines through its ability to make decisions based on information. Usually the processing of that information will provide a simple yes or no answer. The inflow to the diamond descends vertically and splits in two to provide the yes/no options.

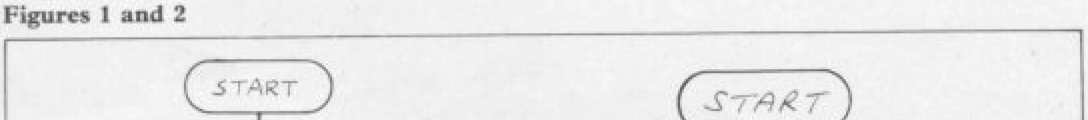
The option which contradicts the program flow goes out to the side of the box and can be directed up, to form a loop until the action has been performed correctly, or down if alternative action is required to that of the normal flow. Finally, the operation rectangle is used to show that the computer has to perform some kind of calculation. That may be adding numbers, assigning numbers to variables, or scanning a string of characters. The use of that and the other boxes is illustrated in figure one.

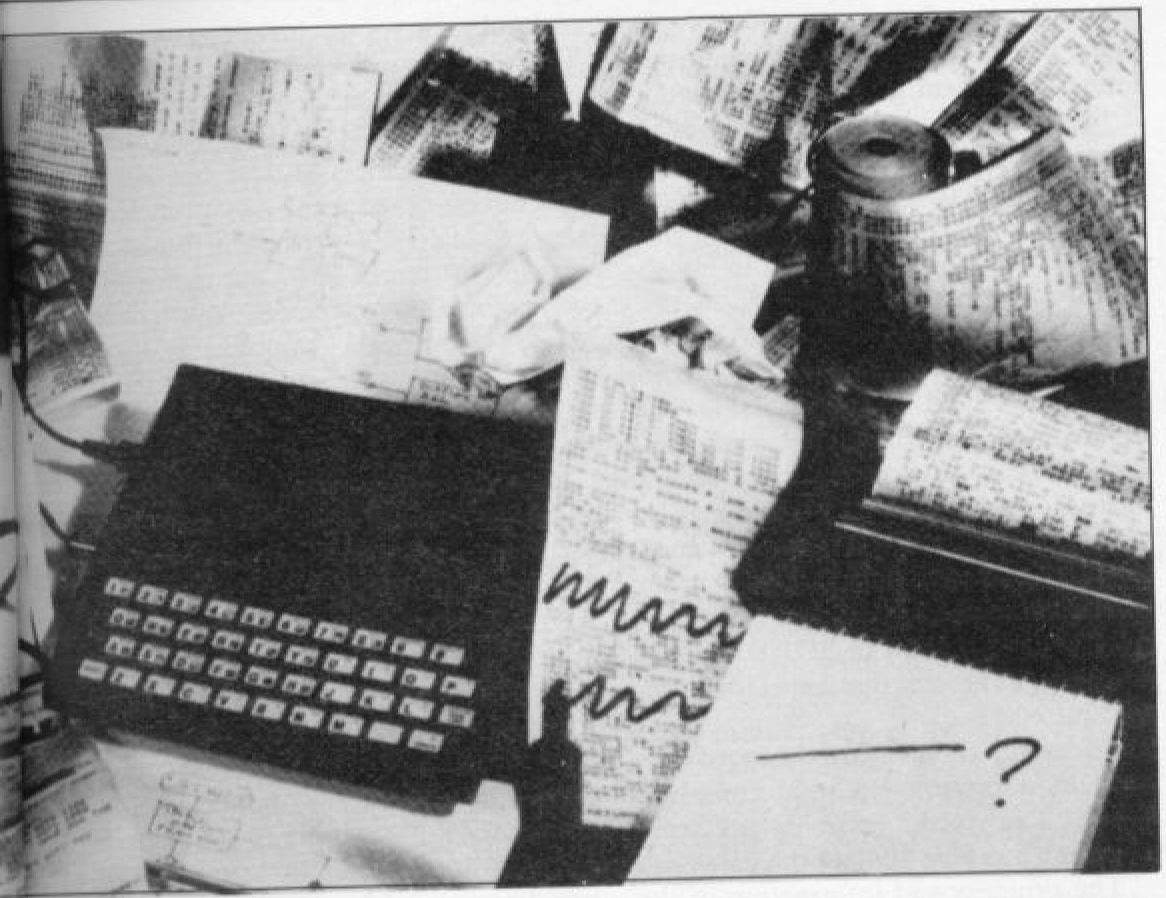
Flowcharts usually are constructed before writing a program but it is a good idea to draw up one from the finished program to see if the program flows as it was originally intended.

When drawing a chart the boxes should be balanced as much as possible to the left and right of the main stem of the flow. The whole point of flowcharting is to create an easily-understood diagram. The labels inside the symbols should be written in English and not in Basic.

The diagram in figure one uses several decision diamonds and they branch to both left and right. A flow on just one

Figures 1 and 2





side of the diagram looks sloppy if there are more than two decisions to be made.

The way not to structure a flowchart is shown in figure two. The flow lines at the side have been run together, making it almost impossible to decide what happens next. That is remedied easily by making the chart longer and re-structuring the lines into separate boxes as shown in figure three.

When writing a program it is a good idea to draw several flow diagrams. The first would be an overall plan showing the sections of program to be written and subsequent diagrams would expand each box to show the flow of the various routines.

A program is structured in a similar way to a flowchart. Most programs are constructed in the way figure one shows. The technique is called modular programming because the structure is broken into subroutines, or sections, called modules.

The reason is to eliminate as many GOTO statements as possible, or to make a GOTO statement jump only to a part of the routine in which it is situated, i.e., to make what is termed a local jump, or the control routine at the top of the program.

The control routine consists of a series of GOSUBs. It is the part of the program which is used most, so it is the first thing the computer encounters when scanning the program. In that way the program is faster in execution, so it becomes more efficient.

A control routine can have two distinct structures. The first is used in a game-type program. That type of program will execute routines by going down through each of the GOSUBs in turn and then returning to the beginning.

The other type of structure is that which we shall use for the database. The program will first jump to the menu routine where the user will select an option. Control is then sent back to the control routine and, using a series of IF . . . THEN statements, the program will go to the subroutine selected by the player. The control program will not go down through all the GOSUBs but will jump only to those specified by the user through the menu.

The database will function using a keyboard recognition function. The user enters a few words which act as a key phrase. The program will then look through the list, or file, of information in the program and, if a match is made between the key phrase and part of a piece of information, that piece of information will be output. The computer will output all information which is associated with the key phrase.

The program will also have to support separate data files and be user-friendly so that anyone can use it. The program menu will have search files, display entries, load or save files, and will create new files. The program structure will look like this from top to bottom, with the control routine at the top.

The complete flowchart of the database is shown in figure one. Using that chart it will be easy to translate each operation into program code.

With programs such as the database, which is being constructed, the screen display is very important. The computer uses the screen to display information to a user.

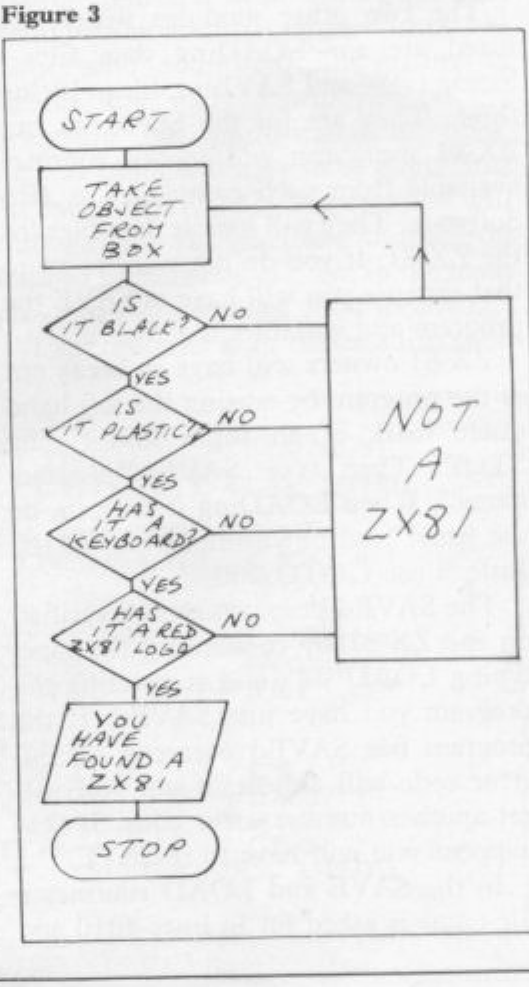
Many people who use programs such as databases or word processors will know nothing about how a computer works or what it expects as input. The more information given to the user by a program the better. That does not mean clogging the screen with vast quantities of text. Instructions on the screen should be easy to read and well-spaced. The clear screen command should be used as often as possible to break-down information into easily-read pieces but at the same time the jump between one screen of instructions and another should be almost transparent to the user.

The text must flow naturally and there should be no illogical jumps in the instructions. That is a matter of experience but the database has been designed with clarity of instructions and prompts for inputs in mind.

The most important instructions should stand out from the rest. That can be done by liberal use of the commands BRIGHT, INVERSE and FLASH on the Spectrum and the GRAPHICS mode on the ZX-81.

Listing one is the menu subroutine for the database. It is written for the Spectrum but is easy to re-write for the ZX-81. To do that take out the IN-

Figure 3



VERSE in 1020 and 1021, BRIGHT in 1021 and FLASH in 1040. INVERSE can be substituted with the GRAPHICS inverse lettering of the ZX-81 and the title MENU should have spaces between each letter to make it more prominent.

The menu should not be surrounded by a border of a different colour, as that can confuse a user and make the screen look crowded. It is also important not to use excess graphics commands. A display with FLASH or INVERSE all over the screen is just as confusing as if they were not there.

One other thing which people tend to forget is that they have the use of all the screen. Do not cram things into the corners or sides of a screen. Titles should be central and if there is only an input prompt on the screen, a good place to put it is the top left-hand corner.

Listing one is the first module of the program. If there is a menu in a program it should be situated at the top of the code after any single or array variables have been declared, as in line 100 of listing one.

The menu module will display the options available from the program. Ask which you require, put the number of the option selected in a string variable A$, and transfer to the control program, also in listing one, at the top of the code. The control program, consisting of a series of IF . . . THEN instructions, will then transfer to the subroutine selected from the menu.

The two other modules which are listed are for LOADing data files—listing two—and SAVEing files—listing three. They are for the Spectrum but ZX-81 users can add toolkit routines available from such companies as JRS Software. They will handle data files for the ZX-81. If you do not want to go to that expense you will have to SAVE the program and variables together.

ZX-81 owners will have to break out of the program by erasing the left-hand quote mark of an input and typing STOP. Then type SAVE "program name". When LOADing it back in, do the usual load but do not press RUN. Instead use GOTO 200.

The SAVEd program can be verified on the ZX-81 by re-winding the tape, typing LOAD " ", and re-entering the program you have just SAVEd. If the program has SAVEd correctly an 0/0 error code will appear; if not you will get another number error code. If that happens you will have to re-SAVE.

In the SAVE and LOAD routines a file name is asked for in lines 4010 and 5010. On the Spectrum a file or program name can be only 11 characters long. To prevent the program crashing if the name typed-in by the user is too long, lines 4020 and 5020 will take only the first six characters of a file name. The other letters are cut off or truncated. Truncation is also used to cut the YES/NO inputs to Y or N.

The displays in both the SAVE and LOAD routines are important. Error messages should be displayed in the same position on the screen every time they appear. Every program should have a standard error message area to which the user will become accustomed. In the case of the database, that is in the middle of the screen. Also the error message in line 800 is BORDERed in red to warn the user that a specific input is required.

Be careful about using the word "input" in a prompt, as it is computer jargon. The word "enter" is preferred as is shown in line 1040 of the database.

The prompts and information in the SAVE and LOAD routines may seem simple and not worthwhile but the writer knows about the internal workings of the program and what to enter. A newcomer to a program, on the other hand, needs to be taken through it step by step.

When a module of a program has been written it is a good idea to test it on a friend or relative who knows nothing about computers. If they can follow the prompts, leave the display as it is; if they are confused, you know you have more writing to do.

Data structuring comprises a series of topics which most beginners like to leave alone, unless they are doing O or A level courses in computer science. There is only one data structure in the Basic language and that is the array. An array is a series or table of data items which are grouped under one name. They are indexed using a number. Most professional data processing programs are not written in Basic but in machine code, because of speed restrictions. Also, using data compression techniques in machine code you can enter much more data into the machine than would normally be possible.

Databases written in Basic do not often use arrays as it is better to use long strings of characters into which the data is put, or concatenated, as it is entered. In that way a user does not have to specify the length of a file or how many data items can be entered as a maximum.

Arrays occupy more memory space and their length has to be fixed before data entry but sorting and searching is easier to understand by using one- or two-dimensional arrays, as we can deal with data one element at a time.

The file creation and data entry routines of the database being constructed are shown in figure four.

The listings shown are for the Spectrum but can be used for the ZX-81 with the changes mentioned earlier. Line 100 of listing one has been altered slightly to contain another array, c$. That array will hold the number of the next available space in a file, the maximum number of data items it is possible to store, and the maximum length, in characters, of a data item.

In the file creation routine the user can define the file length and word length but if they are over the maxi-

Listing 1

```
 100 DIM b$(100,20)
 200 GO SUB 1000
 300 IF a$="1" THEN GO SUB 2000
 400 IF a$="2" THEN GO SUB 3000
 500 IF a$="3" THEN GO SUB 4000
 600 IF a$="4" THEN GO SUB 5000
 700 IF a$="5" THEN GO SUB 6000
 800 INK 6: CLS : BORDER 2: PRIN
T FLASH 1;AT 10,8;"ENTER A NUMBE
R 1-5": PAUSE 200
 900 GO TO 200
1000 PAPER 0: BORDER 0: INK 4: C
LS
1010 PAPER 0: BORDER 0: INK 4: C
LS
1020 INVERSE 1
1021 PRINT INVERSE 0; BRIGHT 1;
INK 6;TAB 14;"MENU"
1030 PRINT AT 5,7;"1  SEARCH FIL
E    ";AT 7,7;"2  DISPLAY FILE  "
;AT 9,7;"3  LOAD FILE      ";AT 1
1,7;"4  SAVE FILE      ";AT 13,7;
"5  CREATE FILE   "
1040 INPUT '''' INK 6; FLASH 1;"
ENTER OPTION (1-5)";a$
1050 IF a$="" THEN GO TO 1040
1060 RETURN
```

```
4000>REM LOAD ROUTINE
4010 CLS : INPUT "NAME FILE TO B
E ENTERED ";c$
4020 IF LEN c$>6 THEN LET c$=c$(
1 TO 6)
4030 IF c$="" THEN GO TO 4010
4040 PAUSE 10: CLS : PRINT ''''
'' FLASH 1;"SET UP TAPE AND PRES
S NEW LINE": PAUSE 0
4050 PRINT '' INVERSE 1;"FILE '"
;c$;"' LOADING"
4060 LOAD c$ DATA b$()
4070 PRINT ''''''TAB 10;"FILE L
OADED          ": PAUSE 100
4080 RETURN
```

Listing 2

mum bounds, the computer will set them automatically to 100 and 20 respectively. The data entry routine is not yet included on the menu. It is best to put it in the menu routine at line 750 and set it to the Entry Subroutine at line 7000. The menu display at line 1030 will have to be changed so that "ENTER DATA ITEM" is displayed under "CREATE FILE".

When those alterations have been made, the enter routine at line 7000 can be added to the listing.

Figures five and six show program listings for two techniques which can be adapted for use with the database. They are two of many. The study of data sorting techniques has interested academics for a long time and some sophisticated algorithms, step-by-step methods of solving problems, have evolved.

Figure five shows the Bubble or Shuttle Sort. It is fairly simple to use but very slow when dealing with large amounts of data. It is called the Bubble Sort because the data is sorted into a list in ascending order with items in the incorrect order 'bubbling' up to the top. The data items are sorted using a keyword, usually the first word of or number in the item. The Bubble Sort will take the first character in an item and compare it to the first character of the next data item.

You may like to change the routine to look at the first three or four letters of a word. The routine can take anything up to three minutes to sort 50 items of data. If you are not concerned with the time element, the routine is easy to program and use.

In the program, line 8010 defines the pointers being used in the routine. Pointer p1 will show the first item on the list and p2 the one beneath it. The variable sc will contain the number of swaps made in one pass. When that is zero at the end of a pass, the sort is concluded.

Line 8020 will send control to the swap routine if the first data item is greater than the second. Line 8030 advances the pointers and line 8040 will return to the menu if the sort is complete. Line 8070 returns control to make another pass of the data.

The swap routine starts at line 8080 and 8090 will swap the data items which are in the incorrect order. Line 8100 returns control to the main sort routine and line 8110 informs the user that the sort is complete. The routine can be improved by adding a line to make the computer display a message, such as 'SORTING', to inform the user a sort is taking place.

The Basic listing in figure four can be used as a base for the other sorting technique known as the Shell Sort and the line numbers which have been modified are shown in figure six.

The Shell Sort, shown in figures six and 7b, is slightly more complex than the Bubble Sort but will run faster with large amounts of data. If a relatively small amount of data, for instance 100 items, is to be used, the Bubble Sort will do the job just as quickly as the Shell Sort.

For the Shell Sort the data file is split approximately into two halves. The pointer p1 is, as with the Bubble Sort, set to the first data item in the file. The pointer p2 is set to the item halfway through the data file.

The item pointed to by p1 is compared to all the items in the second half of the file and swaps are made, as before, if p1 is greater than p2. When p1 has compared item one to all of the last half of the file it will point to the second item and again compare that to the second half of the file.

The sort is finished when either p1 or p2 arrives at the last item of the file and the sc swap counter variable is 0. That shows that no swaps were made in the last pass through the data and the information in the file is in the correct order.

Both structures created by the sort routines from figures five and six are illustrated in figure seven. As I have said previously, they are just two of many routines and it is possible to create several other sorts using the two basic structures outlined.

There are many methods of searching data and no technique is better than another but in the proper circumstances one technique can be used in favour of another. The Binary Search deal is one of the fastest searching routines and one of the easiest to learn.

Obviously, speed is important in any searching routine. Someone cannot wait for an hour for data contained in a list of 30 items to be retrieved by the computer. If that were the case it would be easier to search through a list on paper.

One method of searching a list would simulate a person doing it by hand very well. That is called the Serial Search—figures eight and nine—which can take up to half an hour to search a list of 100 names and addresses. The listing in figures nine and 11 will run on the Spectrum but multi-statement lines, such as 2030, will have to be eliminated for use on the ZX-81. The routines will run without the database but if you want to include one of them in the main program you use lines 2000 to 2040.

The computer scan the data list, stem

Listing 3

```
5000 REM SAVE ROUTINE
5010 CLS : INPUT "ENTER FILE NAM
E ";c$
5011 IF c$="" THEN GO TO 5010
5020 IF LEN c$>6 THEN LET c$=c$(
1 TO 6)
5030 PAUSE 10: CLS : PRINT ''''
'' FLASH 1;"SET UP TAPE AND PRES
S NEW LINE": PAUSE 0
5040 PRINT '' INVERSE 1;"FILE '"
;C$;"' BEING SAVED          "
5050 SAVE c$ DATA b$()
5060 PRINT ''''''"FILE '";c$;"'
SAVED."'''"REWIND TAPE TO VERIFY"
'"AND PRESS NEW LINE.": PAUSE 0
5070 VERIFY c$ DATA b$(): CLS :
PRINT '''''''"FILE '";C$;"' VERI
FIED"
5080 RETURN
```

by item, trying to match the key, a name or telephone number entered by the user with the items in internal memory. When a match is made the item will be displayed in full on the screen.

The length of time of the process will depend on the speed of the high-level language, such as Basic, or the speed of the central processor if the program is written in machine code. The number of items in the list to be searched will also be a factor in the amount of time taken by the computer.

If you are planning to use only a few data items you could use the Serial Search technique. The coding is simple. Use a variable as a counter to point to each data item in the listing in turn. Increase the pointer by one every time an item is compared to the user's entry and no match is made. When the match is made, print it to the screen.

The Binary Search is as easy to program as the Serial Search but the routine is much faster, as it does not have to search every data item on the list. The data must be sorted in alphabetical or numerical order and the computer will look at the element at the centre of the list of data as the starting-point—figure 10—cutting the list into two halves.

If the identification keyword or number, typed-in by the user to trace an item in a file, and the element in the file does not match, the computer has to continue its search. The computer will find if the alphabetic character or number is higher or lower than the keyword typed-in by the user. If it is lower the computer will take the last element examined and make that the end of the file, cutting away the other half.

If it is higher, the computer will take the last examined item as the beginning of the new, shorter file. The computer will then find how many elements are in the short file and divide it by two. The number found will be the element at the middle of the file and that is the element looked at next by the computer.

The process continues until a match is made. Then the computer will print-out the full data item found in the file. The whole process is shown in figure 10 and a listing of the resulting program is shown in figure 11. The listing can be used alone with a short data entry routine or with the database which has been discussed as an example.

The listing in figure nine includes a short entry routine so that you can test the speed of the Binary Search. Line 30 of the routine will check to see if the keyword entered at line 25 is the same as the element pointed to by variable L in the data file, b/. If it is less than the value of the file entry, the pointer, L, will move further up the list and if it is greater it will move down the list.

That part of the program is executed in lines 30, 40 and 50. The number of the file element in L must be an integer, so INT is used in those lines. The 0.5 at the end of the lines mentioned must be added to the INTeger in L to round it up and not down, as the Sinclair machines do automatically during mathematical operations.

In some cases it is necessary to display every item in a data file which has an entry which corresponds to the keyword input by the user. That can be done by adding an extra condition, IF . . . THEN, statement before line 40 so that if a match occurs the program will continue to search the file in case there are more corresponding items. If a long file of data is to be processed and displayed it is a good idea to introduce paging into the display routine.

A list of data must not turn over the end of the screen so that a 'scroll?' message is displayed. The computer should display the data in pages and at the bottom of each page a message to the user should be displayed asking whether he wants to see more. If the user types-in 'no' the program control can be returned to the main menu.

The display of data is an important facet of any program and the guidelines apply to data output as well as data entry. When a menu is used in a program and is not displayed for some time while other operations are taking place it is important to let the user know what other options are available at that time. That can best be done by displaying a band across the bottom of the screen with the options and the 'return to main menu' option on it.

The main point I want to stress is that not all users are programmers. That may be less so now than in the past with the advent of home computing but there are still some users who have difficulty even getting around the keyboard.

While it is important to know about techniques for file processing, it is more

Figure 4

```
  10 REM ADDITIONAL POINTER ARRA
Y 'C'VARIABLE 'p' POINTS TO NEXT
 AVAILABLE   SPACE FOR DATA.
 100 DIM b$(100,20): DIM c(3): L
ET p=1
5000 REM CREATE FILE
6010 CLS : INPUT "HOW MANY DATA
ITEMS IN FILE? ";d$
6020 LET d=VAL d$
6030 CLS : INPUT "WHAT IS THE MA
XIMUM"'"LENGTH OF EACH ITEM ";e$
: LET e=VAL e$: IF e>20 THEN LET
 e=20
6040 DIM b$(d,e): LET c(2)=d: LE
T c(3)=e
6050 PRINT ''''; INVERSE 1;"NUMB
ER OF ITEMS ";d;''''" LENGTH ",e
;'''"OKAY?"
6060 INPUT f$
6070 IF f$(1 TO 1)="n" THEN GO T
O 6000
6080 RETURN
7000 REM DATA ENTRY ROUTINE
7010 IF p>d THEN GO TO 7070
7020 INPUT "ENTER ITEM ";b$: IF
b$=" STOP " THEN RETURN
7030 IF LEN b$>LEN b$(p) THEN GO
 TO 7060
7040 LET b$(p)=b$: LET p=p+1
7050 GO TO l
7060 PRINT "ENTRY TOO LONG. RE-E
NTER.": GO TO 10
7070 PRINT "CURRENT FILE FULL":
GO TO L
7080 RETURN
```

Figure 5

```
8000 REM BUBBLE SORT
8010 LET p1=1: LET p2=2: LET sc=
0
8020 IF b$(p1)(1 TO 1)>b$(p2)(1
TO 1) THEN GO SUB 8090
8030 LET p1=p2: LET p2=p2+1
8040 IF sc=0 AND p2>p THEN GO TO
 8120
8050 IF p2>p THEN GO TO 8020
8060 IF sc=0 AND p2>p THEN GO TO
 90
8070 GO TO 8030
8080 LET sc=sc+1
8090 LET h$=b$(p1): LET b$(p1)=b
$(p2). LET b$(p2)=h$
8100 RETURN
8110 CLS : PRINT "SORTED": PAUSE
 200
8120 RETURN
```

Figure 6

```
8000 REM SHELL SORT.
         MODIFICATIONS TO FIGURE
 ONE.
8010 LET p1=1: LET p2=INT (p/2):
 LET sc=0
8030 IF p2>p THEN RETURN
```

Figure 7

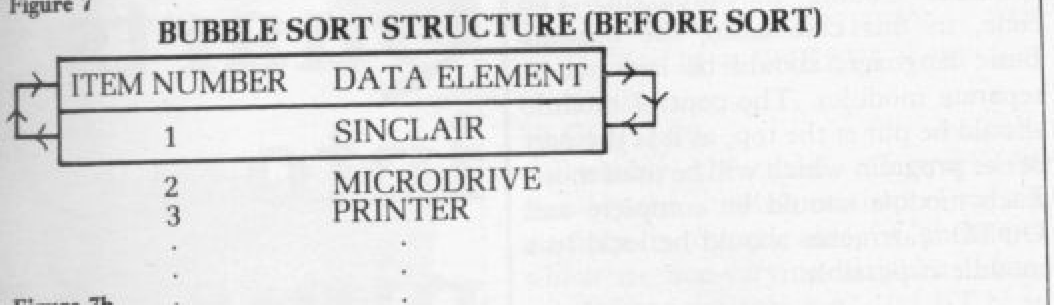
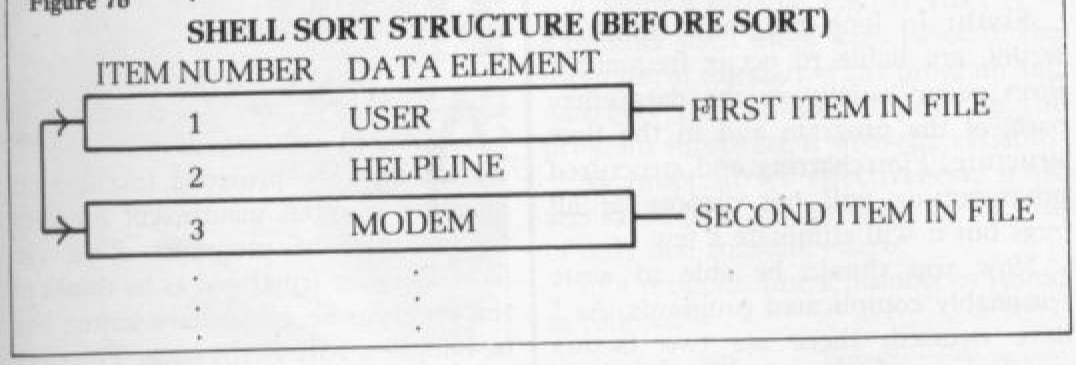


Figure 7b



important to know about the user interface, i.e., how the user will interact with the computer and how he will cope with your program. Once you can see the likely reaction of the user and fit the program around your intended user, everything else will seem simple.

Now I want to take you back to show how a program is constructed in a series of numbered steps. To begin, however, I want to discuss errors which can occur in programs, commonly called bugs, and how to rid yourself of them.

Much has been written about bugs. The term is derived from the early days of computing when only large mainframe computers existed. They had to be, and still are, kept in specially-protected areas where dust or insects could not enter the machinery. If computers were not kept clean and serviced they could often crash and data and programs could be lost.

Most of the bugs encountered by programmers occur because program structure is incorrect. The statements causing the crashes are usually the FOR . . . THEN and GOTO lines.

When a program crashes, look at the report code shown on the bottom of the screen and find the line number where the computer ceased the RUN. A GOTO effect should be limited to one routine. If an error occurs at a certain line you have to look at that routine.

On some computers the system offers a command which prints-out the line number to which the computer goes during operation. It is called the TRace ON (TRON) routine. It can be simulated on a Spectrum if you wish to add an extra instruction to each of the troublesome lines. That instruction will make the computer print-out each line number as it is executed.

It is not important that you know how it works but for those who are interested the PPC system variable is PEEKed and the current line number returned from the two addresses accessed by the statement. The instruction to be appended to program lines is PRINT PEEK 23621+256*PEEK (23621+1).

That instruction can be put anywhere any number of times. It is best to append it to the end of lines.

It is also possible to chart the course of FOR . . . NEXT loops. It is done by printing the variable used in the loop. The print statement should be put at the end of the FOR . . . statement and not at the end of the NEXT statement. If you do the latter the count printed will be incorrect. Both techniques are illustrated in diagram A—for GOTOs and—B for FOR . . . NEXT.

The two techniques can be incorporated together in a program and, with the error message you get when the program crashes you should be able to locate and correct the error.

When talking about these techniques and correcting errors, I must stress again that they will be of help only if you have used the structuring techniques which have been explained.

I have detailed the pathway to structured programming as a series of structured steps:

First: Decide what type of program you want and what you want it to do. It is best to write it as a series of statements in plain English.

Second: Decide how you are to handle the task and whether it can be done with your machine and with the level of experience you have.

Third: Break the task into a series of headings and sub-headings. For example, most programs need some form of instructive display—an entry procedure, a calculation procedure and an output procedure. They must be tailored to meet your needs.

Fourth: Draw a flowchart. That should take a long time in many cases. It is a good idea to draw several charts, expanding the most important boxes in the main one. In that way you will know

Figure 9

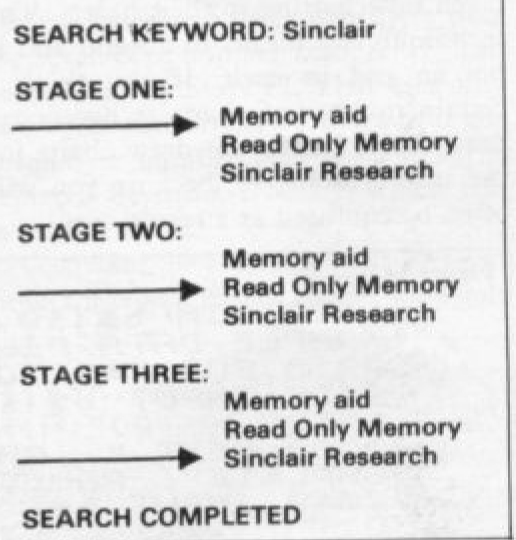


Figure 8

```
2000 DIM b$(10,12): REM SET UP E
XAMPLE DATA FILE
2010 FOR k=1 TO 10
2020 INPUT b$(k): REM DATA FILE
ENTRY
2030 NEXT k: PRINT "FILE ENTRY C
OMPLETE": PAUSE 100: CLS
2040 INPUT a$: REM SEARCH KEYWOR
D
2050 LET a=1: REM a=NUMBER OF FI
LE ELEMENT BEING MATCHED
2060 PRINT AT 10,10; FLASH 1;"SE
ARCHING"
2070 IF b$(a)(1 TO LEN a$)=a$ TH
EN GO TO 2110
2080 IF a=5 THEN GO TO 2100
2090 LET a=a+1: GO TO 2070
2100 PRINT AT 10,10; FLASH 1;a$;
" NOT FOUND": STOP
2110 CLS : PRINT AT 12,0; FLASH
1;"ITEM LOCATED"; FLASH 0,b$(a):
 STOP
```

Figure 10

SEARCH KEYWORD: Sinclair

STAGE ONE:

- Alphabet
- Bubble
- → Memory
- Research
- Sinclair
- ZX-81

STAGE TWO:

- Alphabet
- Bubble
- Memory
- → Research
- Sinclair
- ZX-81

STAGE THREE:

- Alphabet
- Bubble
- Memory
- Research
- → Sinclair
- ZX-81

SEARCH COMPLETED

the concepts of the program before you begin coding. If flowcharting does nothing else it will concentrate your mind on what you want to do.

A word of warning to peole who have taken flowcharting to their hearts. The technique is a means to an end and is not an end in itself. If you think a certain number of charts is necessary, draw them, but do not draw charts for the sake of doing so, because you will often be confused as a result.

Fifth: Coding the program. The code, in this case using the Sinclair Basic language, should be laid out in separate modules. The control module should be put at the top, as it is the part of the program which will be used most. Each module should be complete and GOTO statements should be local to a module if possible.

Sixth: In long programs, bugs, or errors, are liable to occur frequently. Bugs occur usually in the data entry parts of the program and in the flow structure. Flowcharting and structured programming will not dispose of all bugs but it will eliminate a few.

Now you should be able to write reasonably complicated programs. As I have stressed, there are two factors which are important, more so than some of the others, which are necessary to make programming easier and more enjoyable.

The first is structure. If your program is well-structured it will contain fewer bugs and will, in most cases, run faster. The second point is that the program must be designed for ease of use, so that it can be used by someone who knows nothing about computers.

If you have taken my advice, or even modified it using your own techniques, you are on the way to being an efficient programmer. You will find that you can do more and programming will become less of a difficult task and more rewarding.

Figure 11

```
2000 DIM b$(10,12): REM SET UP E
XAMPLE DATA FILE
2010 FOR k=1 TO 10
2020 INPUT b$(k): REM DATA FILE
ENTRY
2030 NEXT K: PRINT "FILE ENTRY C
OMPLETE": PAUSE 100: CLS
2040 INPUT a$: REM SEARCH KEYWOR
D
2050 LET l=INT (5/2)+0.5
2060 IF b$(l)(1 TO LEN a$)=a$ TH
EN PRINT FLASH 1;"ITEM LOCATED "
; FLASH 0;b$(l): STOP
2070 IF b$(l)(1 TO LEN a$)>a$ TH
EN LET l=INT (l/2)+0.5
2080 IF b$(l)(1 TO LEN a$)<a$ TH
EN LET l=INT ((*2)+0.5
2090 GO TO 2060
```

Diagram A

```
  10 PRINT : PRINT PEEK 23621+25
6*PEEK (23621+1)
  20 PRINT PEEK 23621+2556*PEEK
(23621+1): GO TO 10
```

Diagram B

```
10 FOR k=1 TO 10: PRINT k
20 PRINT
30 NEXT k
```

Quarts into pints

AS MANY of the thousands of ZX-81 owners will know, the 1K memory provided in the basic machine is often insufficient for even the simplest of programs. That can leave the user frustrated as he thinks of the program he could have writen had he bought a 16K RAM pack. Therefore any bytes of memory which can be saved in a program are of great importance.

There are many ways in which vital bytes can be saved on a 1K machine. Once mastered, the user will discover that his compter is very versatile.

The ZX-81 1K—1,024 bytes—of memory is filled with 124 bytes used by the machine as its system variables; x bytes for the program; y bytes for the display file—varies depending on how much is printed—and finally z bytes for the variables defined during the program.

Each line in a ZX-81 program takes five bytes as soon as it is entered—two bytes for the line number, two more for the length of line, and one byte at the end for the NEWLINE. Anything typed in the line will take up more bytes. Thus REM statements should be removed as a needless waste of memory, as they serve no useful purpose except in a machine code program.

More memory can be saved by making one line out of two. Thus:

```
10 LET G = 15
20 IF INKEY$ = "5" THEN LET
G = G - 1
30 IF INKEY$ = "8" THEN LET
G = G + 1
40 GOTO 20
```

can be replaced by

```
10 LET G = 15
20 LET G = G + (INKEY$ = "8") -
(INKEY$ = "5")
30 GOTO 20
```

The removing of one line saves the five bytes which are taken up by each line and because of that saving the program will RUN considerably faster.

The method shown, making two INKEY$ lines into one, is a type of conditional statement similar to IF. The condition inside the brackets can either

be correct or incorrect. If it is correct the result of the brackets will be 1 but if it is incorrect the result of the brackets will be 0. Therefore if the user were pressing "8", the result of the first bracket would be 1 and the second would be 0. Under those circumstances G would increase by one. An expansion of the system is shown in the following two programs:

Original

```
10 LET G = 15
20 IF INKEY$ = "5" THEN LET G = G - 1
30 IF INKEY$ = "8" THEN LET G = G + 1
40 IF G < 0 THEN LET G = 0
50 IF G > 30 THEN LET G = 30
60 GOTO 20
```

After revision

```
10 LET G = 15
20 LET G = G - (INKEY$ = "5" AND G > 0) + (INKEY$ = "8" and G < 30)
30 GOTO 20
```

Probably the most effective and possibly the simplest method of saving memory is by the use of constants which are non-varying variables. First, you define a variable, e.g., 10 LET N = a number used more than twice in the program, at the start of the program and wherever that number appears in the program substitute it with the variable.

As proof of its effectiveness, 0.9K was saved in a 16K program by the use of only one constant—zero.

In a program line a number is stored as follows:

Number as it appears in the program	Byte containing 126-(7Ehex)	Exponent Byte	Four mantissa bytes
storage of the number ten			
1.0	126	132	32.0.0.0

When using a variable the number will take up only one byte of memory. Constants will save memory only if the number is used three times or more in the program.

'The removing of one line saves five bytes'

If, however, there is a number in a program which is used only once or twice there are two more relatively efficient methods of saving memory in this.

If the number is betweeen 0 and 255, the function CODE can be used:

10 LET T = 14

can be replaced by:

10 LET T = CODE ":"

That will save four bytes of memory. This method will be effective for those numbers which have a character assigned to them but will not work for those numbers 67 to 127, as they are not used in the ZX-81 character set.

The second method is for those numbers above 255 or between 67 and 127. It makes use of the function VAL, thus:

10 LET V = 300

would become

10 LET V = VAL "300"

The use of VAL will save three bytes of memory.

Numbers above 999 which can be written easily in scientific notation should be written in scientific notation in conjunction with VAL, which will save even more memory—10,000 can become VAL "10000" which can become shortened even further to VAL "1E4" which saves another two bytes.

After all those methods have been employed there are still a few more ways of finding those extra bytes. A 0 can be replaced by NOT PI. That takes up two bytes and saves five bytes over the seven-byte number zero. A "1" can be replaced by SGN PI. That uses only two bytes and saves five bytes over the seven-byte number one. A "3" can be replaced by using INT PI, which also saves five bytes.

Galaxians is a simple 1K invaders program which makes use of the whole

```
   2 LET R=CODE "+"
   3 LET N=NOT PI
   4 LET U=VAL "2"
   5~LET W=SGN PI

   6 DIM H$(VAL "5")
  10 LET H=N
  20 LET S=N
  30 LET P=CODE "?"
  40 LET Y=R
  50 LET X=RND*R
  60 LET A$="=(9t:96:9y)"
  70 IF S>VAL "1E3" THEN LET A$=
"C(9y:97:9t)"
  80~IF S>VAL "2E3" THEN LET A$=
"W(9t:i":9y)"
  90 IF RND>VAL ".9" THEN LET A$
="COS (98:-:95)"
 100 FOR T=U TO R-W
 110 LET P=P+(INKEY$="8" AND P<2
8)-(INKEY$="5" AND P>N)
 119~CLS
 120 PRINT AT T,X;A$(U TO );AT Y
,P+W;":";AT R,P;"(99:96:9w)"
 130 IF Y>T OR ABS (P-X)>=U THEN
 GOTO VAL "140"
 131 PRINT AT T,X;"-=-";AT T,X;C
ODE A$;CHR$ N
 132 LET S=S+CODE A$
 133~GOTO CODE "C"

 140 IF Y<=T THEN LET Y=R
 150 LET Y=Y-U*(INKEY$=STR$ N OR
 Y<R)
 160 LET Y=INT (X+RND*3-W-(X>28)
+(X<W))
 170 NEXT T
 200 IF H<S THEN INPUT H$
 210~IF H<S THEN LET H=S
 220 PRINT AT R,P;"X*X";AT N,N;"
SCORE ";S;" HI ";H;" BY ";H$
 230 INPUT A$
 240 CLS
 250 IF A$="" THEN GOTO CODE "="
```

screen area. There are four types of invaders—on-screen display of base, missile and alien plus score, highest score and highest scorer's name. All those superb features are achieved by some powerful memory-saving programming.

The features to note in the LISTING are the absence of REM statements, as few numbers used as possible and the use of four constants. Conditional brackets are also put to their fullest use.

Four constants are used—R, N, U and W which are set in lines 2 to 5. They are followed by the variables H$, H, S, P, X, Y and A$ in lines 6 to 90.

The FOR-NEXT loop T is used to move the invader down the screen. That is the most effective way of doing it.

Line 110 is the INKEY$ line which controls the movement of the missile base. That makes the fullest use of conditional brackets. Note that the AND inside the brackets stops the base from moving off either side of the screen.

In line 120 everything to be displayed is printed, for three reasons—memory is saved as all the printing is done in the one line; everything is displayed speedily and that leads to little flicker after the CLS in line 119.

Line 130 tests to find whether an alien has been hit. Line 140 resets the missile when necessary. Line 150 moves the missile upwards. Line 160 moves the alien horizontally to either side at random.

Lines 200 to 250 are used at the end of the game. Note that the CLS in line 240 empties the display, thus leaving sufficient memory for the initilisation routine in lines 20 through to 90, i.e., memory is saved here by adding an extra line to the program.

To play Galaxians first RUN the program. Alien after alien will dive towards you. Use keys "5" and "8" to manoeuvre your base. Key "0" to fire. You cannot move and fire at the same time due to the basic INKEY$. As your score increases the type of invader will change. The random invader scores a bonus of 200 points.

At the end of the game if the score does not appear but instead an INPUT prompt appears that means you have set a new highest score and the computer is waiting for you to INPUT your initials of up to five characters. Once the score has appeared, press NEWLINE for another game; any other INPUT will end the game. In that eventuality GOTO 20 will start the game once more with the highest score retained.

Two into one will now go on ZX-81

Merging programs is made possible

ONE PROBLEM until now with the ZX-81 has been that when a program is loaded from tape any previous program in the computer is NEWed and lost. That means that two programs, each saved separately on tape, could not be merged into one program. That would clearly be useful for, say, writing subroutines which could be written and tested separately and then merged with a main program on a different tape. The merge feature has been included on the Spectrum and the routine we developed means that now two programs can quickly be combined into one, with few restrictions on the programs so merged, on the ZX-81.

The procedure used is outlined and explanations of the important steps are given. The basic theory is to reserve space in the 16K memory by putting RAMTOP, the address of the first byte, above the Basic system area, to a value below the normal 16K value of 32768, and then transferring the first program, byte by byte, into the reserved space.

The second program is then entered, along with a routine which will retrieve the first program, add it to the second program, and correct the values of line lengths—the third and fourth bytes in each program line, see page 171 of the ZX-81 manual—and so on, so that the program will not crash or behave abnormally. Here is the procedure:

First, the user's first program is entered from cassette tape. Restrictions on the program are that the line numbers must all be lower than 9899 but greater than the greatest line number of the user's second program, yet to be loaded. That is so that when the two programs are merged eventually, the line numbers will be in the correct order and the computer will not crash. To this program must then be added the final line:

XXXX REM END

where XXXX is the line number, which must be big enough to make it the last line of the first program but must still be less than 99. When the program is entered, the following direct commands must be entered, in the correct order:

LET X = PEEK 16396 + 256*PEEK 16397 − 1
LET Y = 49230 − X
PRINT Y

and the value of Y should be written or remembered. Then:

POKE 16388, Y − 256*INT(Y/256)
POKE 16389, INT(Y/256).

X, the value of D FILE—see chapter 27 of the ZX-81 manual—minus one, is the last byte of the program and is thus the last byte which must be transferred. Y is the location where the first byte of the program will be stored—49230 was chosen as it leaves a little spare room before the end of the computer's 16K; and RAMTOP is poked to the value—the two POKE commands—so that when NEW is executed the transferred bytes will not be erased. That can be verified by typing:

PRINT PEEK 16388 + 256*PEEK 16389

which should give the value of Y which was written.

Next the routine given in listing one, lines 9900 to 9920, is added to the first program. Then type:

GOTO 9900 followed by:

NEW as soon as the computer has finished the loop starting in line 9900.

That routine copies the bytes of the first program into the memory, starting at the location given by the variable Y. NEW is executed to clear everything except those stored bytes. To verify that the first program is still stored above RAMTOP, type:

PRINT PEEK (your value of Y)*256 + PEEK (your value of Y + 1).

That should return with the first line number of your first program.

The next stage is to load your second program from tape. The program should have all of its line numbers greater than 13 but smaller than the smallest line number in the first program, so that, again, the eventual lines when merged will be in the correct order. Once that is done the routine given in listing two—lines 1 to 13 only—should be typed-in, and then the following direct commands:

Listing 1.

```
9900 FOR F=16509 TO X
9910 POKE Y-16509+F,PEEK F
9920 NEXT F
```

LET Y= (your previous value of Y)
LET K = PEEK 16396 + 256*PEEK 16397
and after those, REM statements must be typed-in, as lines 9900 and 9901 in listing two. Each REM statement should include 100 Xs and there should be 10 such REM statements for each 1K of the first program which has already been copied above RAMTOP.

After one REM statement has been typed-in, the other can be duplicated quickly by using the EDIT facility and adding 1 to the line number each time the line is copied. In that manner the correct number of REM statements can be produced quickly.

As soon as the REM lines have been typed-in, the command GOTO 1 should be entered. The routine—lines 1 to 13—will then transfer the first program back from above RAMTOP back to the main program, and when the report code 9/13—STOP executed in line 13—is seen, the final. The merged program can be tidied by deleting lines 1 to 13 and any left-over REM lines after line 9900 may also be removed.

This is how the final 13-line routine works:

The first two lines POKE RAMTOP to its usual (16K) value of 32768. That is so that as the routine progresses, the space where the first lines were stored can be over-written once those lines have been re-transferred, and that enables longer programs to be merged.

Lines 3 to 4 check for the location of the REM command in the line:
XXXX REM END
at the end of the first program, that being the marker used to indicate the end of the first program, and the loop passes to line 5 with the variable F being the location of this REM command. The variable K which was defined previously by direct command is the location of the first byte in the first REM line—line 9900—and is the location into which the first byte of the first program is POKEd. The REM statements are used to reserve space below the Display File—we spent some time trying to POKE the system variable D FILE but with no success—and the first program is simply POKE into the locations occupied by the REM lines. Thus lines 5 to 8 POKE the first program into the space reserved by the REM lines, except for the CHR$ 118 which indicates the last character in the line:
XXXX REM END
Lines 7 and 9 are to clear the space which was occupied by the first program. Finally, lines 10 and 11 count the extra Xs left over from the last REM line POKEd into and line 12 POKEs the new length of the last line into the appropriate location. That means that the line
XXXX REM END
will finally become:
XXXX REM END (+ left-over Xs from the REM line)
e.g., XXXX REM END XXXXXXXXXXXXX
and the appearance of this is an indication that the merging is successful. Of course, if the final REM line had exactly the correct number of Xs, the line will remain
XXXX REM END

Using this routine we believe that any two Basic programs can be merged into one. Of course, that does not extend the limit of 16K imposed by having only a 16K RAM pack and it should be noted that any variables stored will be lost as the procedure is followed.

A large number of direct commands have been used where they could have been included in the two routines we have given—listing one and two—since if the procedure is to be worthwhile it should be as short as possible.

For long programs it is obviously superior to run the two routines in FAST mode but that is not very informative if for any reason the computer should crash. Therefore until you are confident that you have the procedure correct, we suggest that you take the extra time to run the routines in SLOW mode and keep an eye on the computer, even if that takes a long time.

Listing 2.

```
1 POKE 16388,0
2 POKE 16389,128
3 FOR F=Y TO 99999
4 IF PEEK F<>234 OR PEEK
(F+1)<>42 OR PEEK (F+2)<>51
OR PEEK (F+3)<>41 OR PEEK
(F+4)<>118 THEN NEXT F
5 FOR G=Y to F+3
6 POKE G+K-Y,PEEK G
7 POKE G,0
8 NEXT G
9 POKE G,0
10 FOR G=F+K-Y+4 to 99999
11 IF PEEK G<>118 THEN NEXT
G
12 POKE F+K-Y-2, PEEK
(F+K-Y-2) (F+G-K+Y-4
13 STOP
9900 REM XXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX
X
9901 REM (100 Xs)
```

10 lines per 1K stored program

Chess is in check

Chris Whittington

BROADLY, there are two main schools of thought concerning the problem of programming computers to play a game such as chess. The first and most dominant suggests that the way forward is to use the number-crunching power and speed of modern computers to plough their way through as many possible paths arising from a particular position, apply some simple evaluation to the positions arising in the path, and by that method arrive at the best move. That approach has become known as the brute force method.

The second school of thought points to the highly-successful techniques already applied by human players to the problem and suggests that the best approach would be to teach the computer to think like the best human players. We shall call this the knowledge method.

Faced with the problem of producing an effective chess-playing program for a home computer such as the Spectrum, we can probably discard the knowledge method, if only because we still have no real idea how it is that human grand masters decide on their next best move. Indeed, what do we mean by best move? There is no real choice at present other than to use the brute-force method for solving the problem.

Before we begin to explore a possible solution it would be best to examine in some detail the nature of the problem. Chess is a two-player game.

Whatever is good for one side is bad for the other and vice versa; such games are known as 'zero-sum games'. That will allow us to make statements such as this move gives White a score of +250 and thus gives Black a score of −250.

The game is played on a board of fixed size with a fixed number of pieces; therefore we should be able to represent the board and pieces in some way in the computer memory.

The laws of chess define how the pieces move, whether any position is illegal—for example, leaving one's king in check—and how to decide on wins, losses and draws. In principle, therefore, we should be able to define and encode all the possible ways of moving a chess piece, detecting checks and so on.

Unfortunately, from our point of view, a number of different chess pieces each move in a different manner and some of them move in some different manner in special circumstances—for example castling, *en passant.*

I suspect that the apparently daunting task of defining all those possible moves for each of the pieces and encoding them into assembly language deters many a budding chess programmer from beginning the task.

Some of the early chess programs, and even some one can still buy, avoided the problem by not implementing the difficult portions, with the result that their program just never did castle or capture *en passant.*

The problem is not as daunting as it seems; one can work out routines to do the task and there are published algorithms, and even listings, which can remove some of the tedium.

Since we have to decide on a best move we will need some way to evaluate, or attach a number to, any move. The conventional wisdom is to keep that position evaluator as simple as possible, usually measuring the balance of material and a few positional factors. It will be of interest to note that the positional evaluation will be the only part of the program where chess knowledge and skill, as apart from a knowledge of the rules of the game, will be required.

There has been much controversy between strong chess players and computer programmers as to whether poor chess players can produce good chess programs.

> 'We must concentrate on refining our search through the tree of moves'

Certainly the best programs usually have been prepared with the help of strong players and my feeling is that substantial chess knowledge is of most use after the program has been completed and is being play-tested to tweak the program to its optimal performance by adjusting the evaluation function.

Then we have the problem of how to plough our way through all the moves to arrive at our brute-force solution. In chess programmers' parlance that is known as the 'exponential explosion problem' and it goes something like this:

On the Spectrum we can evaluate a position in, say, 25 milliseconds. In any one chess position there will be around 30 legal moves to consider. Thus to look one half-move deep we need to evaluate some 30 positions to arrive at the best move. That takes 30 times 25 milliseconds, or 0.75 seconds.

To search two-ply deep we need to look at 30 positions and all the 30 positions arising from each of those—30 times 30 positions in all will take 900 times 25 milliseconds or 18 seconds.

By that reckoning to search three-ply takes nine minutes, four-ply takes 4.5 hours, five-ply takes almost a week, and so on.

To achieve good results against strong players, a program will need to search between eight- to 10-ply moves deep. According to the previous calculations an eight-ply search would take 400 years to complete. Such a game, starting in pre-history might now be nearing completion.

If we are to get anywhere using brute-force methods we must concentrate on refining our search through the tree of moves to reduce or eliminate the blow-up effect. As an aside, a game such as draughts, which has far fewer possible moves arising from any one position, does not generate such severe problems and thus machine-coded draughts programs are more likely to defeat strong players than similarly-encoded chess programs.

Fortunately there are several methods to refine the tree search. The programs available on the Spectrum and ZX-81 have reduced the multiplication factor for each ply from about 30 times to around four to eight times.

Adopting a top-down approach to the problem, we can formulate our first statement of the approach we shall be taking.

We shall use the method of brute force with refinements, evaluating each node—position—in the tree of moves according to a simple and fast evaluation function, biased heavily by material factors.

The program will need to be as fast as possible and must therefore be encoded into machine code. Basic would be far too slow.

We shall require adequate tools for this task. As a minimum we shall need a powerful editor to work on the assembly language files. Since those files may be as large as 40K bytes—100K bytes when properly documented—we shall need a disc-based system to hold them.

We shall need an assembler program to turn the assembly language files into machine code and a monitor program to help remove bugs.

Only after the program is running without bugs shall we consider putting it on to our small home computer system.

Our next task will be to break down the problem further and define particular areas of the program to which we must attend.

Main modules—tree-search algorithm; move generator; is king in check? detector; position evaluator.

Database—how to represent the chessboard; how to store the moves and positions as we move up and down the tree; tables containing knowledge about the position which can be used by the position evaluator.

REFERENCES

Chess skill in man and machine. Editor, Peter Frey. Publisher, Springer-Verglag.

Sargon, a computer chess program. Authors, Dan and Kathe Spracklen. Publisher, Hayden.

The machine plays chess? Author, Alex Bell. Publisher, Pergammon.

Think like a grand master. Author, Alexander Kotov. Publisher, Batsford.

Putting brains into monsters

Donald Hughes writes adventures

HOW MANY TIMES have you wished you could write adventure programs? How many times have you started, only to cease an hour later, frustrated by the awful complexity? If the answer to the second question is too many times, you are probably approaching the problem from the wrong end.

An adventure game should be split into two parts—the brain and the adventure.

The brain is the interactive part of the program which communicates with the adventurer, answers queries, picks up items and drops them. The adventure can then be rendered into data suitable for the brain.

The brain is a loop, normally large, which uses a READ statement to take data for the adventure from successive DATA statements. Normally the brain program will run its course once per adventure location, so when a location is left, it draws new data from the adventure. Using such a brain, each adventure location can be stored in one DATA statement of set format.

You must start by deciding what initial intelligence the brain will have. I recommend you do not aim for anything more intelligent than the simple example. The example brain understands only one word at a time and they must be keywords. If the command given is not a keyword, the brain is so unreceptive that it proceeds merrily along as if it had found a keyword. It works, though, and you can write adventures, using the brain, in two hours.

> 'Do not expect your first brain to run adventures of a professional standard'

Starting with a brain similar to the example, you can add keywords slowly until it is reasonably intelligent, and you can have complex adventures, but start simply, or you will find yourself trying to debug a program which does not work far enough for you to find the bugs, let alone see what is not correct; and it will be back to the frustrations.

Do not expect your first brain, or the example, to run adventures of a professional standard; that is not the aim—it is to let anyone starting on a Spectrum have fun with programs they thought were only in the realm of professionals.

A simple brain operates in this way. The text of the first location is read in; the number of items in a room is read in; the items in the room are read into a string array, then printed-out along with a copy of the text; the monster in the room is read in, along with its strength. Logically, before the player may take an item he must defeat the monster. Once the monster is dead, the brain asks what function is required. The functions understood are:

Take—to pick up an item.
Drop—to drop an item
Inve—produces an inventory.
Leave—leaves the room.

Those instructions, combined with an imaginative adventure, are sufficient to entertain.

SCORE SHEET

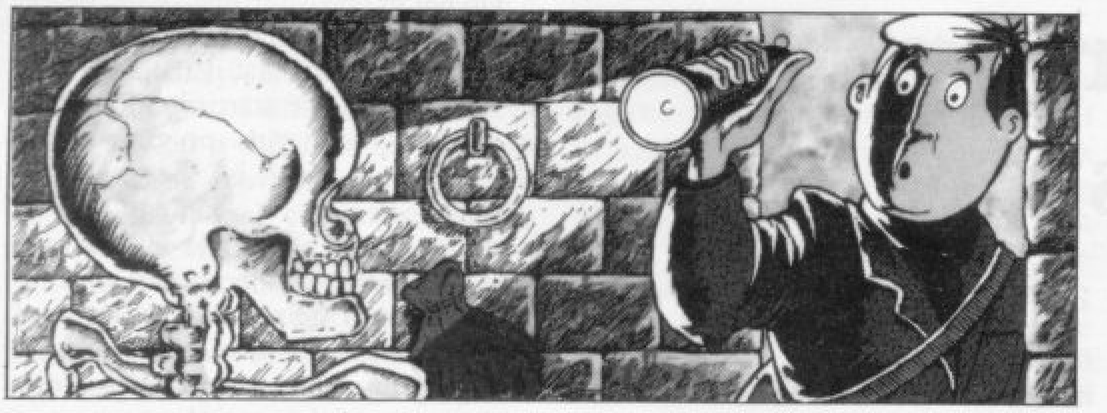
Due to the limitations of the brain, here is how to score your performance in this adventure:

If you cheated with the priests, 0; if you took the heavenly being, 0; if you died, 0.

You should have with you a sword, bag of coins, box of coins, ruby and Spectrum—the most valuable item.

If you have all five, excellent; only four, good; only three, fair; only two, bad; only one—have you not played an adventure game previously?

```
  10 LET s=350
  15 DIM y$(10,10)
  20 LET N=0
  95  REM basic first view
  97 CLS
 100 READ t$
 105 IF t$="end" THEN GO TO 9900
 110 PRINT t$
 120 PRINT "IN THIS ROOM THERE I
S"
 130 READ rc
 135 DIM a$(rc,40)
 140 FOR q=1 TO rc
 150 READ a$(q)
 160 PRINT a$(q)
 170 NEXT q
 180 REM monsters
 190 READ m$,m
 195 IF m$="none" THEN GO TO 265
 200 PRINT "YOU SEE A ";m$;".","'"
DO YOU WISH TO FIGHT/RUN (F/R)"
 210 INPUT w$
 220>IF w$="f" THEN GO TO 450
 230 IF w$="r" THEN PRINT "BY WH
ICH EXIT DO YOU WISH TO MAKE YOU
R ESCAPE"
 240 INPUT e$
 250 GO TO 430
 260 REM return fromfight
 270 PRINT "YOU MAY NOW TAKE OR
DROP AN ITEM OR YOU MAY LEAVE"
 280 INPUT c$
 290 IF c$="take" THEN GO TO 570
 300 IF c$="drop" THEN GO TO 620
 315 IF c$="leave" THEN GO TO 41
0
 320 IF c$="inve" THEN GO TO 820
 330 PRINT "USE COMMANDS TAKE ,D
ROP,INVE,LEAVE"
 340 GO TO 280
 400 REM
 410 PRINT "IN WHICH DIRECTION"
 420 INPUT c$
 430 GO TO 9000
 440>STOP
 450 REM fighting
 460 PRINT "THE ";m$;" HAS ";m;"
 STRENGTH PTS,DO YOU STILL WISH
TO FIGHT"
 470 PRINT "(Y/N)"
 480 INPUT f$
 490 IF f$<>"y" THEN PRINT "BY W
HICH EXIT DO YOU WISH TO MAKE YO
UR COWARDLY ESCAPE"
 495 IF f$="y" THEN GO TO 510
 500 GO TO 240
 510 LET s=s-(m*(RND+1))
 520 IF s<=0 THEN PRINT "HA HA ,
YOU ARE DEAD,BYE"
 525 IF s>0 THEN GO TO 540
 530 GO TO 440
 540 PRINT "THE ";m$;" IS DEAD"
 550 GO TO 260
 560 REM taking
 570 PRINT "WHAT DO YOU WISH TO
TAKE"
 575>LET N=N+1
 580 INPUT Y$(n)
 582 IF N>=5 THEN PRINT "YOU MUS
T DROP AN ITEM BEFORE YOU MAY TA
KE AGAIN"
 600 PRINT "YOU MAY NOW CONTINUE
"
 610 GO TO 270
 620 REM dropping
 630 PRINT "WHAT DO YOU WISH TO
DROP"
 635 DIM d$(10)
 640 INPUT D$
 650 FOR C=1 TO N
 660 IF d$=y$(c) THEN GO TO 710
 670 NEXT c
 680 PRINT "YOU DO NOT OWN SUCH
AN ITEM"
 690 PRINT "TRY AGAIN"
 700 GO TO 630
 710 FOR z=c TO n
 720 LET y$(z)=y$(z+1)
 730 NEXT z
 735 LET n=n-1
 740 PRINT "O.K."
 750 PRINT "DO YOU WISH TO TAKE
AN ITEM (y/n)"
 760 INPUT C$
 770 IF c$<>"y" THEN GO TO 600
 780 PRINT "WHAT ?"
 790 LET n=n+1
 800 INPUT y$(n)
 810 GO TO 600
```



The brain then asks a question, depending on the keyword used. For example, What do you want to take? Type-in, e.g. "sword". What do you wish to drop? Type-in, e.g., "coin". If INVE was used, the brain prints-out an inventory.

When LEAVE is typed-in, the brain asks in which direction. That is merely for show as, when a direction is typed-in, it PRINTS "you are going to the next room", PAUSES for a second; CLEARS the screen; and RESTARTS THE LOOP, reading in the next portion of DATA.

One of the first improvements to the brain you should make is a way of flipping through the DATA to find any adventure location, so one can go N, S, E, W. The example brain suffices without that.

In writing adventures, first examine the listing. The brain is obvious, lines 95 to 850. Remove the data statements, 1000 to 8000. Insert your own data, in this form:

1000 DATA "A", "X", "B", "B", "B", "M", S

A is text describing the room; X is the number of items in the room and must be equal to the number of strings following it; B is a string, an item in the room; M is the name of the monster; S is the strength of the monster. Should you not want a monster, type-in the DATA positions for M and S "none", 0

'A simple brain operates by reading in the text of the first location; the number of items in a room is read in; the items in the room are read into a string array; the monster in the room is read in, along with its strength'

Try typing-in the adventure detailed into your Spectrum and playing it to see the possibilities, within the limitations of even a simple brain. You should have plenty of fun. After that, write your adventure for it. Real beginners should be very careful when typing-in the adventure; a bug will make itself immediately and annoyingly apparent.

To use the brain on a ZX-81, you will have to circumvent the "who needs READ statements" problem.

Combat works this way; your strength is S; monster strength is M. In combat, this operation takes place

LET S=S-(M*(RND+1))

which means a random figure varying between M and nearly M*2 is removed from your strength in defeating the monster. For different adventures, different Ss can be given, so only by choosing carefully what to fight can you win.

List of variables used in the program.

T$	=text
rc	=number of items in a room
A$	=array of room contents
M$	=monster name
M	=monster strength
W$	=answer to fight/run option
C$	=command word
C	=used in loops
Z	=used in loops
S	=your strength
F$	=answer in fight routine
Y$	=answer in take routine
d$	=answer in drop routine
e$	=exit direction

```
  10 LET s=350
  15 DIM y$(10,10)
  20 LET N=0
  95  REM basic first view
  97 CLS
 100 READ t$
 105 IF t$="end" THEN GO TO 9900
 110 PRINT t$
 120 PRINT "IN THIS ROOM THERE I
S"
 130 READ rc
 135 DIM a$(rc,40)
 140 FOR q=1 TO rc
 150 READ a$(q)
 160 PRINT a$(q)
 170 NEXT q
 180 REM monsters
 190 READ m$,m
 195 IF m$="none" THEN GO TO 265
 200 PRINT "YOU SEE A ";m$;".","
DO YOU WISH TO FIGHT/RUN (F/R)"
 210 INPUT w$
 220>IF w$="f" THEN GO TO 450
 230 IF w$="r" THEN PRINT "BY WH
ICH EXIT DO YOU WISH TO MAKE YOU
R ESCAPE"
 240 INPUT e$
 250 GO TO 430
 260 REM return fromfight
 270 PRINT "YOU MAY NOW TAKE OR
DROP AN ITEM OR YOU MAY LEAVE"
 280 INPUT c$
 290 IF c$="take" THEN GO TO 570
 300 IF c$="drop" THEN GO TO 620
 315 IF c$="leave" THEN GO TO 41
0
 320 IF c$="inve" THEN GO TO 820
 330 PRINT "USE COMMANDS TAKE ,D
ROP,INVE,LEAVE"
 340 GO TO 280
 400 REM
 410 PRINT "IN WHICH DIRECTION"
 420 INPUT c$
 430 GO TO 9000
 440>STOP
 450 REM fighting
 460 PRINT "THE ";m$;" HAS ";m;"
 STRENGTH PTS,DO YOU STILL WISH
TO FIGHT"
 470 PRINT "(Y/N)"
 480 INPUT f$
 490 IF f$<>"y" THEN PRINT "BY W
HICH EXIT DO YOU WISH TO MAKE YO
UR COWARDLY ESCAPE"
 495 IF f$="y" THEN GO TO 510
 500 GO TO 240
 510 LET s=s-(m*(RND+1))
 520 IF s<=0 THEN PRINT "HA HA ,
YOU ARE DEAD,BYE"
 525 IF s>0 THEN GO TO 540
 530 GO TO 440
 540 PRINT "THE ";m$;" IS DEAD"
 550 GO TO 260
 560 REM taking
 570 PRINT "WHAT DO YOU WISH TO
TAKE"
 575>LET N=N+1
 580 INPUT Y$(n)
 582 IF N>=5 THEN PRINT "YOU MUS
T DROP AN ITEM BEFORE YOU MAY TA
KE AGAIN"
 600 PRINT "YOU MAY NOW CONTINUE
"
 610 GO TO 270
 620 REM dropping
 630 PRINT "WHAT DO YOU WISH TO
DROP"
 635 DIM d$(10)
 640 INPUT D$
 650 FOR C=1 TO N
 660 IF d$=y$(c) THEN GO TO 710
 670 NEXT c
 680 PRINT "YOU DO NOT OWN SUCH
AN ITEM"
 690 PRINT "TRY AGAIN"
 700 GO TO 630
 710 FOR z=c TO n
 720 LET y$(z)=y$(z+1)
 730 NEXT z
 735 LET n=n-1
 740 PRINT "O.K."
 750 PRINT "DO YOU WISH TO TAKE
AN ITEM (y/n)"
 760 INPUT C$
 770 IF c$<>"y" THEN GO TO 600
 780 PRINT "WHAT ?"
 790 LET n=n+1
 800 INPUT y$(n)
 810 GO TO 600
```

Sorting through memory for some useful addresses

Dilwyn Jones explains the workings of systems variables

SYSTEM VARIABLES are the bytes in memory from address 16384 to address 16508 in RAM on the ZX-81. They are used by the computer to remember certain things about its workings, such as where to print next.

You can make use of some of them in your programs either by reading their value—PEEKing—or replacing them with new values—POKEing—so as to use the information they contain or make the computer do something it might not otherwise do.

Not all of them can be used in this way; some may ignore you, whereas changing the contents of some of them may cause strange effects, like making a mess of the screen display. At worst, a little nasty known as a crash may be caused.

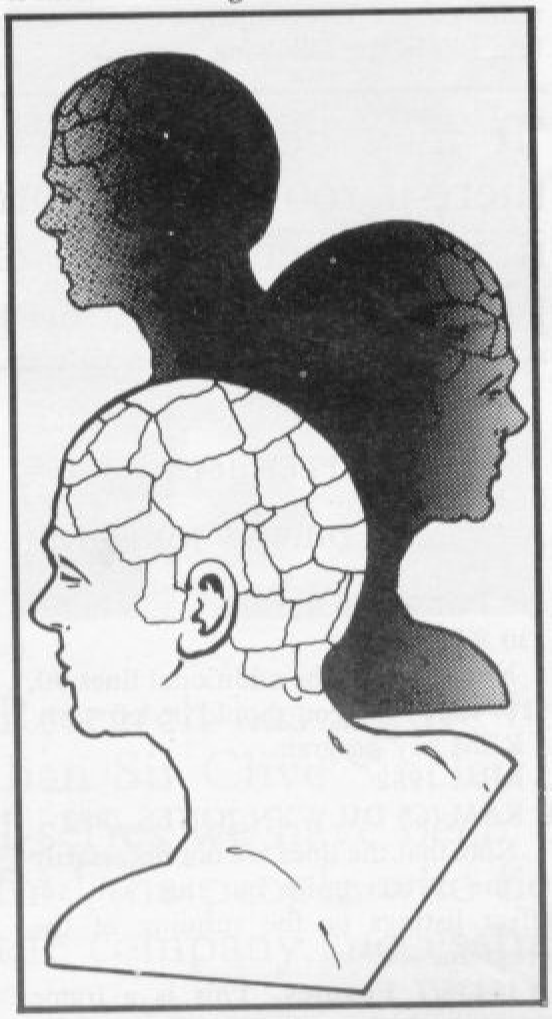
● **16384 ERR-NR (Error report number).** The value contained in address 16384 determines the report code. If you POKE a number into 16384 which is anything other than 255, the program will stop and display an error code, which may be non-standard but meaningful in some way.

For example, if you wanted to arrange that if the user entered an incorrect value the program stopped with error U—standing for USER ERROR—you would arrange that POKE 16384,29 was executed. To determine which value to POKE, remember that 16384 has a value of 1 less than the report code.

The error report code can be suppressed by POKEing certain values into this system variable. That may be useful at exhibitions, where a report code may be an unsightly distraction from the rest of a display. Experiment with POKEing some of these numbers into 16384: 43, 70, 72, 73, 74, 75, 76, 77, 79 81, 82, 89.

● **16386/7 ERR-SP.** This system variable contains the address of the first item on the machine stack after the GOSUB returns. With PEEKing this two-byte system variable you can check how many GOSUB return addresses are present on the stack, for example to check if any bug in your program had been causing it repeatedly to call and jump out of subroutines. Here is how to check:
PRINT (PEEK 16388 + 256 × PEEK 16389)—(PEEK 16386 + 256 × PEEK 16387)

● **16388/9 RAMTOP.** RAMTOP is the address of the first byte of memory above that used by Basic. NEW operates only this far, so anything placed above RAMTOP is safe from every Basic function except POKE—that is how you would put it there. That makes it ideal for storing machine code or data



you want to pass between two programs, so that it is not lost when you execute normally-destructive commands like LOAD or NEW. On a 1K ZX-81, 16388/9 have the values 16388 0 16389 68.

Using the formula from chapter 28 of the ZX-81 manual, 0 + 256 × 68 is 17408. That is the normal address of RAMTOP in a 1K ZX-81. If you have a RAM pack plugged-in and wish to see if a program you have would fit into 1K, it may be tested by POKEing the foregoing values into RAMTOP, then entering the command PRINT USR 1040. The machine will behave like a 1K ZX-81.

The value contained in RAMTOP also determines how the display file—screen picture—is made up after CLS. If the value in RAMTOP is less than 19712—16388 = 0, 16389 = 77—the display file is contracted to minimum size consisting of only 25 NEWLINE characters. If RAMTOP is 19712 or higher, the display file is expanded by filling with spaces. A contracted display file has the advantage that it takes five seconds less to LOAD or SAVE programs.

● **16391/2 PPC.** Contains the line number of the statement being executed. It could be used as an aid to debugging a program which has computed GOTOs/GOSUBs all over the place; a few PRINT PEEK 16391 + 256 × PEEK 16392 statements here and there could determine whether or not the program went where you thought it should do.

In the last line of a program it determines the line number to be printed by the report, e.g., 0/100. You may like to use that to print a score on the screen at the end of a game.

● **16396/7 D-FILE.** Contains the address of the start of the display file. The character pointed to is the first NEWLINE character in the display file. Since the display file floats above the program in memory, you can use it to tell you where the program ends, giving you an indication of the length of the Basic program, since the Basic program starts at 16509: PRINT PEEK 16396 + 256 × PEEK 16397—16509 will tell you the length of the program in bytes.

If you want to PEEK/POKE into the display file for any reason, this system variable helps you by telling you where in memory it starts.

● **16398/9 DF-CC.** Tells you where in memory the current PRINT position lies. PEEK 16398 + 256 × PEEK 16399 gives the address in RAM of the current PRINT position. That could be POKEd to change the PRINT position. Alternatively, if you PEEKed the address of the PRINT position, you would obtain the CODE of the character already at that position—useful for detecting collisions and so on in games, or for programs which require a screen cursor to be highlighted in inverse video such as word processors:

LET AS = CHRS (PEEK (PEEK 16398 + 256 × PEEK 16399))

IF AS > "A" AND AS < = "Z" then print at Y, X; CHRS (CODE AS + 128)

The statement PRINT AT Y,X; moves the cursor without printing.

● 16400/1 VARS. This pair of system variables enables you to find the address of the start of the variables area if you want to go PEEKing or POKEing around, or the end of the display file if you want to work backwards to POKE characters on to the bottom two lines of the display which cannot normally be PRINTed upon.

● 16404/5 E-LINE. Contains the address of the end of the variables area. We can examine it to give a rough idea of how much memory we have used, including system variables, program, display and variables:

PRINT PEEK 16404 + 256 × PEEK 16405—16384

● 16412/3 STKEND. Contains the address of the top end of the calculator, immediately below spare memory. Used in conjunction with ERR-SP 16386/7, we can obtain an approximate idea of how much memory we have left in which to work. PRINT (PEEK 16386—PEEK 16412) + 256 × (PEEK 16387—PEEK 16413) The figure is in bytes.

16417 not used. This system variable is not used but is available to the user; you could use it to store information in the form of an integer from 0 to 255. That would be saved on tape when the program is saved.

● 16418 DF-SZ. Define screen size, or the number of lines in the lower part of the screen. If you POKE a value of 1 or 0 into this system variable you can use lines 22 and 23, so that PRINT AT 22,0; and PRINT AT 23,0; become acceptable statements. If using INPUT or SCROLL, you should restore the original value, normally 2, or you may cause a crash. Conversely, if you are short of memory and using a SCROLLing display, you can make scrolling start from further up the screen by POKEing a value greater than 2 into 16318, a Basic part screen scroll.

● 16419/20 S-TOP. This contains the number of the top line in automatic listings. Automatic listings are those produced when you press NEWLINE. It can be annoying when you are trying to work on one part of a listing and the computer insists on displaying a different part.

To place any line number you want, say line X, at the top of auto listings you must first move the cursor to a line number greater than the one you want at the top. Then enter the commands:

POKE 16419,X—INT(X/256) × 256
POKE 16420,INT(X/256)

● 16425/6 NXTLIN. The address of the start of the next program line. You could use it to run machine code in a REM statement anywhere in the program, e.g.:

100 LET A = USR (PEEK 16425 + 256 × PEEK 16426 + 5)
101 REM . . . machine code . . .

Or you could use NXTLIN to security-lock lines into programs to point out that, for example, you hold copyright to a program. It should not be possible to edit out those lines easily. The easiest way of doing so is to change the line number to 0, which cannot easily be removed. As an example, we will lock line 100 in the following program:

'It may be necessary to determine whether there is room for a word on the current line or if it is necessary to move to a new line to prevent the word being chopped in two'

```
1 REM any program
2 REM 1982
90 LET A = PEEK 16425 + 256 ×
PEEK 16426
100 REM (C) DILWYN JONES 1982
110 POKE A,0
120 POKE A + 1,0
130 STOP
```

Now edit out the additional lines 90, 110, 120, 130. You should be left with:

```
1 REM any program
2 REM 1982
0 RAM (C) DILWYN JONES 1982
```

Note that the lines are not necessarily in the correct order but that will not affect listings or the running of the program.

● 16436/7 Frames. This is a frame counter which counts the frames of a picture sent to a TV set. It is incremented 50 times a second and can be used for timing with a range of about 11 minutes before repeating. To set the timer initially we use:

POKE 16437,255
POKE 16436,255

They may be entered as direct commands or used as program statements, although of more use within programs because of the limited timing range. The values of the frame counter start at 65535 and count down to 32768, because bit 15 is normally 1. Once it has been re-set, its value is read like this to give a value in seconds: LET TIME = (65535—PEEK 16436—256*PEEK 16437)/50

The variable TIME then contains the time elapsed in seconds since the frame counter was re-set.

Remember that PAUSE uses the frame counter, so you cannot be timing and use PAUSE, too. If you want a delay while using the frame counter for timing, use a FOR/NEXT loop of about 1 to 60 for every second of delay. Remember also that bit 15 should always be 1 when timing. If both bytes of the frame counter reach zero, the program will crash.

● 16441/2 S-POSN. After you use PRINT at Y,X; where Y and X are print co-ordinates:

PEEK 16441 would be 33−X
PEEK 16442 would be 24−Y

16441 contains information as to the PRINT column number but it is not very easy to use. If X is 0—the PRINT position is somewhere on the left-hand side of the screen—the value of 16441 starts at 33 and decrements by 1 for every column across the screen. The value of 16442 starts at 24 if the PRINT position is at the top of the screen and decrements by one for every line moved down the screen; 16441 may be used in programs which handle text.

It may be necessary to determine whether there is room for a word on the current line or if it is necessary to move to a new line to prevent the word being chopped in two. Suppose the word to be printed was A$. To prevent A$ being chopped in half you could use:

IF PEEK 16441 < LEN A$ + 1 THEN PRINT

always assuming, of course, that the previous PRINT item ended in a semi-colon or comma. It may help to think of 16441 as the number of characters + 1 which can still be printed on this line of the screen.

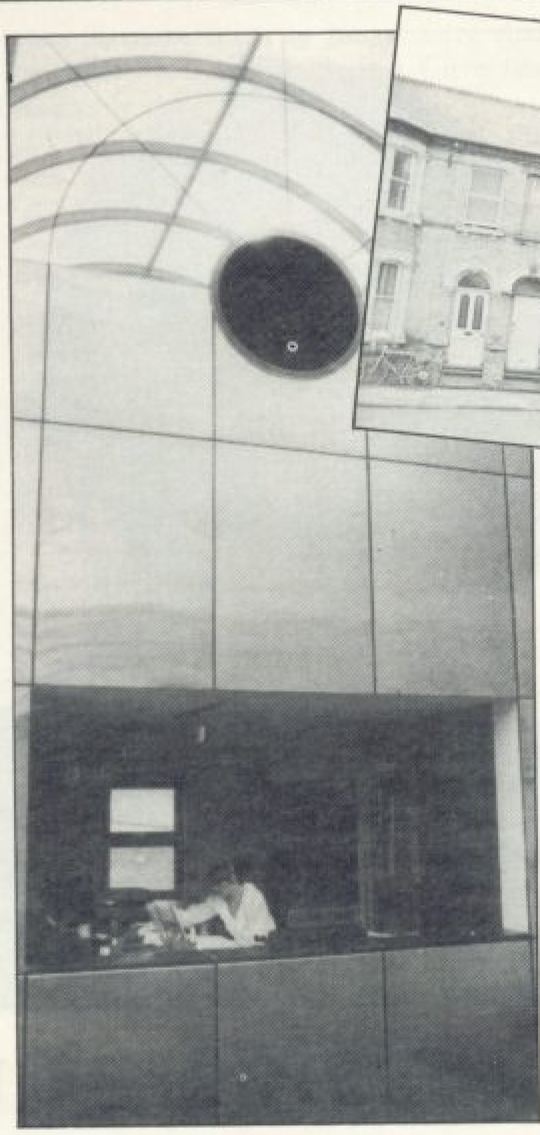
● 16444 to 16476 printer buffer. If the printer is not used, may be used to store information if you have nowhere else to put it.

● 16507/8 not used. These two unused system variables can be used by the programmer to store integers if needed. They are saved on tape along with the program.

Inside Sinclair

Sinclair Research has always been much more than Sir Clive Sinclair. In our occasional series we have been behind the scenes to find the people who help to make the company successful.

Unlock Your Imagination
...with SCOPE Computer Graphics Language and write arcade-style game programs to compete with the professionals.
THIS sensational SCOPE utility and your Spectrum will unlock your imagination. It's amazingly simple and fast and enables you to write fast-moving games in MACHINE CODE.
SCOPE realises your potential and releases it. You'll amaze family and friends and surprise the professionals
They said it couldn't be done.
TECHNICAL DETAILS
SCOPE is a fully structured multitask compiled language, specifically for writing arcade-type games enabling you to program machine code without any knowledge of machine code.
SCOPE can be used as an assembler with SCOPE words as plain language mnemonics, therefore no knowledge of machine code is required.
Each command word will write into the program memory the relevant machine code.
It is generated from BASIC and is fully compatible with BASIC.
The language from the user end is very easy to comprehend and is comprehensive, providing many features not available from BASIC. SCOPE has to be used to be believed.
☐ As fast as MACHINE CODE
☐ Simpler than BASIC
☐ Instruction Manual
☐ £11·95
48K SPECTRUM
SCOPE
Computer Graphics Language
Unlock Your Imagination
SCOPE
CLIP THE COUPON NOW AND UNLOCK YOUR IMAGINATION
Post to: ISP Marketing Ltd, Crown House, 38b High Street, Godalming, Surrey
Tel: Godalming (04868) 24151
Please send me ... copies of the SCOPE computer graphics language tape and instruction manual at £11.95 (p&p free). (Cheques payable to ISP Marketing Ltd)
Block capitals please
Name
Address
Dealer Enquiries Welcome
A



PEEPING from behind rows of terraced houses near the centre of Cambridge is the Sinclair Research ultra-modern new computer centre.

The architect, Cristoph Grillet, pictured left, of Lyster, Grillet and Harding, is said to have combined Cambridge traditions with major innovations in architectural design, interior furnishing and environmental control systems.

One of the ducts for the system can be seen in the picture on the immediate left above the office which overlooks the entrance hall. The hall is dominated by the sculpture, shown far left, Double Torso II by Helaine Blumenfeld.

Stainless Sinclair

Former school chum now looks after the money

In the management changes at Sinclair Research last year, Bill Matthews became financial director. He talks to Claudia Cooke

BILL MATTHEWS sits in his office and smiles at the recollection of his old school chum, Sir Clive Sinclair, tinkering with radios while the other boys listened to them.

"He was innovative even at that age, you know. He was always tinkering with something and he made his first matchbox-sized radio while we were still at school.

"We lived very near each other in Guildford and I remember going to his house for enthusiastic demonstrations of his latest plans. I think he was very much an individual character and did not fit into the standard schoolboy image at all".

A great deal has happened to both men since those early days at St George's School, Weybridge, Surrey. Now, once again, they are together, Bill Matthews having joined Sinclair Research as finance director last September.

"I was working in Toronto last year and Clive had to go there for a Mensa conference. We had lunch and he asked me if I was interested in this job. I thought what the hell, why not? Here I am, thoroughly enjoying it".

The two men, in fact, worked together previously, but in the very early days of Sinclair Radionics. "I think I kept his cash book or something when he was busy packing-up and selling little bits by mail order. It seems a long time ago".

Matthews, now 43 and married with four children, went straight from school to read economics and business administration at Sheffield University. He followed that with training at a firm of accountants in London, qualifying as a chartered accountant in 1963.

The first five years of his career were spent in public accounting in Turkey and Italy. "It was great to be able to travel and, having been released from all that studying, I did a reasonable amount of work but certainly nothing you could call strenuous".

Then followed marriage and the offer of a job with ICI as a group accountant. He spent four years there and still considers it one of the best-run companies for which he has worked.

He moved on to become financial controller for Sony U.K. for two years

'He asked me if I was interested in this job. I thought what the hell, why not?

and was involved in setting-up the company's successful television factory in Wales. Emigration to Canada was the next step, working for management consultants Price Waterhouse. His most recent job, in Houston, Texas, was as executive vice-president of a company with clients from the oil and gas industries.

A varied career but one which has always involved him in the use of computers, particularly for management information. The finance department at Sinclair Research is now in the process of computerising its system with an ICL 25. Matthews expects it to be of considerable help in financial reporting and control.

One of his first tasks at Sinclair was to place 10 percent of the company shares with institutional investors, an offer which was well-subscribed. He says:

"We see ourselves getting a Stock Exchange quotation in two years. It is still our intention to keep the company small in terms of the number of people employed; that concept has not disappeared.

"Sales have grown substantially and we have reached a size where we could issue shares and get a public quotation. It will be our coming of age, if you like".

Matthews sees one of his tasks as ensuring that Sinclair has sufficient resources to continue to grow rapidly and

'We have to be careful because the life expectancy of our product is fairly short'

ensuring that those resources are generated from within the company — from its profits. No easy task, he admits.

"Coming from a financial background, you find all companies have a great deal in common. The exceptional thing here is the sheer speed with which we are growing and the dynamism of our research activities.

"It can be a problem to find the resources to cover all that development. It creates its own problems and puts exceptional pressure on everyone who works here but it also creates a much more optimistic attitude among the staff. We tend to pay better than the normal market rates to get good people and it is a marvellous team".

Matthews says the company philosophy will always be to design products which are inexpensive and produced in large volume, thus making them widely-available.

"I think the Sinclair computers are marvellous little machines. We are just seeing the beginning of a revolution in the use of computers. The first stage was to make them so inexpensive that anyone could use them and Clive has certainly done that.

"We have to be careful because the life expectancy of our product is fairly short. You can be fairly sure other firms will catch-up rapidly. We have to make the best use of that technical lead and exploit it fully, to the best possible advantage. We cannot afford to sit still. In three years I do not suppose the ZX-81 will be marketed. We are not making pyjamas for Marks and Spencer after all; that kind of thing you can probably go on selling for about 200 years.

"We must have a stream of new and innovative products all the time and we must remain inventive".

Listening to those exacting standards is tiring but they are the standards which have taken Sinclair to the top so quickly and Matthews has absorbed them with equal speed. Approximately half the staff of Sinclair Research is involved in research and development and Matthews intends to concentrate substantial resources for its continuation, buying new premises and rationalising existing ones.

With Sinclair products tending to be market firsts, Matthews admits there is often a tendency to under-estimate the product. "When sales are restricted they are restricted by the ability to produce rather than lack of demand for the product but one has to be cautious to a certain extent," he says.

"You are making a commitment to the subcontractor, so you have to be careful not to over-estimate the demand. Then the subcontractor has to buy components and perhaps hire people. So you have to be as realistic as you possibly can.

"Producing new concepts means a greater risk element and greater flexibility in financial control. You have to keep re-assessing the situation in financial terms and try to predict what will happen".

Because the Sinclair market is largely domestic it is by nature more volatile than selling to big industry. This is another factor Matthews has to take into consideration. With all that on his mind, Matthews, who displays a Casio calculator on his desk, has not yet had much free time to experiment with the Sinclair products.

His eldest 15-year-old son has compensated for that. He has two Spectrums at home and, his father says, spends most of his time sitting solidly in front of them, lost to the world.

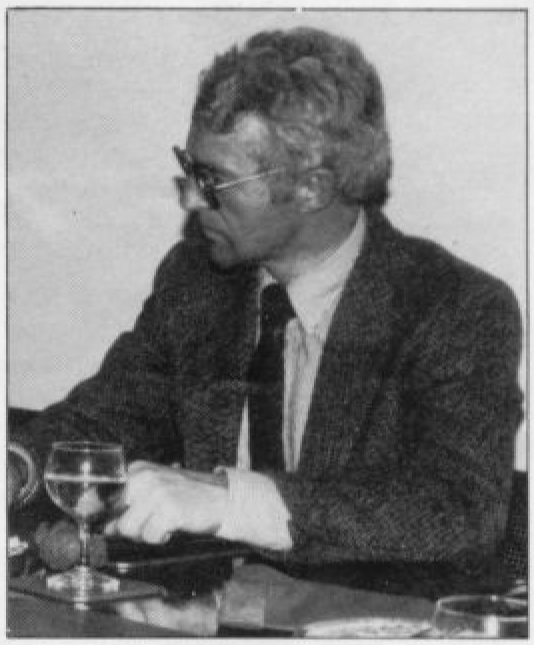
He has has two sisters, aged 12 and 13, and a five-year-old brother, but none is yet showing the same enthusiasm.

Matthews likes to keep fit in his spare time and when I visited him was keenly awaiting the Sinclair half-marathon due in Cambridge on July 17. No newcomer to running, he participated in last year's Boston marathon. Nevertheless he was taking the Cambridge event seriously. So, no doubt, was one of his fellow runners, Sir Clive Sinclair. Their speed, it seems, is not confined to work, nor their competitive spirit.

Matthews is reluctant to predict how long he will stay with Sinclair but says: "The chances are it will be a long time. I am really enjoying it so far and with the effects of recession hitting so many companies, it is fine to work for one as successful as this.

"We have built a very big turnover and we have the financial resources to expand into the future. That all gives the company a great deal of strength".

As part of the Sinclair management team, Matthews attends the monthly meetings of the board, together with Sir Clive, Nigel Searle the production director, and two research directors.



BILL MATTHEWS
'Sinclair computers are marvellous little machines'

At the moment the board doubles as a management committee which has enabled Matthews to look at every aspect of the company within a short space of time. He likes what he has seen and is full of praise for everyone, from the genius of Sir Clive through the dedication of the rest of the staff down to the excellent quality of the company canteen.

In short, he appears to have been converted and he speaks as if he has been with the company for years. St George's, Weybridge should be proud.

"ADVENTURES IN

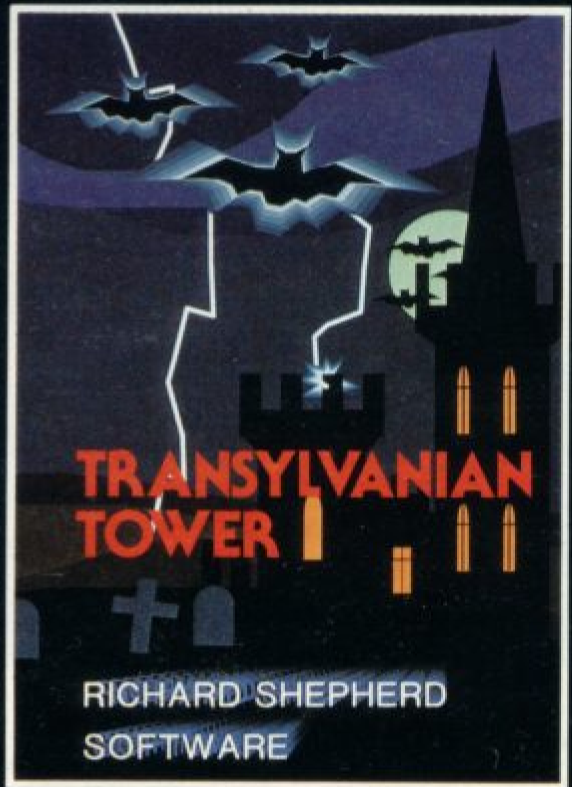
AVAILABLE FROM W.H. SM

AND ALL LEADIN

Transylvanian Tower

A spine chilling adventure... enter via the dungeons... navigate your way through 500 3-D rooms... survive the swooping vampire bats ... reach the terrifying top... confront and kill Count Kreepie, ridding the world of this Transylvanian Terror.

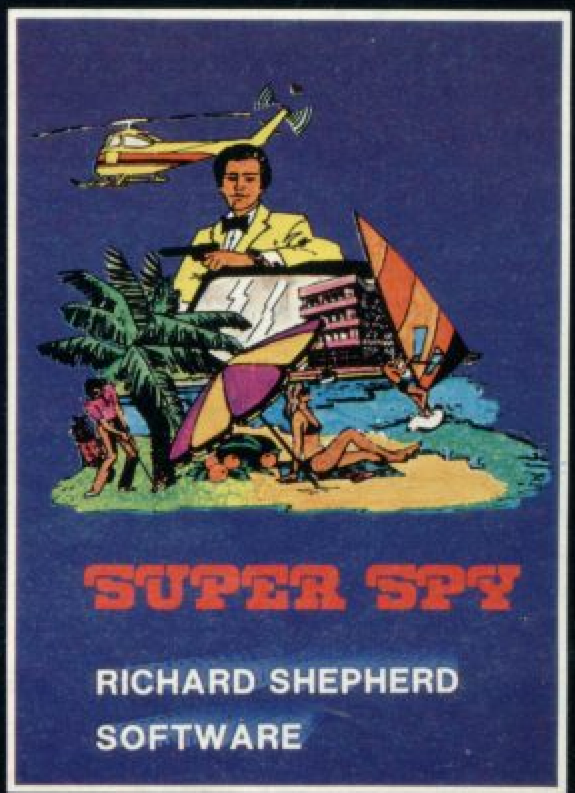
Can you survive the Top of the Tower? Full save routine for use during the hours of darkness! 48K Spectrum **£6.50**

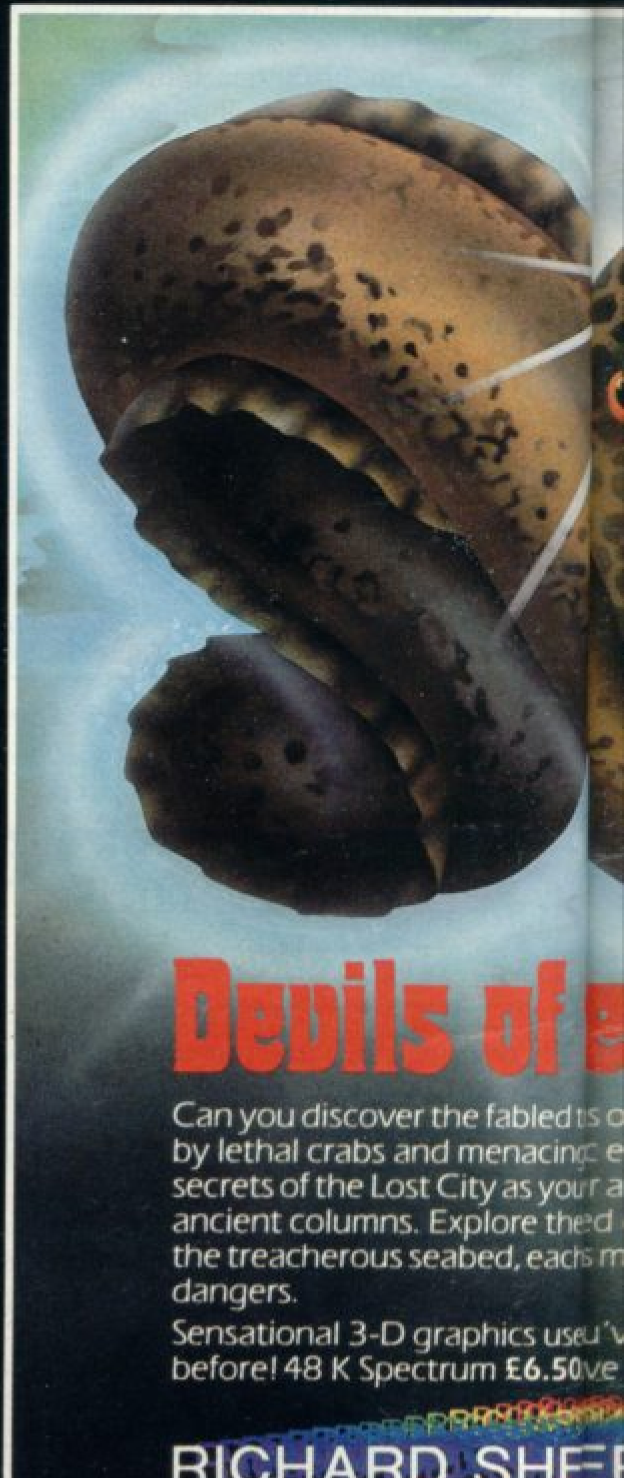


Super Spy

Locate the secret island hideaway of the mysterious meglomaniac Dr. Death. Follow his trail across continents, through complex puzzles and 3-D mazes. Discover the entrance to his underground lair — but beware — even with your death-defying gadgets his evil henchmen may still win the day!

With save routine for part time secret agents! 48K Spectrum **£6.50**





Can you discover the fabled ts o by lethal crabs and menacin e secrets of the Lost City as your a ancient columns. Explore the d the treacherous seabed, each m dangers.

Sensational 3-D graphics used v before! 48 K Spectrum **£6.50**ve

RICHARD SHE

SOFTWARE

DEALERS — GENEROUS DISCOUNTS AVAILABLE CO

RICHARD SHEPHERD SOFTWARE

ELM HOUSE, 23-25 ELMSHOTT LANE, CIPPENHAM, SLOUGH, BERKSHIRE.

CREDIT

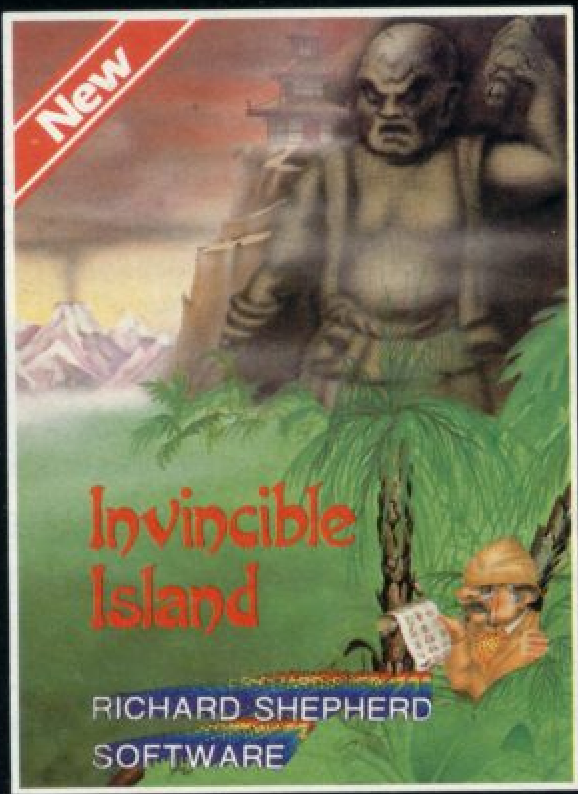
RING

FOR EX SE

All programs are sold subject to the condition that they may not, by way of trade or otherwise, be out.

NTO IMAGINATION"

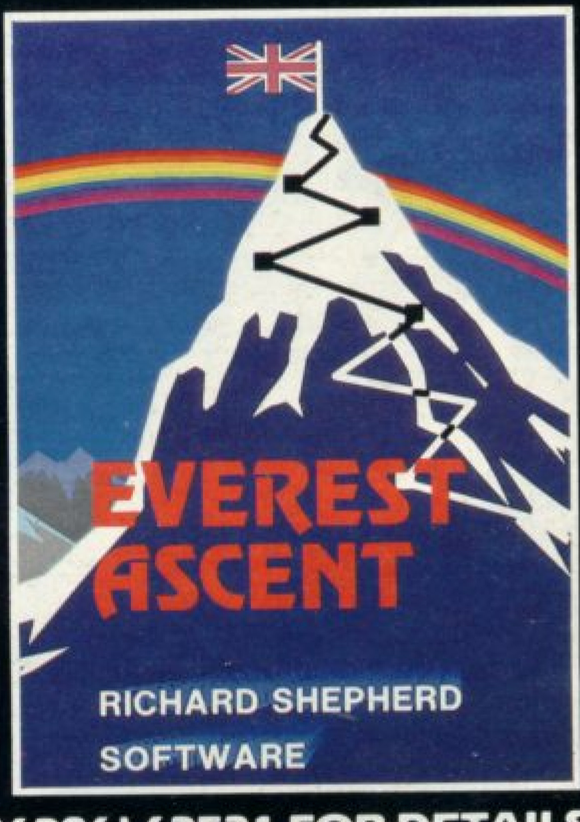
SMTH, JOHN MENZIES, BOOTS*
DINCOMPUTER STORES



Invincible Island
by Peter Cooke

Exhaust your ingenuity in the quest to find The Seven Parchments of Xaro and their meaning! Will they lead you to undreamt of treasures or eternal doom? Explore the mysteries of the stockade and puzzle within the Pagoda! A spectacular split screen graphic and text adventure to braintease you for weeks!

Adventurous graphics for every location. Save routine. 48K Spectrum **£6.50.**



Everest Ascent

Stake your claim to the top of the world in this strategic vertical adventure. Conquer the summit of the world's highest peak! . . . struggle from base camp to base camp . . . survive the elements . . . watch out for avalanches, thin ice and wayward sherpas . . . encounter abominable snowmen and cross bottomless crevasses!

A game of skill, strategy and planning — a graphic simulation of man's ultimate endeavour! Save routine. 48K Spectrum **£6.50.**

CONCT JOHN SALTHOUSE ON (06286) 63531 FOR DETAILS

EDIT CHOTLINE
RING (0)63531
OR EXPSSERVICE

RICHARD SHEPHERD SOFTWARE

ELM HOUSE, 23-25 ELMSHOTT LANE, CIPPENHAM, SLOUGH, BERKSHIRE.

* Selected titles available at larger branches

We meet the people who deal with your queries and complaints. A behind-the-scenes look at the Sinclair mailing house

The go-betweens

SINCLAIR RESEARCH computers are now in a class of their own in the home computer market, which the company has done most to create. Despite its policy of selling many of its products by mail order, however, few of its customers have had direct contact with the company.

The reason is one of its other well-known policies — sub-contracting. As much as possible of the business of making and selling Sinclair products is delegated to outside companies, allowing Sinclair Research to have a very small staff, slightly more than 50 at the moment.

Most people who manage to obtain their machines and software without too much trouble find that the nearest they get to Sinclair Research is a company which works from small offices and a warehouse on a featureless industrial estate in the Surrey town of Camberley.

The offices are those of GSI U.K., the address and telephone number of which appear in Sinclair advertisements and to which the mail order coupons are sent. It is GSI which has the task of being the first line of complaint and query, answering the growing number of Sinclair users throughout the country. It is also the GSI switchboard which plays music to soothe the patient person waiting for a query to be answered.

GSI U.K. is the British subsidiary of a French company which has built a reputation as a supplier of marketing services to the motor trade. It maintains a mailing list for a large number of motor dealers which can be used when a particular group of people need to be contacted. GSI also looks after the despatch of the information.

NIGEL BROWN
In charge of the Sinclair account

With its associates, GSI claims to be the biggest computer bureau for the motor trade in Europe. Its ultimate parent is CIT-Alcatel, a division of CIT, the power generating company which is the private equivalent of Britain's Central Electricity Generating Board.

The expertise for large mailings with a certain amount of mail order distribution led to the company being considered when Sinclair was looking for a distributor.

"We had a good deal of experience with keeping names and addresses on file and mailing, so it was natural to start doing something like the Sinclair work," says Nigel Brown, product manager at GSI, who is in charge of the Sinclair account.

"We started working with Sinclair in October, 1980 when the ZX-80 was still being produced. Before that Sinclair had looked after the distribution and we needed only four people at that stage."

At that time the company was known as Jaserve, changing to GSI when it was taken over in April, 1981.

"With the launch of the ZX-81 in March, 1981 the work became much bigger, which was when I became the account manager, "Brown adds.

Now there are 58 people working exclusively on the Sinclair contract, which has grown to become a substantial part of the British company's turnover.

Every one of the Sinclair products, the range of which now includes two computers — with different versions for the different television and power systems throughout the world — a printer, printer paper and a growing amount of software for all uses, passes through Camberley on its way to the corners of the world.

Each day two large, articulated container lorries make the long journey from Dundee to Surrey with the latest batch of Spectrums, ZX-81s and printers. Another slightly smaller lorry takes more products from the other Sinclair suppliers.

They all go into the large, highly-secure GSI warehouse but rarely stay for long. The items to be sent to distributors in export markets are sent to another warehouse, leaving GSI to deal with the distribution in Britain and those countries which have no agents.

Many items go almost immediately to the growing number of retailers now stocking the Spectrum and the ZX-81, while the rest go in batches of 50 to the mail order customers.

When the mail order system is running smoothly there is a clear number of stages in the processing of each order. Every morning the post is sorted into the queries and the orders, which are then further divided, depending on the goods required and the method of payment.

Cheques and cash are paid into a special holding account, where the money stays until the order has been satisfied; then it is transferred to the Sinclair sales account. Credit cards are not charged until the order has been fulfilled.

The orders are then entered on the computer and all the data stored on tape. When the goods are available to be sent, address labels are printed and another Spectrum or box of cassettes is ready for the post.

At the moment it is possible for GSI to say which stage a particular order has reached but not where it is within that stage. That resulted in many complaints during the Spectrum delay difficulties last year. Although an anxious customer could be told that their order had been received but was not about to be despatched, they could not find the place of their order in the queue. A new system is being installed which will allow the state of orders to be seen at a glance.

Brown adds that at the height of the problems it was difficult to be more specific, so that customers were not misled.

"We could give only general replies as the situation was always changing, so we thought it better not to give a specific answer which might then have to be altered," he says.

"Sinclair made a policy that everyone should be kept as informed about the position as possible and I think we were able to achieve that."

Most of the queries and complaints were by telephone, a customer service which has grown rapidly along with the rest of the Sinclair business. In October, 1980 there were only two lines. They have grown in stages, first to four, then eight, then 18, and last summer another 10 were added.

"In the early days on our two lines we were receiving about 100 calls a day. When the ZX-81 was launched that exploded to 1,000 a day. Now I think we probably have a capacity of about 2,000 a day," Brown says.

The calls deal with a vast number of subjects. Apart from complaints about non-delivery or faulty machines, many users need advice on particular aspects of using the machines. Some things, such as saving and loading difficulties, are the subjects of regular queries and there are almost 40 leaflets giving advice which can be sent.

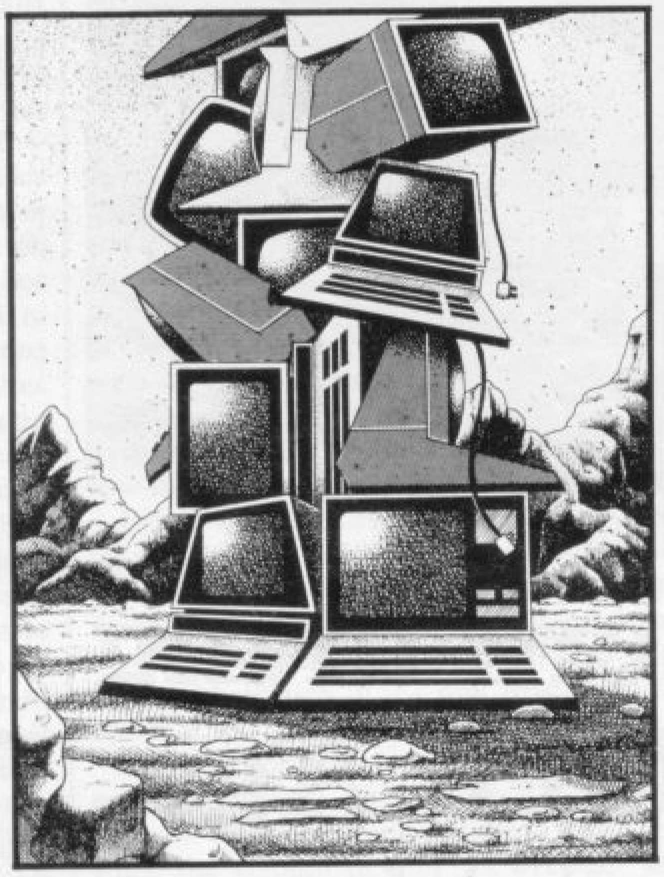
For the more complicated queries there are three more technically-minded people available who attempt to find an answer.

Whatever the difficulty most of the callers are still polite. Even when the delays were at their worst last year, no-one became abusive when making complaints.

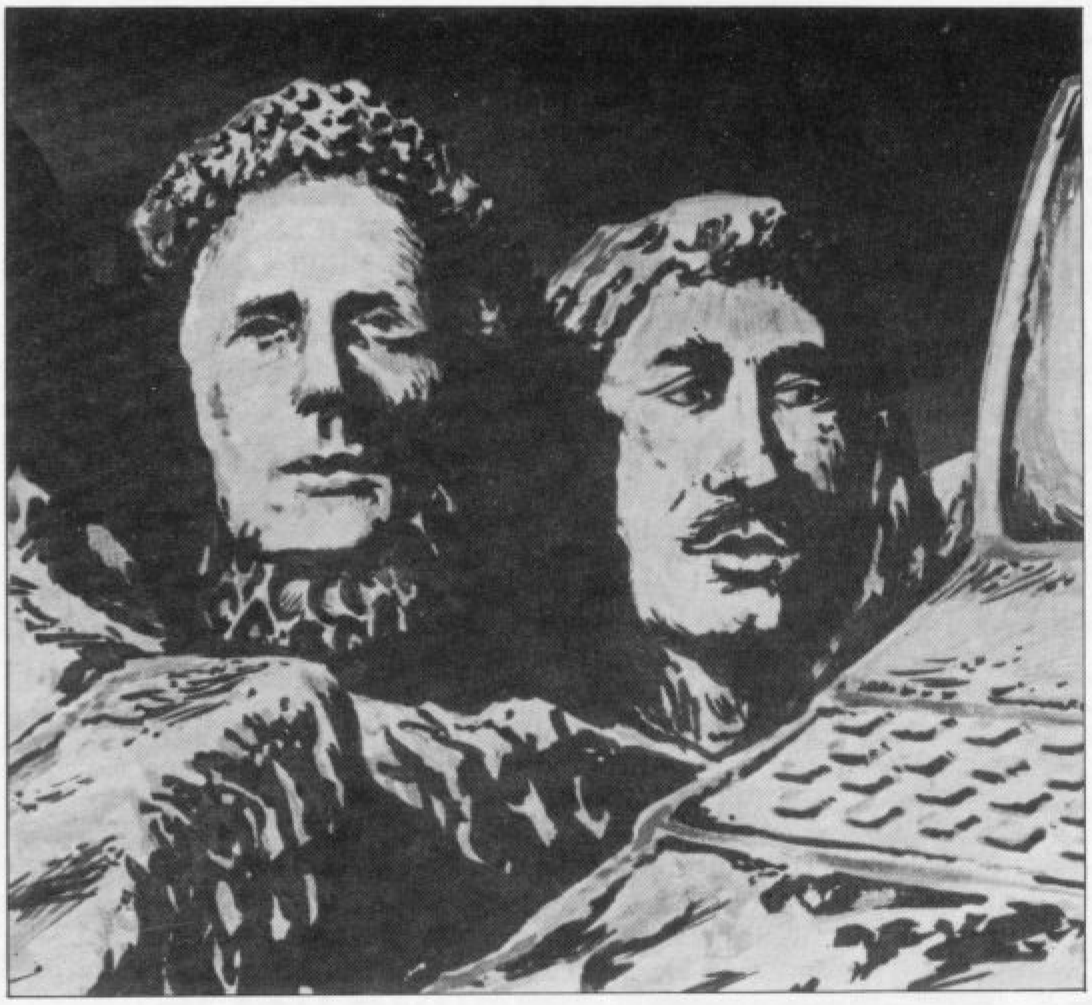
The GSI position as the public face of Sinclair Research will be reduced as the retail sales of Sinclair products increases. It is likely to remain the first place people will contact with their problems and complaints and will remain an important part of a string of companies involved in the production and selling of Britain's most popular computers.

The warehouse filled with Sinclair products ready for despatch

Futurology



A speech by Sir Clive Sinclair to the British Mensa Society on the expectation of a new Golden Era being caused by the growth of computer use began a series on what people believe will be the result of the increase in computer ownership.



Computers bring new Golden Age

Sir Clive Sinclair sees a bright future

AS WELL as being head of Sinclair Research, Sir Clive Sinclair is chairman of British Mensa, an exclusive club whose members have IQs which reach the genius level.

In a speech at the Mensa Golden Ages symposium at Cambridge, Sir Clive outlined his ideas for the future, not of his range of personal computers but of the Western civilised world. He said: "I intend arguing that the most Golden Age of man's history may well lie before us, if we can only move in the right direction."

The new age would need to be triggered by an event which will startle society. The trigger, he explained, would be something similar to the invention of writing or moving type. He said: "Both of those developments reduced the cost of data transmission by a factor of 100."

He saw leisure, or periods of time not occupied by formal work, as an opportunity to broaden the mind. If the trigger occurs at the proper time and the Golden Age arrives "the body of men arises which can turn its attention to matters other than necessities. Thus wealthy patrons produce the great flowerings of arts which are a feature of the Golden Ages.

"Equally, the Golden Ages are often marked by one great individual, a type of philosopher-prince, e.g., Pericles, Augustus, Lorenzo de Medici, Elizabeth I and Louis XIV."

In business operations, Sir Clive seems to regard the personal approach best — one man at the head of a company. He has stressed that approach many times through Sinclair Research, so that now he is as famous as his machines, whereas other manufacturers remain masked by their company exteriors.

During his speech he referred his ideas to the present day. He saw the Golden Age as being very close. Some of the features which marked the Golden Ages of the past could be identified within our time. That could place us on the threshold of a new Golden Age. To demonstrate it, Sir Clive returned to the idea of a trigger.

"Is there a trigger? It so happens that another hundred-fold reduction in the cost of data publication and transmission is about to occur. A single 12in. diameter optical disc, being developed for use with TV can, remarkably enough, contain the information of 10,000 books and that disc will cost not much more than a few books—almost, in fact, a thousand-fold reduction in costs."

The reduction in costs and the innovations in mass marketing are compared to what Sir Clive calls "the potential of the individual". Until now, society has accepted that people will work together in large groups. People work in large companies, they commute into towns and cities every working day. That massing of the working population is the motive force behind the present state of the economy.

Sir Clive said: "We have for some time been passing through a great industrial age in which the economic basis of society has demanded the bringing together of people in great numbers, many thousands per factory, many millions per city. I believe that our move away from this type of organisation will restore the potential of the individual."

Individual human potential is something he seems largely to favour. While Sinclair Research is a company, like many others, where everyone pulls together, it is still mostly a one-man operation. Sir Clive is the man who defines what he wants and lays-out the timetable for its development.

That style of operation has so far proved successful, first with the ZX-80 and ZX-81 and then with the Spectrum. The hundreds of firms which give support to Sinclair microcomputers would also seem to prove his point to be correct.

He sees the new Golden Age as being a time of the mind, with less stress put on the body and building culture rather than labouring. He feels that another Golden Age requirement is an abundant supply of patrons, people who can appreciate, as well as create, art.

"We have a well-educated population, a society which reveres the arts, and have become a world centre for music and for the written word."

The reason for the swing towards

cultural pursuits is marked with the stigma of a current curse on society.

Sir Clive said: "We have potential artists, partly for the sad reason that we have three million unemployed; this is not a passing phase of recession but a trend which will last until the end of the century, during which I expect the manufacturing industry to shed a further seven million jobs and for the proportion employed in manufacturing to decline from some 42 percent of the population to less than 10 percent. This will occur as automated systems are now radically cheaper than manual costs."

The resulting factors of unemployment due to technological innovation and automation will leave the population with a great deal of spare time if present trends continue. If the number of unemployed rises to more than 90 percent it may be necessary to re-define the term altogether.

"Many, if not all, of today's young people will always work for small organisations and indeed must found them. We must encourage people to follow this route if we are to create future employment—whether in high technology, in a revival of a class, or in service industries."

Sir Clive foresees a new "creative endeavour". People are experiencing new technology, seeing what it can do for them. It can relieve them of manual tasks so that they can use their minds more fully.

Young people were just beginning to learn about new technology. According to Sir Clive, the learning process would be only the beginning.

Learning about new technology through machines such as the ZX-81 and the Spectrum which, because of low prices, were within the reach of nearly everybody, was the beginning of a process which may lead to what he believes is a new Golden Age.

"Because we no longer need to devote the bulk of our time to the production of objects, I can see the plateau of a Golden Age before us. Certainly we may need inspiration and leadership, great building, a bridge over rather than a tunnel under the Channel.

"Early in the next century we will have made intelligent machines ending for all time the pattern of drudgery. It may be that Western civilisation, seeded in seventh-century Ireland, is only just about to flower."

To some, his ideas may seem like science fiction but some cynics said that a machine like the Spectrum was not possible only a short time previously.

Less work, more computer play

Future leisure activities surveyed

MORE THAN 2,000 years ago, Aristotle made the comment that we work to have leisure. For him, leisure was a reality, since all the toiling and much of the producing was done by human slaves. For most other people then, and since, work has been the essential part of human life as people have struggled just to survive. Leisure has been, at the best, a residual and generally meagre amount of time.

Today, because of the introduction of computers and microprocessors, we are developing a new type of slave, in the form of electronic robots of all kinds. That represents a major change in our lives, which will give new meanings and new dimensions to work and leisure.

Perhaps, most of all, new technology means a growth in the amount of leisure time. That is not an unmixed blessing. Although people will need to work less, they will also have to learn how to develop their lives in the way Aristotle meant when he talked about leisure.

In some measure, we have already entered an age of leisure, with a milestone at the beginning of the 1970s when the amount of time the average full-time worker spent at work during a year fell below the amount of time available for leisure.

Since then, a typical worker's leisure time has increased by some seven percent to more than 2,500 hours a year, while the number of working hours has fallen to 1,950, including travel to and from work.

Less positively, we now have the large amount of enforced and maldistributed free time represented by more than three million unemployed.

With the expansion of free time and the need to find ways of occupying it in a satisfying manner, the development of the microcomputer, with its time-intensive quality in use, seems to be particularly fortuitous. Certainly home computers will become increasingly important in people's lives but we need to be realistic about just how fast private ownership of computers will develop and also about exactly how the machines will be used.

Without becoming involved too deeply in the psychology of leisure, it helps when thinking of the possible roles microcomputers might play in our leisure lives to consider for what we use our leisure. It is possible to identify three functions of leisure which form something of a hierarchy—rest and recuperation from work, entertainment and the relief of boredom and, finally, personal and social development.

In the era of leisure we are entering the function of leisure as rest and recuperation declines in importance as the amount of work lessens and the effort involved decreases. That puts the emphasis in leisure time use on the two other functions of entertainment and self-development, both areas where we expect to see the microcomputer play an increasingly large part.

An obvious role microcomputers play in the area of entertainment is of being virtually an infinite compendium of games. They can provide an unending source of pastimes for those who wish to occupy their leisure in this way.

It could be argued that people could use a chess set or a pack of cards instead of a microcomputer for those simpler activities. The answer lies probably in the basic attraction of using new technology, as well as in the perpetual self-challenging quality of computer games and the wide variety of pastimes available from the one machine.

Once attracted to the computer, the games player is likely to look progressively for more complex games to play and may eventually take the step towards developing improved or original programs. At that stage, the computer moves from being a source of entertainment to potentially a very time-consuming hobby, offering great opportunities for individual learning and development.

Many people have turned to microcomputers without any particular emphasis on games playing. Either way, and whether interest lies primarily in the hardware or software, all computer hobbyists have found a leisure occupation of absorbing interest. In the future, when there may be considerably more free time but possibly not so much extra

money to spend on leisure, the time-absorbing nature of computing is a valuable characteristic.

In addition, as members of computer groups know, there can be a strong social element in the hobby, producing new friendships as well as a useful exchange of experience and ideas.

There are other aspects of what we have termed self-development in which we believe the microcomputer potentially has an important leisure role to play. By self-development, we mean the whole process of learning and enlarging one's mental and physical capabilities with the aim of leading a fuller and richer life.

Naturally the microcomputer cannot take the place of weightlifting, jogging or other sports in improving physical capabilities but, as a provider of aid to educational and cultural development, it could have a vital part to play. Previously it was often people of leisure who went to university to enjoy the benefit of learning, often for its own sake; in the future, the micro will help to take learning to the homes of all who want it.

How quickly will all this happen? How soon can we expect to see a micro in every home, as some commentators envisage? How many people will, in practice, be encouraged to use this new gadget to educate themselves and their families?

Our forecasts appear to be somewhat more conservative than those of many others. We expect that, by the end of the decade, something like one-fifth of all households, some four million homes, will have a micro bought for personal use; others, as now, will have machines used both for the business and family.

Behind that view lies the assumption that microcomputer prices will fall to around one-third of what they are at present. An even sharper fall in price obviously would boost demand. We do not think a very high proportion of the 33 percent of households where the head of the household is over 60 years of age will be buying a microcomputer, even if it costs only £10.

Despite the growing amount of free time, the majority of people are likely to be fairly slow to recognise what the microcomputer can offer them. The attraction of games-playing is obvious but for many people the idea of leisure as a period of education is a novel one and, during the 1980s, most of the demand for educational use is likely to be stimulated by children's needs.

It will probably not be until the 1990s, when the first generation of children to whom computers are a part of everyday life become adults, that a real widespread educational use of home computers will develop.

Since many people, not least the unemployed, will be having a leisure problem well before then, we hope that those already keen on computing will do all they can to try to make our forecasts seem too pessimistic.

Aristotle probably would have enjoyed the challenge of microcomputing but even he might have needed some encouragement to take the first steps in this new leisure direction.

● *Bill Martin and Sandra Mason of Leisure Consultants, Sudbury, Suffolk have published a report on leisure in the 1990s.*

Micro junkies

Chris Reynolds

MANY YOUNGSTERS have home computers because they, or their parents, feel that owning such a computer will help them in their careers. As a university lecturer responsible for training future computer professionals, I have my doubts. Let me explain.

In the last 30 years the uses of computers as part of practical working information systems have mushroomed. For most of that time there has been a desperate shortage of suitably-experienced staff. Salaries rocketed as companies bid to obtain employees with the greatest length of experience, apparently regardless of quality. The whole was surrounded with the prestige of being at the forefront of modern technology, at least in the eyes of one's neighbours.

Much of the gloss has now been shed. There is, of course, still a shortage of good computer professionals but the incompetent now find it almost impossible to climb on the bandwagon, and salaries are no longer so wildly out of line with other occupations. One no longer hears people boasting that they know someone who knows someone who works with computers. In fact, it seems likely that in five years having a computer in the house will be socially as significant as is owning a digital watch today. When acne-embarrassed schoolboys with home computers are ten-a-penny the simple ability to program will have no value in the job market.

Those changes will have a major effect on the structure of the computer profession. The need for highly-skilled people to work on research and development projects in the computer industry will continue. The majority of existing professionals work for companies which use computers as tools to help the company business and it is in that area that the biggest changes will take place. The pressure will be for more flexible and easier-to-use systems with the minimum of fuss.

A prime requirement will be for staff able to communicate with other people, verbally and in writing, with the minimum of jargon. Knowledge of management, economics and psychology, and the design of systems are next on the list. A good understanding of what a computer can reasonably be expected to do is of far greater importance than the ability to PEEK and POKE on a particular make of microcomputer.

Universities already have moved in that direction and introduced courses which anticipate that future need. For instance, Brunel University has a Systems and Information Management course which has been running for five years and which attracted 28 good students this year, compared to 22 on its more conventional course. Because it is felt that breadth of experience is important, students who do not have A levels in computer science and mathematics are preferred to those who have already specialised narrowly with double mathematics and computer science.

Many of the leading 21st century computer professionals are now at school and in the light of the foregoing comments, it is useful to speculate what they are doing now. We can be certain that they will be well-acquainted with modern electronics technology. Digital watches and pocket calculators will be taken for granted. Their parents will have television sets with teletext and a variety of electronic games. As soon as they are old enough they will use autobanks and credit cards to buy things such as electronic organs. Even if they never saw a general-purpose computer they would take for granted keys to be pressed, video displays, and automatic information processing.

While at school they will almost certainly have been given a computer appreciation course and may have had computer-aided instruction. Most will have taken O and A level computer science. That will not be because this is necessary for their careers but because

'The pressure will be on for more flexible and easier-to-use systems'

the education system encourages early specialisation.

Socially, most of them will be good mixers. They are therefore likely to be found in the Boy Scouts, the school band or the local cricket team.

One thing not mentioned is ownership of a home computer. The reason is that the evidence at Brunel suggests that in many cases a private micro can have an adverse effect on student studies and sometimes on employment prospects.

There are a number of reasons. The first is that to understand and use a language well, you have to be able to think in that language, be it French, Arabic, Pascal or Cobol. Students who have written a large number of programs in a single language, often on a single machine, have difficulty in transferring to other languages.

Gerald Weinberg, in his book *The Psychology of Computer Programming,* showed how easy it was to guess a student's former language by the stylistic errors he makes in learning a new language. Ten years later the problem is much the same. Most novice students learn rapidly the essentials of modern programming concepts, such as block, structuring and recursion.

Students who have extensive experience in old-fashioned languages which lack those features often insist on using more powerful high-level languages as if the newer features did not exist. That is apparently because they find it easier to write longer, inelegant programs than to learn something new.

The second problem is that writing programs and playing games on a home computer is most entertaining. The ability to use a home computer is, however, of little relevance to any but the most junior jobs in the computer field. Most adults would discourage a child from wanting to become a television news reader simply because he had built a crystal radio at the age of eleven. Unfortunately computers are so new, and so mysterious to many adults, that most teachers and parents are not sufficiently knowledgeable to recognise a fun hobby as just that and, as a result, they encourage children to attempt to follow a career path for which they may be almost totally unsuited.

The big danger seems to be the bright but socially-gauche adolescent. He finds the school computer more friendly than his school-mates and develops an interest rapidly. A home computer is bought by enthusiastic parents to encourage him in what is certain to be a wonderful career in this marvellous technology. He is delighted to be allowed to spend many hours every evening enjoying himself in the privacy of his room. Socially, he finds he can boast about his wonderful programs which look most impressive, with flashing screens and perhaps even sound effects.

Because of all this praise and the absence of anyone who can assess the quality of his work properly, he soon becomes convinced that he is a computer genius and spends even more time at the keyboard. He has become a code junkie who craves for his two-hours-a-day session at the keyboard.

In fact, the idea of code junkies is not new. In the last decade we have had several who have become addicted to

'The big danger seems to be the bright but socially-gauche adolescent. He finds the school computer more friendly than his school-mates and develops an interest rapidly'

the university computer. From experience we know that almost all such students fail the course. They prove to be almost unemployable because of their inability to get on well with people, and because of a marked reluctance to work with, rather than play with, computers.

In 1981, for the first time, we had the problem of new undergraduates who were already code junkies before they arrived. This year the number of students who have home computers on arrival has more than doubled and it is suspected that many of them will prove to be junkies.

It has been said that home computers and computer games will help to keep the future unemployed occupied in an enjoyable manner. That may well be true in the long term, when society has become adjusted to the new technology. In the short term there is a danger that the majority of the unemployed who play such games will be code junkies who caught the bug during a vulnerable period of adolescence.

● *Dr Chris Reynolds is reader in computer science at Brunel University. He organises the computer science teaching for first-year mathematics computer science students. He also researches improved systems for the man–computer interface.*


Doric Computer Services
Present
The Oracle's Cave
AN AMAZING NEW ADVENTURE GAME WITH ANIMATED GRAPHICS FEATURING:-
• A completely new cave layout, monster position and event sequence generated for every game.
• A time limit of 5 'days' on every adventure.
• Continuously displayed high resolution graphics and text.
• All monsters, articles and locations depicted to a level of detail that pushes SPECTRUM graphics to the limit.
"The closest yet to a true animated graphic adventure... ...excellent value for money." POPULAR COMPUTING WEEKLY.
FOR THE 48K SPECTRUM
AVAILABLE FROM W.H. SMITH, BOOTS AND OTHER LEADING RETAILERS OR DIRECT FROM:-
DORIC COMPUTER SERVICES
3 THE OASIS, GLENFIELD
LEICESTER LE3 8QS.
£7.95
(Inc VAT)
Postage & Packing free.


THE FABULOUS
CASSETTE
50
FROM cascade
50 GAMES ON ONE GREAT CASSETTE
ONLY £9-95
DON'T MISS THIS INCREDIBLE OFFER
50 FANTASTIC GAMES ON ONE CASSETTE
ONLY £9.95 (INC.P&P and VAT)
SPECTRUM
50 fabulous games
VIC 20
50 fabulous games
EXPRESS DELIVERY - ORDER NOW
Please send me by return of post, Cassette 50 at £9.95 per tape. I enclose a cheque/postal order for £
made payable to Cascade Games.
Name
Address
Post Code
Country
Dealers & Stockists enquiries welcome.
Please send me (tick appropriate box)
Spectrum
VIC
cascade
Cascade Games,
Suite 4, 1-3 Haywra Crescent,
Harrogate, N. Yorkshire, HG1 5BG.
England.
Telephone: (0423) 504526
SUA/83

Hooked on arrogance

J R Bird supports micro training

COMPUTING is much more than programming. It is concerned with the flexible use and application of ideas. The ideas are structured logically by means of a language.

That is one of the reasons why many of the conclusions of Dr Chris Reynolds are incorrect. Although much of what he says is valid, the tone was prejudiced and negative.

All the problems of vocational computer training were blamed on owners of the home computer. The only slight comment in favour of the powerful little machines appeared in the last paragraph but even then they were dismissed as little more than games machines.

To condemn home microcomputer owners as potential code junkies—people with an addictive, perverse interest in computers—probably as failures in academic life and as unemployable is a form of professional arrogance. Most people who have an interest in computing also have other interests.

It is the same arrogance which leads many people to claim exclusive rights to comment on particular branches of knowledge, especially how those branches will develop in the future.

There are likely to be many problems associated with the future in areas such as jobs or careers, leisure and education. None of those can be answered by any one group, be they teachers, employers, lecturers or workers, claiming they have the exclusive right to comment and decision-making.

Neither should Reynolds be advocating novice students in computing. That kind of logic is rather like a professor French insisting on novice speakers—those who do not speak yet—if he is to produce good speakers of French.

A teacher of a foreign language must take account not only the students' own language but also their development within that language.

If a student has had experience of a computer language it is reasonable to expect lecturers to take it into account. If specific languages cause problems, the lecturer must identify the problem and work out ways of overcoming them. A lecturer should not advocate novice students; it is no solution.

There is a clear and unfortunate rift between academics and non-academics, professionals and amateurs. Academics tend to be more concerned with protecting their access to knowledge rather than sharing it. Professionals seem to be more concerned with keeping their right to knowledge rather than sharing it with amateurs.

Computing at a high level is for a small number of highly-qualified students and to suggest that the needs of those few are best served by denying the vast majority of home micro owners use of their leisure time activities is wrong. Academics appear to see things in one of two ways. Either they are shut in ivory towers researching and learning for pure knowledge, or are selling their ability to industry.

> 'To condemn micro owners as failures is a form of arrogance'

While I should not subscribe wholly to that view, I wish the apparent rift which contributes to it did not exist.

With the introduction of new technology, people will find vast changes in their life-style; work may never be the same again. The concept of five to 16 education being sufficient may disappear: re-training several times may become the norm.

Most people with home microcomputers would not suggest that their work with the computer would change the world but they would not suggest that those with computer PQX were inferior to them or that all mainframe work was necessarily superior.

The assumption common to many vocation-minded computer studies/computer science lecturers, that their work was important, simply because they train professional workers, is to misread the future. Before long, the day of the large "brain" will be over and the day of the small, powerful computer will begin.

It is also clear that plain language or even interactive voice computers are on the way. When the man in the street can talk to a computer and ask it questions, the mystery of the languages probably will disappear. Before that day arrives, however, we have the problem of many computer languages, but it is surely not beyond the understanding of professionals in the computer field to structure courses to take that fact into account.

I notice that the Open University is to use a further form of Basic in its course Computing and Computers. I am not condemning the intellectual level of the OU course but rather regretting the introduction of yet another dialect.

Even with the limitation of the new OUSBASIC, the Open University is showing the way to other universities and polytechnics. The course has been designed to introduce beginners to skills and techniques of computing. Most micro owners would concede that their programs and computing would be improved for a better knowledge of structure and logic, not to mention better techniques.

So, in effect, what we have is a university or polytechnic department with the skills and resources to teach both the structure and techniques to improve the standard of home micro owners. Alongside that pool of skill we have a large number, perhaps some half-million at the moment, who would no doubt welcome some of that skill being taught to them.

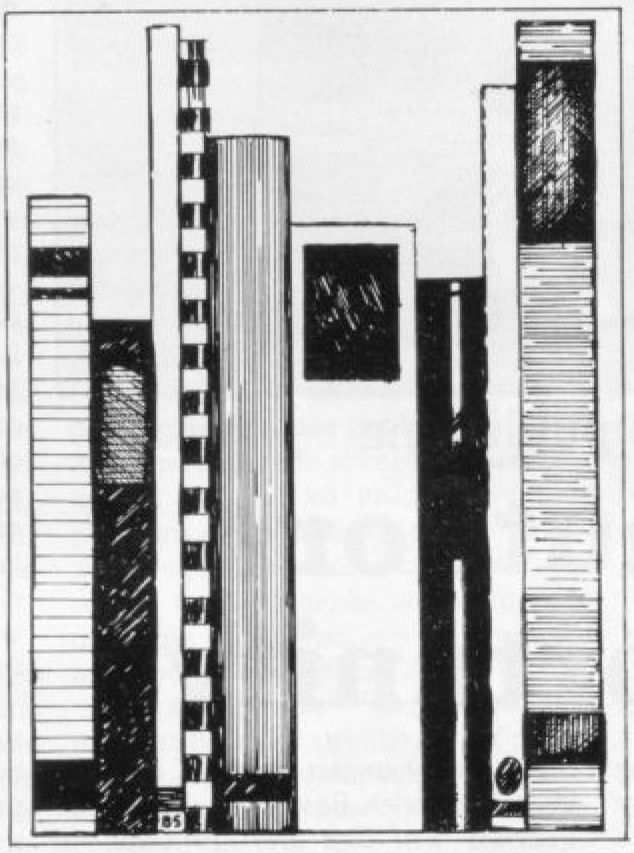
What better way of bridging the academic, professional and non-academic amateur rift than bringing the two together?

That would create a new area of non-vocational studies for the institutes of higher education. As a spin-off, it would also retain in those institutions some of the teaching jobs which are disappearing under the present retrenchment in the higher education sector.

Clearly there would be advantages for the students of such courses. They would learn better techniques, they would learn better ways of using computers, they would learn other computer languages, not to compete with the professionals but to use in their private studies.

The interaction between micro owners and the professionals, both at the formal teaching and informal social levels, could only benefit both. Experience from both sides could stimulate more courses and lead to a leisure time education industry.

Books



Publications of varying quality and quantity have gone on sale during the year. John Gilbert looks at the growing bookshelf and selects some typical examples of recent developments.

John Gilbert assesses publishing

Young authors get opportunity

THE COMPUTER publishing market has developed so quickly this year that publishers have been desperate to lay their hands on anyone who knows something interesting about Sinclair computers, programming techniques, or who has some programs they want to sell.

Some publishers have even asked teenagers to write books because insufficient adult writers have been able to get to grips with the subject. That kind of move sets a precedent in the publishing industry. No other sector has ever sought young writers with such vigour. If you can write and you have an above-average knowledge of computers there is a good chance that a publisher will contract you for at least one book. The problem is, and has always been, that young writers know little about the publishing industry and, through no conscious fault of the signing company or the young author, writers do not obtain everything they should. Fortunately that does not happen often but it is a side-effect of the market growth and the urgency with which publishers seek titles. The youngest writer of the year must be Patrick Bossert, author of the Penguin *You Can Do The Cube.* In August, Penguin released one of its first micro-computer books by the 14-year-old. Unfortunately *Micro Games* was little more than a book of listings, a stage though which many companies such as Shiva Publishing, Interface and Melbourne House passed earlier in the year.

Penguin seems to have relied on Bossert's fame with the Rubik Cube and that the puzzle and computers share the same intellectual image in the public mind. Just because Bossert can do the cube, however, does not make him an automatic genius at computer programming.

The concepts for most of the programs in his book existed earlier in the year when all you could buy in terms of the Spectrum were books of listings. There is little that is new in the title — a pity, since it is from Penguin, a publisher renowned for its quality of output.

Books of listings were popular at the beginning of the year when the ZX-81 had more of the limelight than the Spectrum. Authors such as Tim Hartnell were having at least one book published a month. Most of those books were for the ZX-81, as most publishers had not yet advanced to the Spectrum, although it was launched in April, 1982.

Before the beginning of 1983 the only publishers to try for something extra from the ZX-81 were Interface, Melbourne House and Shiva. All were still small but it is a mark of their innovation which shows their success and expansion to date. Now all three have a large share of the publishing market where Sinclair machines are concerned.

By May all three companies had done something different for the Spectrum market. Machine code programming for the Spectrum was a subject which would sell books and the big three publishers knew it.

Shiva produced *Spectrum Machine Code,* by Ian Stewart and Robin Jones. It was launched as part of the Friendly Micro series and, although it did not cover the area in as much depth as some of the American books about the Z-80 processor, it provided an excellent grounding in machine and assembly language. It also added a humorous element missing from many other books with 'bug' cartoons spread throughout the pages.

The other publisher renowned for its stock of titles on machine code is Melbourne House. It has two machine code titles which cover the ZX-81 and Spectrum. Both are similar in approach and it seems as if the Spectrum version was edited from that of the ZX-81, with extra examples showing colour and sound added.

The other range of machine code books from Melbourne House is by Dr Ian Logan and, in the case of *The Complete ROM Disassembly,* written with Dr Frank O'Hara. The books are excellent value and contain a good deal of necessary information for the machine code programmer.

Because of his knowledge of the Spectrum ROM, Logan was asked by Sinclair Research to write the routines for the Microdrive ROM. As a result, and with the blessing of Sinclair Research, he wrote the *Spectrum Microdrive Book.* It includes much information about the drives, Interface One and the possibility of adding or patching-in extra commands to the Basic.

The emergence of the book resulted in a rash of similar texts from publishers trying to keep in the race for the most up-to-date information. In most cases the follow-up texts represented a re-arrangement of the original but, unfortunately, that is not so with the new Microdrive texts from Interface and Sunshine Books.

When first exhibited at the Personal Computer World Show in September, the Interface book was little more than a slim cardboard-bound photo-copy. It was planned to use it as the basis for a 'proper' publication. The Sunshine effort, however, was better-presented.

The author of the Sunshine *Master Your ZX Microdrive* is Andrew Pennel, a friend of Logan. His book contained information which Logan's could not. One reason was that he was not limited by what he could say. Although Logan speaks with an authority which is difficult to match, Pennel's book is slightly better as it contains information which Sinclair Research did not want used in Logan's book.

Even with the restrictions, however, the Logan book is good value so far as money and information are concerned.

The release of the Microdrive texts has introduced a new area to the computer publishing market. We have had books on machines, books of listings, and books showing software techniques but there had, until then, been no books on one specific peripheral for a machine.

The Microdrive seems to have opened an area which could soon include how to get the best from your sound generator or using a disc drive with a Spectrum. Book titles such as that may seem absurd now but with the way books are becoming so machine-dependent, and with the search for new areas to write about, such titles may become available.

The information in the Melbourne House book on the Microdrive contains a good deal of machine code. The publisher is still determined to introduce machine language anywhere it can and the release of another machine code book for the Spectrum was inevitable before too long. The new book is *Supercharge Your Spectrum* and many pages are occupied by machine code listings. They include routines to search for strings in programs, re-number lines, and delete blocks of lines. It has proved extremely useful to Spectrum owners who know nothing about machine code but who want toolkit routines without having to buy several cassette-based programs.

It is useful in another respect. It is possible for someone just learning about Z-80 machine code to read the program listings and get an insight into how programs are structured and the way in which some statements can work with others. It also provides an incentive to use some of the routines in your own machine code programs.

Not all publishers are interested in machine code and the market has plenty of support from companies with other ideas. Yet another new type of book was launched for the Spectrum by Sunshine. Until the release of *Spectrum Adventures,* by Roy Carnell and Tony Bridge, computer owners interested in adventure games-playing or writing had to rely solely on magazines.

The new book improved that situation, however, and showed the reader how to write graphics adventures. Little is said about decoding player responses or generating textual adventures but the book still marks a new area for publishers to exploit.

While the areas which belong to the games and utility sector have developed by leaps and bounds, the business and education markets are still nothing more than a mess. Little has been done in book form to aid this ailing though very important part of the industry. There have been a few general books on business applications, such as *Databases for Fun and Profit* from Granada, but little business-orientated work has been done.

'While the areas which belong to the games and utility sector have developed by leaps and bounds, the business and education markets are still nothing more than a mess'

The same is true of the education market, although some publishers, such as Granada and Longmans, are starting to see the potential. The object seems to be to produce as many programmer-orientated books as possible. Unfortunately that leaves the market for the computer user who does not want to be bogged down by technology as undeveloped as it was early in the year.

Several new areas in the publishing industry include machine code programming and programming techniques. They are over-developed and that is proving expensive to the other users who want to use Spectrums at home or at school for accounts or homework.

In the end that can only do the market and, indirectly, computer manufacturers, a good deal of harm. The areas in which computers can be used, such as education and business, will be under-developed. Many children will know how to program but very few will have ideas on how to use those programming talents.

Interface, 44–46 Earls Court Road, London W8 6EJ.
Melbourne House, 131 Trafalgar Road, Greenwich, London SE10.
Puffin Books, Penguin Books Ltd, Harmondsworth, Middlesex.
Shiva Publishing, 4 Church Lane, Nantwich, Cheshire CW5 5RQ.
Sunshine, 12–13 Little Newport Street, London WC2R 3LD.
Granada, 8 Grafton Street, London W1X 3LA.

PROGRAM PRINTOUT

One of the most popular sections of *Sinclair User* is our program listings. We have reprinted some of the best, with two listings which have not been published previously.

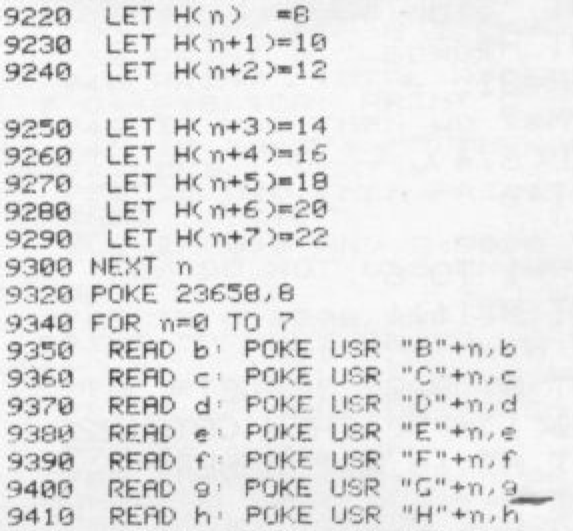
To help with entering graphics characters we have adopted a system of writing the characters for the ZX-81. We indicate inverse characters by the letter i and graphics by g, so that an inverse letter W is shown as iW and the graphics character on key 6 is denoted by g6. Spaces are shown by sp and inverse spaces are isp. If some occur together, for instance a row of six spaces , they are shown by *6sp and where there is a combination of characters each one is divided by a colon, thus sp:isp:6*g6 means a space followed by an inverse space and then six characters on the 6 key. Where whole words are written in inverse letters they appear in the listings as lower-case letters.
In the Spectrum listings, letters to be entered in graphics mode are underlined, while other graphics instructions are underlined and take the form shown above, with the addition that inverse graphics characters are represented by the letters "ig".

KEN RYLETT of Burnage, Manchester wrote **Solitaire** for the 16K Spectrum to allow people to play the traditional game. Move one spot over the next to an empty space. The peg which has been jumped will then disappear. The aim is to finish with only one spot remaining. Moves should be entered as number, then letter.

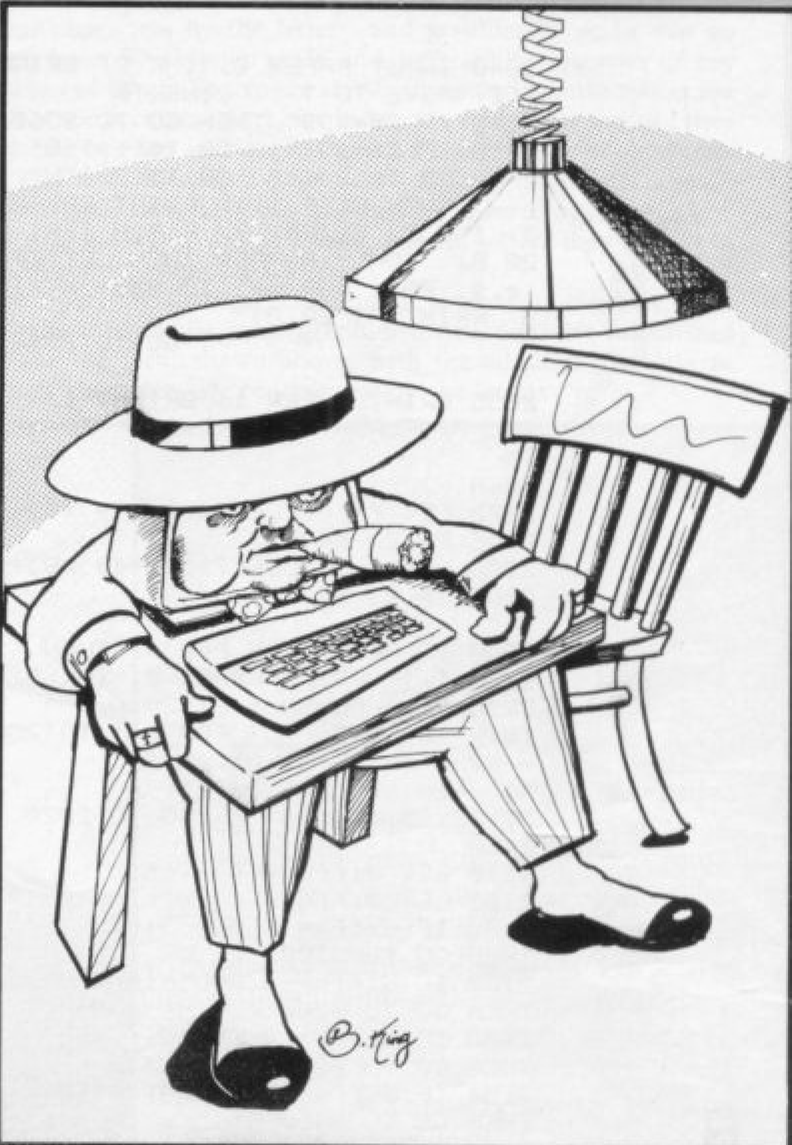
Letters to be entered in graphics mode are underlined.

```
1015 GO SUB 9000
1020 BORDER 5: PAPER 5: INK 0:
     CLS
1030 FOR n=1 TO 64
1040  LET P(n)=1
1050  PRINT BRIGHT 1; INK 2;
     AT V(n),H(n);"a"
1060 NEXT n
1070
1080 FOR n=20 TO 148 STEP 16
1090 PLOT 60,n: DRAW 128,0
1100 NEXT n
1110
1120 FOR n=60 TO 188 STEP 16

1130 PLOT n,20: DRAW 0,128
1140 NEXT n
1150
1160 PRINT AT 2,8;"A B C D E F G
 H"
1170
1180 PRINT AT 4,6;1;AT 6,6;2;AT
8,6;3;AT 10,6;4;AT 12,6;5;AT 14,
6;6;AT 16,6;7;AT 18,6;8
1190
1200 PRINT AT V(28),H(28);" ";AT
 V(29),H(29);" ";AT V(36),H(36);
" ";AT V(37),H(37);" "
1210 LET P(28)=0: LET P(29)=0:
     LET P(36)=0: LET P(37)=0
1220 LET moves=0: LET pegs=60
1230 PRINT PAPER 1; BRIGHT 1;
INK 7;AT 0,0;"MOVES=";moves;AT 0
,25;"PEGS=";pegs;AT 4,0; PAPER 6
; INK 0; BRIGHT 1;"INPUT";AT 5,0
;"99 TO";AT 6,0;"QUIT."

1240 PRINT PAPER 6; INK 0; BRIGH
T 1;AT 9,0;"ONLY ";AT 10,0;"bc d
e";AT 11,0;"fg hi"
2020 INPUT PAPER 0; INK 7; BRIGH
T 1;"MOVE FROM ? "; LINE F$
2025 IF F$="99" THEN GO TO 3560
2030 IF F$(1)<"1" OR F$(1)>"8"
     OR F$(2)<"A" OR F$(2)>"H"
     THEN BEEP 1,1: PRINT PAPER
2; INK 7; BRIGHT 1; FLASH 1;AT
20,0;  "ONLY NUMBER THEN LETTER
 e.g. 2A": FOR n=1 TO 250: NEXT
n: PRINT AT 20,0;"
                    ": GO TO 2020

                    ": GO TO 2020
2035 PRINT PAPER 1; BRIGHT 1;
INK 7; FLASH 1;AT 10,25;"FROM ";
F$
```

```
2040 INPUT PAPER 0; INK 7; BRIGH
T 1;"MOVE TO ? "; LINE T$
2045 IF T$="99" THEN GO TO 3560
2050 IF T$(1)<"1" OR T$(1)>"8"
     OR T$(2)<"A" OR T$(2)>"H"
     THEN BEEP 1,1: PRINT PAPER
2; INK 7; BRIGHT 1; FLASH 1;AT
20,0;  "ONLY NUMBER THEN LETTER
 e.g. 2A": FOR n=1 TO 250: NEXT
n: PRINT AT 20,0;"
                    ": GO TO 2040

2055 PRINT PAPER 1; BRIGHT 1;
INK 7; FLASH 1;AT 12,25;" TO  ";
T$
2060
2070 LET f=((CODE F$(1)-49)*8)+(
CODE F$(2)-64)
2080 LET t=((CODE T$(1)-49)*8)+(
CODE T$(2)-64)

2090 IF P(f)=0 OR P(t)=1 THEN
BEEP 1,1: PRINT PAPER 2; INK 7;
BRIGHT 1; FLASH 1;AT 20,6;  "
INVALID MOVE   ": FOR n=1 TO 250
: NEXT n: PRINT AT 20,6;"
           ";AT 10,25;"        ";
AT 12,25;"        ": GO TO 2020
2100
2110 LET diffv=V(f)-V(t)
2115 LET diffh=H(f)-H(t)
2120 IF diffh= 4 AND diffv=0
THEN GO TO 3100
2130 IF diffh=-4 AND diffv=0
THEN GO TO 3200
2140 IF diffv= 4 AND diffh=0
THEN GO TO 3300
2150 IF diffv=-4 AND diffh=0
THEN GO TO 3400

2160 BEEP 1,1: PRINT PAPER 2;
INK 7; BRIGHT 1; FLASH 1;AT 20,6
;  "   INVALID MOVE   ": FOR n=1
 TO 250: NEXT n: PRINT AT 20,6;"
                ";AT 10,25;"
  ";AT 12,25;"        ": GO TO
 2020
3120 BEEP 1,1
3125 IF P(t+1)=0 THEN GO TO 4000
3130 PRINT INK 2;AT V(t),H(t);"a
";AT V(t),H(t+1);" ";AT V(f),H(f
);" ";
3140 LET P(t)=1: LET P(t+1)=0:
     LET P(f)=0
3150 GO TO 3500
3220 BEEP 1,1
3225 IF P(t-1)=0 THEN GO TO 4000
3230 PRINT INK 2;AT V(t),H(t);"a
";AT V(t),H(t-1);" ";AT V(f),H(f
);" ";
3240 LET P(t)=1: LET P(t-1)=0:
     LET P(f)=0
3250 GO TO 3500
3320 BEEP 1,1
3325 IF P(t+8)=0 THEN GO TO 4000
3330 PRINT INK 2;AT V(t),H(t);"a
";AT V(t+8),H(t);" ";AT V(f),H(f
);" ";
3340 LET P(t)=1: LET P(t+8)=0:
     LET P(f)=0
3350 GO TO 3500
3420 BEEP 1,1
3425 IF P(t-8)=0 THEN GO TO 4000
3430 PRINT INK 2;AT V(t),H(t);"a
";AT V(t-8),H(t);" ";AT V(f),H(f
);" ";
3440 LET P(t)=1: LET P(t-8)=0:
     LET P(f)=0
3520 LET moves=moves+1
3530 LET  pegs= pegs-1
3540 PRINT PAPER 1; INK 7; BRIGH
T 1;AT 0,31;" ";AT 0,6;moves;AT
0,30;pegs
3550 IF pegs>1 THEN PRINT AT 10,
25;"        ";AT 12,25;"        ":
 GO TO 2020

3560 INPUT PAPER 0; INK 7; BRIGH
T 1; FLASH 1;"END OF GAME , ANOT
HER ? "; LINE A$: IF A$="" THEN
GO TO 3560
3565 IF A$="Y" THEN GO TO 1020
3570 STOP
3580
4000 REM     INVALID MOVE
4010
4020 PRINT PAPER 2; INK 7; BRIGH
T 1; FLASH 1;AT 20,8;" INVALID M
OVE ": FOR n=1 TO 250: NEXT n: P
RINT AT 20,8;"              ";AT
 10,25;"        ";AT 12,25;"
  ": GO TO 2020
4030
9000 REM      SET UP
9010
9020 DIM P(64)
9030 DIM V(64): DIM H(64)
9035 DIM F$(2): DIM T$(2)

9050 FOR n=0 TO 7
9060  READ a: POKE USR "A"+n,a
9070 NEXT n
9080 DATA 60,126,255,255,255,255
,126,60
9100 FOR n=1 TO 8
9110  LET V(n)   =4
9120  LET V(n+8) =6
9130  LET V(n+16)=8
9140  LET V(n+24)=10
9150  LET V(n+32)=12
9160  LET V(n+40)=14
9170  LET V(n+48)=16
9180  LET V(n+56)=18
9190 NEXT n
9210 FOR n=1 TO 57 STEP 8
9220  LET H(n)   =8
9230  LET H(n+1)=10
9240  LET H(n+2)=12

9250  LET H(n+3)=14
9260  LET H(n+4)=16
9270  LET H(n+5)=18
9280  LET H(n+6)=20
9290  LET H(n+7)=22
9300 NEXT n
9320 POKE 23658,8
9340 FOR n=0 TO 7
9350  READ b: POKE USR "B"+n,b
9360  READ c: POKE USR "C"+n,c
9370  READ d: POKE USR "D"+n,d
9380  READ e: POKE USR "E"+n,e
9390  READ f: POKE USR "F"+n,f
9400  READ g: POKE USR "G"+n,g
9410  READ h: POKE USR "H"+n,h
9420  READ i: POKE USR "I"+n,i
9430 NEXT n

9440 DATA 0,0,0,0,127,254,1,128,
0,0,1,128,48,12,1,128,0,0,3,192,
24,24,1,128,0,0,7,224,0,0,5,160,
0,0,5,160,0,0,7,224,24,24,1,128,
0,0,3,192,48,12,1,128,0,0,1,128,
127,254,1,128,0,0,1,0
9450
9460 RETURN
```



THIRTY ONES



WHEN START? is displayed enter M if you wish to play first or Y if you would prefer the computer to start. A set of counters will then be displayed, four of each number from one to six. You and the computer take turns in removing a number to add to the total. The winner is the player who brings the total to 31.

Written for the 16K ZX-81 by J H Entwistle of Cheltenham, Gloucestershire.

```
  8 CLS
  9 PRINT "START?"
 10 INPUT A$
 11 LET R=31
 12 LET M=7
 13 DIM D(6,4)
 14 LET T=0
 15 CLS
 19 FOR N=1 TO 6
 20 PRINT N*1111
 30 NEXT N
 35 PRINT AT 9,0;"TARGET= 31"
 40 IF A$(1)="Y" THEN GOTO 220
100 PRINT AT 18,0;"NUMBER?"
110 INPUT N
111 LET N=INT ABS N
112 IF NOT N OR N>M-1 THEN GOTO
110
113 PRINT AT 18,0;"        "
120 FOR A=1 TO 4
130 IF D(N,A)=0 THEN GOTO 145
140 NEXT A
145 PRINT AT N-1,A-1;"."
150 LET D(N,A)=1
160 LET R=R-N
164 PRINT AT 9,0;"TARGET= 31"
165 PRINT AT 10,0;"TOTAL=";31-R
166 IF R=0 THEN GOTO 340
167 IF R<0 THEN GOTO 350
170 FOR N=1 TO 4
180 IF D(M-1,N)=0 THEN GOTO 220
190 NEXT N
200 LET M=M-1
210 GOTO 170
220 LET G=R-M*INT (R/M)
230 IF R/M=INT (R/M) THEN LET G
=1+INT (RND*(M-1))
240 FOR N=1 TO 4
250 IF D(G,N)=0 THEN GOTO 300
260 NEXT N
270 LET G=1+INT (RND*(M-1))
280 GOTO 240
300 PRINT AT G-1,N-1;"."
310 LET R=R-G
315 PRINT AT 10,0;"TOTAL=";31-R
316 IF R=0 THEN GOTO 350
317 IF R<0 THEN GOTO 340
320 LET D(G,N)=1
330 GOTO 100
340 LET T=1
350 IF T=1 THEN PRINT AT 12,0;"
YOU WIN."
360 IF T=0 THEN PRINT AT 12,0;"
I WIN."
370 PRINT "AGAIN?"
380 INPUT A$
390 IF A$(1)="Y" THEN RUN
400 STOP
```

CASHFLOW

KAREN CRUICKSHANK of Guildford, Surrey has sent a program for the Spectrum, based on the workings of the Stock Exchange. You are given a working week in which to make as much money as possible by buying and selling stocks and shares. Each day you must decide eight times whether to buy or sell bonds.

Decisions you can make are based on the information given, including how much cash you have in hand and the market forecast. Beware—your boss could arrive at any minute and decide to alter your cashflow situation. It is an original game calling for planning and skill. Can you make sufficient money to be appointed manager at the end of the week?

```
   1 LET HS=0
   3 LET B=10: LET CA=1000: LET
P=10
   5 PAPER 0: INK  : BORDER 0: B
RIGHT 1: CLS
   6 FOR W=1 TO 5: IF W=1 THEN P
RINT "MONDAY"
   7 IF W=2 THEN PRINT "TUESDAY"
   8 IF W=3 THEN PRINT "WEDNESDA
Y"
   9 IF W=4 THEN PRINT "THURSDAY
"
  10 IF W=5 THEN PRINT "FRIDAY"
  14 PRINT FLASH 1'"A NEW WORKIN
G DAY"
  15 POKE 23658,8
  20 FOR Q=1 TO 12: BEEP 0.05,Q:
 NEXT Q
 120 FOR D=1 TO 8
 122 LET RN=INT (RND*10)
 123 GO SUB 2000
 125 PRINT ''"WORKING SEGMENT ";
D
 127 IF P<0 THEN LET P=0
 135 LET Q=INT (RND*5)
 136 PRINT PAPER 6; INK 0;''"REP
ORT:"'
 137 IF CA>=0 THEN PRINT "CASH A
T HAND £";CA
 139 IF CA<0 THEN PRINT "OVERDRA
FT £";CA
 140 PRINT B;" BONDS AT £";P;" E
ACH"
 145 PRINT "OVERALL ASSETS £";(P
*B)+CA
 150 PRINT "MARKET FORCAST ";
 160 IF Q=0 THEN PRINT "STEADY":
 LET P=P+(INT (RND*3)-2)
 170 IF Q=1 THEN PRINT "CHANGEAB
LE": LET P=P+(INT (RND*23)-10)
 180 IF Q=2 THEN PRINT "PLUMMETT
ING": LET P=P+(INT (RND*50)-40)
 190 IF Q=3 THEN PRINT "RISING":
 LET P=P+(INT (RND*50)-15)
 195 IF Q=4 THEN PRINT "UNCERTAI
N!!": LET P=P+(INT (RND*60)-30)
 200 INPUT "SELL OR BUY? ";A$
 210 BEEP 0.05,50
 220 IF A$="SELL" THEN GO TO 500
 230 IF A$<>"BUY" THEN GO TO 200
 240 LET NS=INT (RND*100)+25
 242 PRINT '"THERE ARE ";ns;" B
ONDS AVAILABLE"
 243 IF P<=0 THEN LET P=1
 245 PRINT "HOW MANY BONDS AT £"
P;" EACH"
 250 INPUT X
 255 BEEP 0.08,50
 257 IF X>NS THEN PRINT '"I SAID
 ";NS;" BONDS NOT ";X;" BONDS":
LET NS=NS-1: GO TO 242
 260 IF CA<CA-P*X THEN PRINT "NO
T ENOUGH MONEY": GO TO 240
 279 LET CA=CA-P*X
 290 LET B=B+X
 295 PRINT ''
 304 NEXT D
 306 GO TO 700
 510 LET BU=INT (RND*200)+50
 515 IF P<=0 THEN LET P=1
 520 PRINT '"MARKET PRICE £";P
 525 PRINT BU;" BUYERS WANTING B
UNDS"
 530 PRINT "HOW MANY BONDS TO SE
LL?"
 540 INPUT S
 545 BEEP 0.09,50
 546 IF S>BU THEN PRINT ''"I SAI
D ";BU;" BUYERS": LET BU=BU-1: G
O TO 525
 550 IF S>B THEN PRINT "YOU HAVE
N'T THAT MANY BONDS": GO TO 530
 555 IF S<0 THEN GO TO 525
 560 LET CA=P*S+CA
 595 LET B=B-S
 600 PRINT '': NEXT D
 700 FOR Q=-30 TO 30
 705 BEEP 0.005,Q
 706 NEXT Q
 710 IF W<>5 THEN PRINT ''''"THE
 END OF THE DAY"
 715 IF W=5 THEN PRINT '''''"THE
END OF THE WEEK"
 720 PRINT "CASH ";CA
 730 PRINT B;" BONDS AT £";P
 740 PRINT "TOTAL ASSETS £";: LE
T O=(P*B)+CA: PRINT O
 742 IF W=5 THEN GO TO 749
 745 IF INKEY$="" THEN GO TO 745
 747 CLS : NEXT w
 750 IF O<0 THEN PRINT " YOU'RE
FIRED !!!!!!"
 760 IF O>0 AND O<9000 THEN PRIN
T "YOU DO NOT USE ENOUGH INITIAT
IVE"
 765 IF O>9000 AND O<25000 THEN
PRINT "PROMOTION TO HEAD CASHIER
!!"
 770 IF O>25000 AND O<45000 THEN
 PRINT "PROMOTION TO CHIEF SUPER
VISOR"
 775 IF O>HS THEN LET HS=O
 780 IF O>45000 THEN PRINT "THE
MANAGERS SEAT AWAITS YOU"
 785 PRINT '"HIGH SCORE £";HS
 790 PRINT '''"          PRESS ANY
 KEY"
 800 IF INKEY$="" THEN GO TO 800
 810 CLS : GO TO 2
2010 IF RN=1 THEN PRINT ''"SUDDE
N DEVALUATION HAS MEANT": PRINT
"THAT TWO BONDS ARE BEING MERGED
": PRINT "INTO ONE.IE: CUT IN HA
LF"
2020 IF RN=1 THEN LET B=INT (B/2
)
2025 LET C=INT (RND*3)
2030 IF RN=2 THEN PRINT ''"BOSS
IS COMING AROUND CHECKING.": PAU
SE 100: IF C=0 THEN PRINT "HAD Y
OU WORRIED!!"
2040 IF RN=2 AND C=1 THEN PRINT
"HE CUTS OFF YOUR CASH SUPPLY BY
": PRINT "HALF. (HA,HA,HA)": LET
 CA=CA/2
2050 IF RN=2 AND C=2 THEN PRINT
"GIVES YOU EXTRA 100 BONDS.": LE
T B=B+100
2100 RETURN
```

```
  10 GOSUB 1000
  20 CLS
  30 LET W=0
  35 LET SH=1
  40 LET T=30
  45 LET S=0
  50 LET Z=3
  60 LET R=250
  65 FAST
  70 PRINT AT 1,0;"(32*is)"
  75 FOR L=2 TO 20
  80 PRINT "(is:30*sp:is)"
  85 NEXT L
  90 PRINT AT 20,0;"(32*is)"
  91 PRINT AT 0,0;".......... EN
ERGY"
  95 LET C=10
 100 LET B=10
 105 FOR A=1 TO T
 110 LET X=INT (RND*18)+2
 115 LET Y=INT (RND*29)+1
 120 PRINT AT X,Y;CHR$ 23
 125 NEXT A
 127 SLOW
 130 PRINT AT C,B;"(gh)"
 135 LET C=C+(INKEY$="6")-(INKEY
$="7")
 140 LET B=B+(INKEY$="8")-(INKEY
$="5")
 145 PRINT AT C,B;
 150 LET P=PEEK (PEEK 16398+256*
PEEK 16399)
 155 IF P=23 THEN GOTO 200
 160 IF P=128 THEN GOTO 220
 166 IF P=CODE "(gh)" THEN PRINT
 AT 0,W;" "
 167 IF P=CODE "(gh)" THEN LET W
=W+1
 168 IF W=10 THEN GOTO 230
 170 IF P=58 THEN GOTO 500
 180 PRINT AT C,B;"O"
 185 LET S=S+1
 190 IF S>R THEN PRINT AT 20,10;
"UUU"
 195 GOTO 130
 200 LET Z=Z-1
 205 PAUSE .5
 210 IF Z=0 THEN GOTO 220
 215 GOTO 130
 220 PRINT AT 2,1;"GAME OVER";TA
B 20;"SCORE=";S
 225 STOP
 230 PRINT AT 2,1;;"YOU RAN OUT
OF ENERGY";TAB 15;"SCORE=";S
 235 STOP
 500 CLS
 510 PRINT "YOU COMPLETED SHEET
";SH;TAB 20;"SCORE=";S
 515 PRINT "LIVES=";Z
 520 PRINT "MORE OBSTACLES WILL
NOW APPEAR"
 525 PAUSE 200
 530 CLS
 535 LET W=0
 540 LET T=T+30
 545 LET R=R+175
 547 IF SH>=3 THEN LET R=R+125
 550 LET SH=SH+1
 560 GOTO 65
 570 STOP
1000 PRINT "      MOLE"
1010 PRINT AT 2,0;"YOU STEER YOU
R TUNNELING MOLE,  AVOIDING THE"
"*""AND YOUR OWN    TUNNELS""(g
h)"".IF YOU HIT ONE OF THE""U""S
 YOU CAN GO ONTO ANOTHER    SCRE
EN."
1020 PRINT AT 7,0;"IF YOU HIT A"
"*""YOU WILL LOOSE A LIFE."
1030 PRINT AT 9,0;"IF YOU HIT TH
E BORDER YOU LOOSE ALL YOUR LIVE
S."
1040 PRINT AT 11,0;"IF YOU HIT Y
OUR OWN TUNNEL YOU  LOOSE SOME O
F YOUR ENERGY.     THE SAME IF
YOU STAY STILL."
1050 PRINT AT 15,0;"CURSOR KEYS
TO MOVE.";TAB 10;"PRESS A KEY."
1060 PAUSE 4E4
1070 RETURN
```

STEER your tunneling **Mole** through the ground. Hitting a stone will cause you to lose a life and hitting the border will kill you. If you remain still, or hit your tunnel, you will lose some energy. Eating a worm "V" will move you to another screen.

The program was written for the 16K ZX-81 by Simon Reeve of Chessington, Surrey.

Graphics instructions are given in lower-case letters within brackets. A space is represented by 'sp', a graphic character by 'g', and an inverse character by 'i'. Thus 'gh' represents graphic H.

More sensational software

ZX81

Breakout
Bomber
Scramble
Space Invaders
Paintmaze
Frogs
£3.95 EACH
Asteroids
Tempest
Rescue
1K Gamespack
£4.95 EACH
Mines of Saturn/Return to Earth
£5.95
ZX Chess
£6.50

Spectrum 16K

Scramble
Cosmic Raiders
Mines of Saturn/Return to Earth
Space Zombies
Panic
Galakzions
Tempest
Nanas
Creepy Crawler
£5.95 EACH

Spectrum 48K

Masterchess
Sorcerer's Castle
Mad Martha
Mad Marth II
Star Trek
Pat the Postman
Deffendar
Laserwarp
Paradroids
Knockout
Timequest
One Hundred & Eighty
SAS Assault
Land of Sagan
Cruise Attack
£6.95 EACH

Other great new Mikro-Gen programs coming soon – watch out for them!

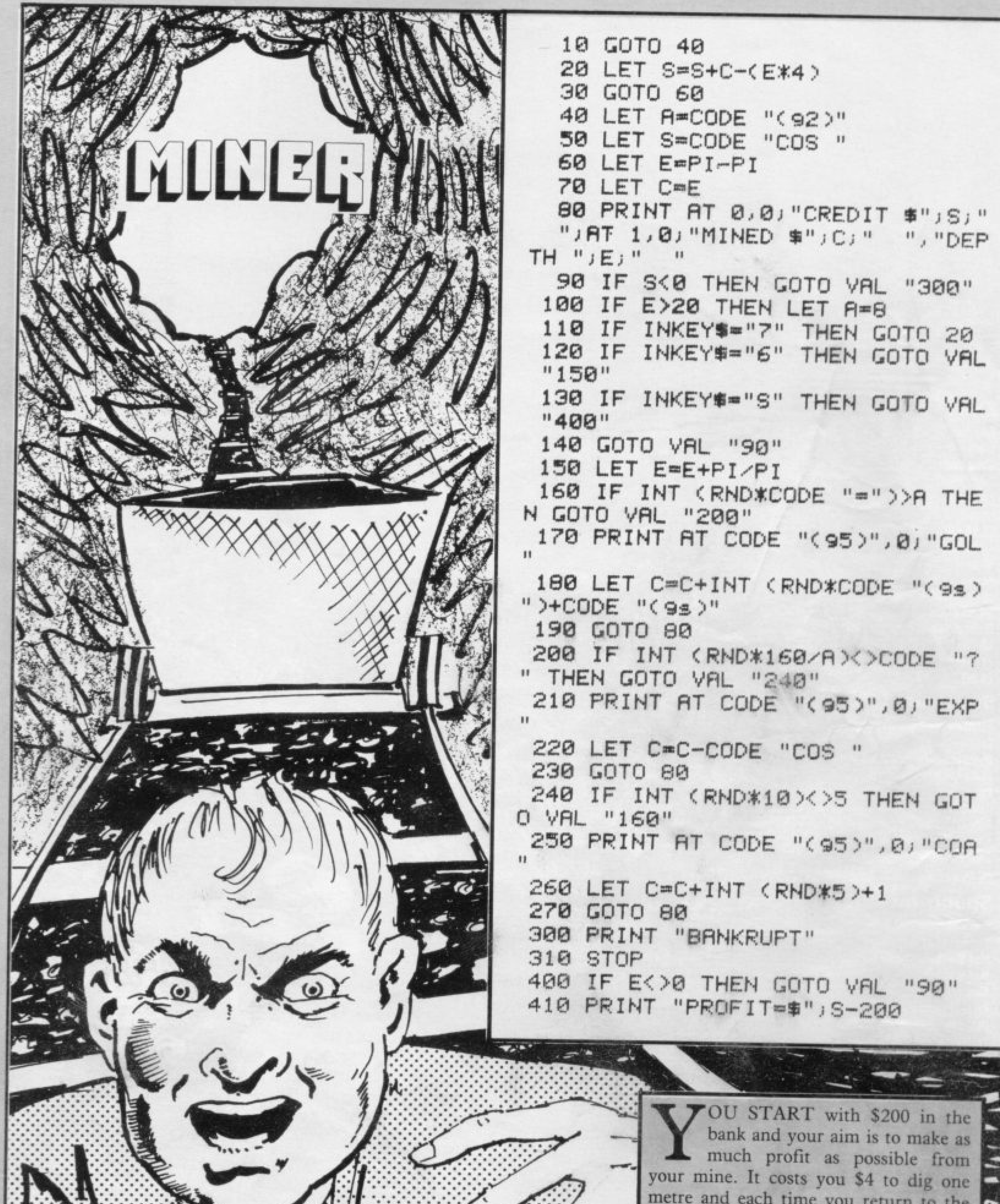
Please make cheques/PO payable to 'Mikro-Gen' and add 40p post & packing per order. Send mail orders direct to:

Available from Lightning, PCS, Tiger, W H Smith, Co-op, Menzies & other leading retailers.

PHONE YOUR VISA OR ACCESS NUMBER

MIKRO-GEN

Mikro-Gen, 1 Devonshire Cottages, London Rd, Bracknell RG12 2TQ Tel: 0344 27317



MINER

```
 10 GOTO 40
 20 LET S=S+C-(E*4)
 30 GOTO 60
 40 LET A=CODE "(92)"
 50 LET S=CODE "COS "
 60 LET E=PI-PI
 70 LET C=E
 80 PRINT AT 0,0;"CREDIT $";S;"
 ";AT 1,0;"MINED $";C;"  ","DEP
TH ";E;"  "
 90 IF S<0 THEN GOTO VAL "300"
100 IF E>20 THEN LET A=8
110 IF INKEY$="7" THEN GOTO 20
120 IF INKEY$="6" THEN GOTO VAL
"150"
130 IF INKEY$="S" THEN GOTO VAL
"400"
140 GOTO VAL "90"
150 LET E=E+PI/PI
160 IF INT (RND*CODE "=")>A THE
N GOTO VAL "200"
170 PRINT AT CODE "(95)",0;"GOL
"
180 LET C=C+INT (RND*CODE "(9s)
")+CODE "(9s)"
190 GOTO 80
200 IF INT (RND*160/A)<>CODE "?
" THEN GOTO VAL "240"
210 PRINT AT CODE "(95)",0;"EXP
"
220 LET C=C-CODE "COS "
230 GOTO 80
240 IF INT (RND*10)<>5 THEN GOT
O VAL "160"
250 PRINT AT CODE "(95)",0;"COA
"
260 LET C=C+INT (RND*5)+1
270 GOTO 80
300 PRINT "BANKRUPT"
310 STOP
400 IF E<>0 THEN GOTO VAL "90"
410 PRINT "PROFIT=$";S-200
```

YOU START with $200 in the bank and your aim is to make as much profit as possible from your mine. It costs you $4 to dig one metre and each time you return to the surface your funds are altered to take account of your profit and expenditure.

You may find coal (coa) or gold (gol), or you may be unlucky and be involved in an explosion, which will cost you $200. Key 6 moves you down the mine and key 7 returns you to the surface.

Mark Evans of Welling, Kent wrote **Miner** for the 1K ZX-81.

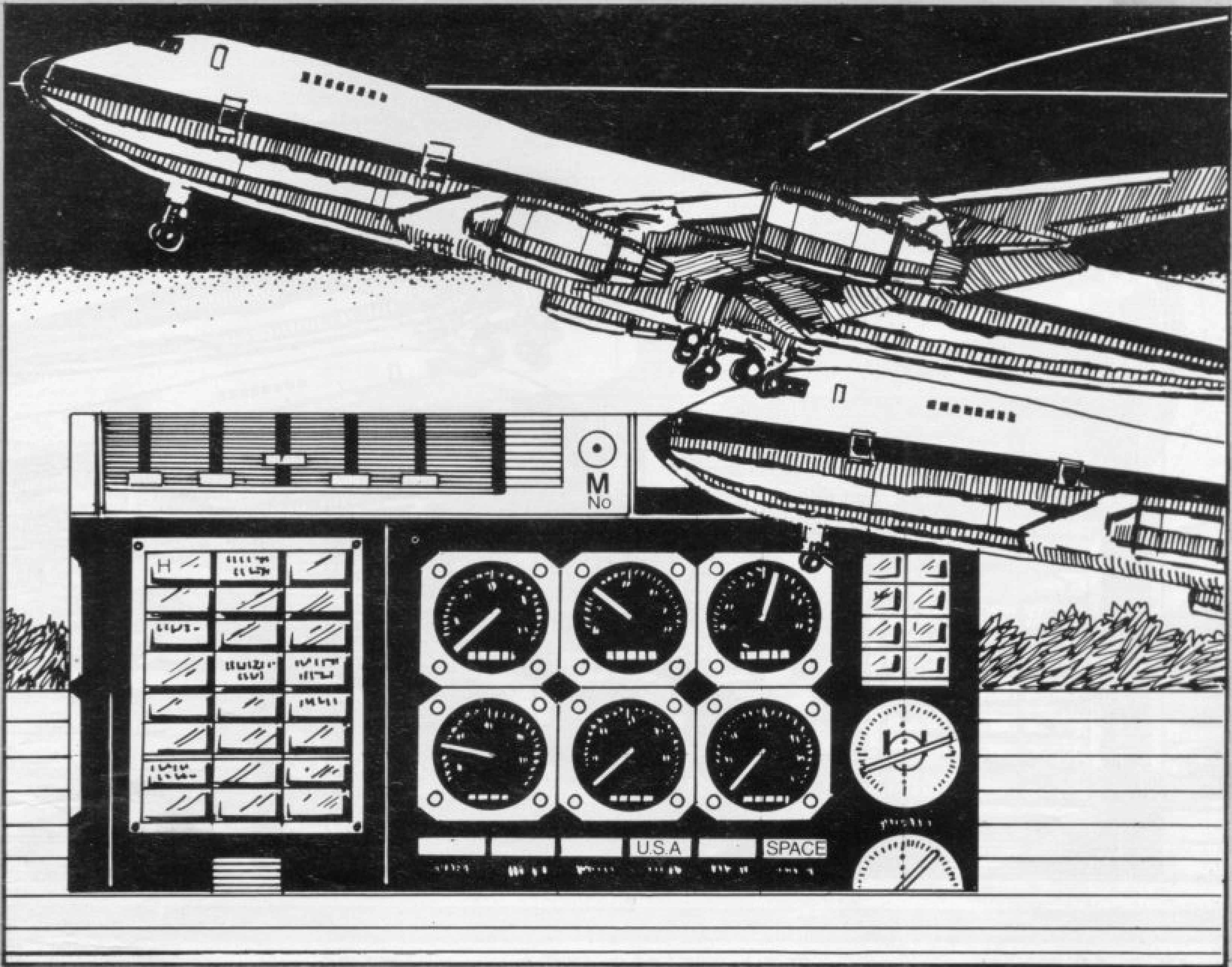
SKI RUN

PHILIP HARPER, aged nine, of Benfleet, Essex sent **Ski-Run** for the 16K Spectrum. Manipulate your skier round the flags using cursor keys 6 and 7. You have 10 lives in which to complete the ski run as many times as possible.

It is an ideal game for people who find that manipulation of characters by the use of four or eight keys is, as yet, beyond them.

4 Graphic S, graphic F
8 Graphic L

```
   1 REM "ski": CLS
   2 LET s=10: PRINT FLASH 1; IN
K 2; PAPER 6;AT 1,5;"PLEASE STOP
 THE TAPE": PRINT AT 2,3;"Press
any key to continue": PAUSE 0
   3 CLS : GO SUB 1000
   4 PRINT AT 1,13;"SKI","Ski is
 a game of skill and it is";"als
o very addictive.          ";"
Use keys 6 & 7 to steer your
";"skier,(k),round the flags,(⚑)
.";"   You have got 10 lives!":
INK 1: PRINT : PRINT : PRINT "
Press any key to continue": PAUS
E 0
   5 CLS : PRINT AT 10,0;"Skiers
 left";: FOR f=s TO 0 STEP -1: P
RINT INK 1;"k ";: NEXT f: IF s=-
1 THEN GO TO 5000
   6 LET a=1: BORDER 5: PAPER 7:
 INK 2
   7 PRINT BRIGHT 1; PAPER 7; IN
K 2;AT 1,0;"             ⚑           ⚑
     ⚑        ",AT 2,0; BRIGHT 1;
PAPER 7; INK 2;"             ⚑           ⚑
     ⚑        ⚑"
   8 PRINT AT 3,0; INK 0; PAPER
6;"______________________________
____",AT 4,0; PAPER 7;"
        ";AT 0,0; INK 0; PAPER 6
;"_______________________________
___"
   9 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
  10 FOR f=1 TO 31
  20 PRINT BRIGHT 1; PAPER 7; IN
K 1;AT a,f;"k"
  21 IF (a=1) AND ((f=8) OR (f=1
8) OR (f=27)) THEN GO TO 3000
  22 IF (a=2) AND ((f=4) OR (f=1
3) OR (f=22) OR (f=31)) THEN GO
TO 3000
  25 BEEP .002,f: PRINT AT a,f;
BRIGHT 1; PAPER 7;" "
  26 LET a=a+(INKEY$="6" AND a<2
)-(INKEY$="7" AND a>1)
  30 NEXT f
  35 LET g=g+1
  40 PRINT FLASH 1; PAPER 1; INK
 7;AT 5,10;"WELL DONE!!"
  50 BEEP .1,5: PAUSE 5: BEEP .1
,6: PAUSE 5: BEEP .1,5: PAUSE 5:
 BEEP .1,6: BEEP .9,8
  60 GO TO 5
1000 LET g=0: FOR f=0 TO 7: READ
 z: POKE USR "s"+f,z: NEXT f
1010 FOR f=0 TO 7: READ z: POKE
USR "f"+f,z: NEXT f
1020 FOR f=0 TO 7: READ z: POKE
USR "l"+f,z: NEXT f
1040 FOR f=0 TO 7: READ z: POKE
USR "e"+f,z: NEXT f
1090 RETURN
2000 DATA BIN 00011000,BIN 00011
000,BIN 00010000,BIN 00011111,BI
N 00011010,BIN 00011010,BIN 0001
0001,BIN 11111111
2010 DATA BIN 10000000,BIN 11101
111,BIN 11111110,BIN 11111100,BI
N 11111110,BIN 11111111,BIN 1000
0000,BIN 10000000
2020 DATA 255,0.0,0.0,0.0,255
2040 DATA 0,0,0,BIN 01001001,BIN
 00101010,0,BIN 00101010,BIN 010
01001
3000 FOR q=0 TO 7: PRINT BRIGHT
1; PAPER 7; FLASH 1; INK INT (RN
D*7);AT a,f;"※"
3010 BEEP .1,-45: NEXT q
3015 LET s=s-1
3016 LET g=g+1
3020 GO TO 60
4000 STOP
5060 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
 PRINT AT 6,7;"Want another go?"
,AT 7,13;"(y/n)"
5065 PRINT AT 1,1,"You finished
with ";g;" runs"
5070 IF INKEY$="y" THEN CLS : LE
T s=10: GO TO 5
5080 IF INKEY$="n" THEN CLS : GO
 TO 4000
5090 GO TO 5070
```



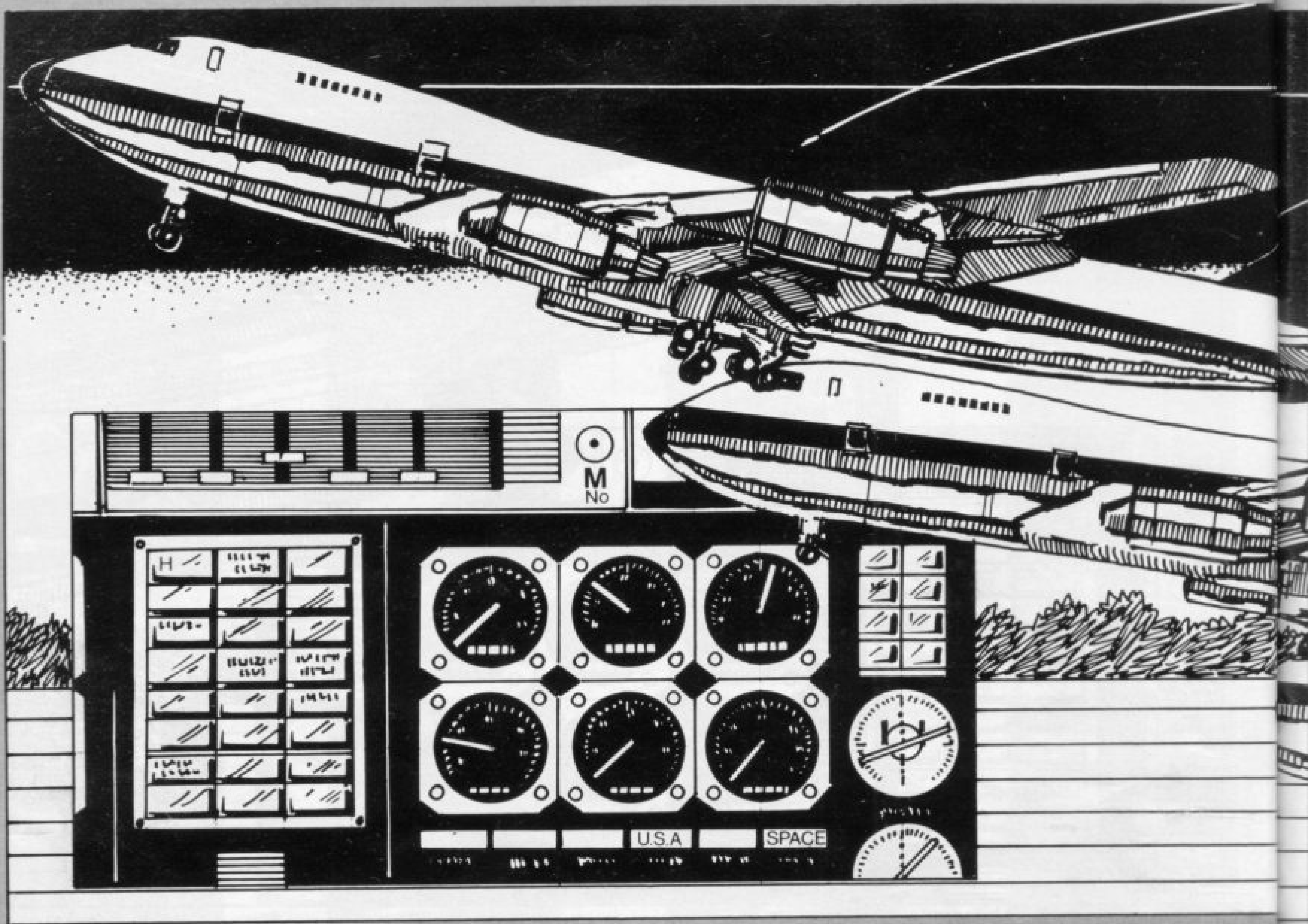
AIRPLANE

AIRPLANE simulates an aeroplane flight from London to one of 10 airports round the world. The screen displays your control panel, complete with information about speed and course.

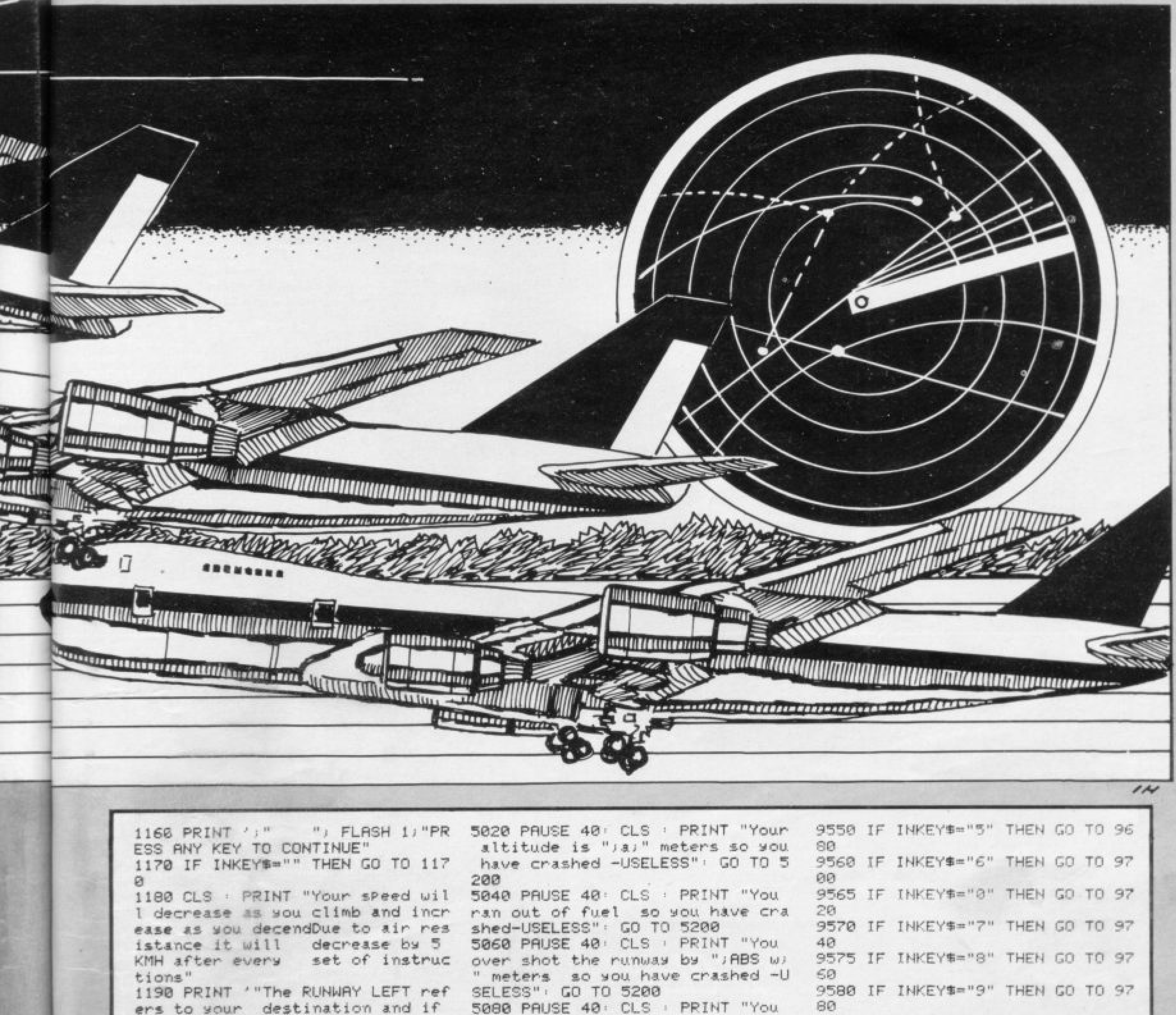
You are given the opportunity to change your speed, height and course. If you choose not to alter any of them, enter "0". Keep your changes within the defined limits and look carefully at the readings displayed before making a change. You have no second chances and a slight mistake will lead to a crash.

The program was written for the 16K Spectrum by David Courtier-Dutton of Bradfield, Berkshire.

```
   1 LET zx=0: LET x=0: LET y=0:
 LET z=0
  10 GO SUB 1020
  11 GO SUB 9500
  15 LET m=l: LET tr=w: BORDER 6
  PAPER 7: INK 2
  20 PLOT 89,145: DRAW INK 3;7,0
 : PLOT (89+INT l/50),145: DRAW I
NK 3;INT w/50,0
  30 PRINT AT 17,15;"DESTINATION
"
 100 LET a=0: LET s=0: LET d=0:
LET r=0: LET c=0: LET f=29: LET
n=0
 200 OVER 1: INK 2: PLOT 0,0: DR
AW 255,0: DRAW 0,175: DRAW -255,
0: DRAW 0,-175
 210 PLOT 0,16: DRAW 255,0: PRIN
T AT 20,0; INK 1;"FUEL"
 220 PLOT 32,0: DRAW 0,16: PLOT
32,8: DRAW 223,0
 230 PRINT AT 20,4; INK 3;"0  50
  100  150  200  250 300"
 240 PLOT 88,16: DRAW 0,159: PLO
T 0,144: DRAW 255,0
 250 PRINT AT 14,1; INK 1;"ALTIT
UDE"
 251 PRINT AT 10,3; INK 1;"RANGE
"
 252 PRINT AT 2,2; INK 1;"SPEED"
 253 PRINT AT 6,1; INK 1;"TAIL F
LAP"
 254 PRINT AT 0,0; INK 3;"INSTRU
MENTS"
 255 PRINT AT 12,0; INK 1;"RUNWA
Y LEFT"
 256 PRINT AT 18,0; INK 1;"FLAP
ANGLE"
 257 PRINT AT 8,1; INK 1;"R/BEAR
ING"
 258 PRINT AT 4,0; INK 1;"ACCELE
RATE"
 259 PRINT AT 16,1; INK 1;"DROP
RATE"
 310 PLOT 89,95: DRAW 166,0: PLO
T 0,160: DRAW 88,0: PLOT 89,112:
 DRAW 166,0
 340 FOR t=32 TO 128 STEP 16: PL
OT 0,t: DRAW 86,0: NEXT t
 390 PRINT AT 21,4;"
                    ": OVER 0: GO SUB
 700
 520 IF s<150 AND a>0 THEN GO TO
 5000
 540 PRINT AT 5,13;"Increase/Dec
rease";AT 6,13;"thrust +75/-75?"
 560 INPUT x: IF x>75 THEN LET x
=75
 565 IF x<-75 THEN LET x=-75
 570 LET s=s+x: LET c=x: LET x=0
 : IF s>600 THEN LET s=600
 572 LET s=s-5: IF s<0 THEN LET
s=0
 573 LET l=l-INT (1.25*(s*(1-d/1
00)))
```




```
575 GO SUB 700
578 IF s<150 AND a>0 THEN GO TO
5000
580 LET x=0: PRINT AT 5,13;"Cha
nge flap angle?";AT 6,13;"50/-50
      ": INPUT y: LET n=n+y:
IF n>50 THEN LET n=50
581 IF n<-50 THEN LET n=-50
582 LET a=a+INT (3.06*n): IF a<
-5 THEN GO TO 5020
583 IF a>=500 THEN LET a=500
585 IF a<4 AND l>100 THEN LET a
=0
590 LET r=INT (3.06*n): LET s=s
-n
595 IF a<0 AND a>=-5 THEN LET a
=0
600 GO SUB 700
605 LET zx=1

610 PRINT AT 5,13;"Alter tail f
lap?  ";AT 6,13;"+45/-45": INPUT
z: LET d=d+z: IF pe>179 THEN LE
T pe=-179
612 IF pe<-179 THEN LET pe=179
620 LET pe=pe-d
630 IF l<=0 THEN LET w=w-s
635 IF f<=0 THEN GO TO 5040
640 IF a<0 AND l>0 THEN GO TO 5
080
645 IF a<=0 AND pe<>0 AND l<=25
0 THEN GO TO 5100
660 LET f=f-INT ((n/10+s/20)/ (
e/2)): IF f<=0 THEN GO TO 5040
670 IF l>400 AND l<500 AND a>5
AND a<50 THEN LET a=100
675 IF s>230 AND a<=0 THEN LET
a=50
680 IF w<=0 THEN GO TO 5060

700 PRINT AT 3,1; INK 0;s;"  ";
AT 15,1;a;"  ";AT 11,1;l;"  ";AT
 7,1;d;"  ";AT 19,1;n;"  ";AT 5,
1;c;"  ";AT 17,1;r;"  ";AT 9,1;p
e;"  ";AT 13,1;w;"  "
810 FOR t=16 TO 144 STEP 16: PL
OT 0,t: DRAW 86,0: NEXT t
860 PRINT AT 8,12; INK 4;"  -
R/BEARING  +"
880 PRINT AT 9,12;"
     "
890 PRINT AT 9,21;"o": PRINT AT
9,21-INT (pe/20);"^"
892 IF zx<>0 THEN GO TO 900
895 RETURN
900 PLOT INK 0;INT ((m-l)/50)+8
9,INT a/18+146
905 LET zx=0
910 PRINT AT 21,f;"   "
920 IF a<=0 AND w>=0 AND f>0 AN
D pe=0 AND l<0 AND s<=0 AND S>=-
1 THEN GO TO 6000


935 IF a<=40 THEN PRINT AT 16,1
5; INK 2; PAPER 7; FLASH 1;"ALTI
TUDE"
940 IF s<165 THEN PRINT AT 14,1
5; INK 2; PAPER 7; FLASH 1;"SPEE
D"
945 IF l<100 THEN PRINT AT 12,1
5; INK 2; PAPER 7; FLASH 1;"RANG
E"
950 IF l<300 AND pe<>0 THEN PRI
NT AT 13,15; INK 2; PAPER 7; FLA
SH 1;"R/BEARING"
955 IF w<200 THEN PRINT AT 12,1
5; INK 2; PAPER 7; FLASH 1;"RUNW
AY LEFT"
960 FOR q=1 TO 100: BEEP .007,-
22: NEXT q: FLASH 0: PRINT AT 12
,15;"          ";AT 13,15;"
          ";AT 14,15;"
     ";AT 15,15;"          ";
AT 16,15;"          "
1000 GO TO 500
1020 PAPER 0: INK 5: BORDER 2: C
LS
1030 PRINT AT 10,10;"INSTRUCTION
S? [y/n]"
1035 IF INKEY$="n" THEN RETURN
1036 IF INKEY$="" THEN GO TO 103
5
1037 CLS
1040 BEEP .3,0: BEEP .3,0: BEEP
.3,0: BEEP .8,-5
1050 BEEP .3,3: BEEP .3,3: BEEP
.3,3: BEEP .8,-2
1060 FOR m=-7 TO 5: BEEP .1,m: N
EXT m: PAUSE 20: BEEP .17,15: PA
USE 15: BEEP .4,-25

1070 PRINT AT 10,12; INK 2;"AIRP
LANE": PAUSE 150: CLS
1080 PRINT " AIRPLANE simulates
an airoplaneflight from London a
irport to  one of ten other air
ports aroundthe world,the choice
 of which isyours.However flying
 a plane is not as easy as you m
ay think,so read these instructi
ons        carefully"
1100 PRINT ''';"    "; FLASH 1;"
PRESS ANY KEY TO CONTINUE"
1120 IF INKEY$="" THEN GO TO 112
0
1130 CLS : PRINT ; FLASH 1;"BEAR
 IN MIND"; FLASH 0;'"You will st
all at under 150 KMH so do not t
ake off before this  speed is at
tained"

1140 PRINT '"To change altitude
a positive   (up) or negative (d
own) value   must be applied to
the flaps;   your altitude will
change by    three times the fla
p angle"
1150 PRINT '"To alter course use
 your TAIL   FLAP,when the runwa
y bearing    [R/BEARING] is 0 yo
u will be on"
1155 PRINT "course.In the R/BEAR
ING display your destination is
represented by 'o' and your actu
al course by"
1158 PRINT "'^';when the 'o' dis
sappears youwill be on course"
```



```
1160 PRINT ';"     "; FLASH 1;"PR
ESS ANY KEY TO CONTINUE"
1170 IF INKEY$="" THEN GO TO 117
0
1180 CLS : PRINT "Your speed wil
l decrease as you climb and incr
ease as you decendDue to air res
istance it will    decrease by 5
KMH after every    set of instruc
tions"
1190 PRINT '"The RUNWAY LEFT ref
ers to your  destination and if
this value isless than 0 you wil
l crash."
1200 PRINT '"Your Position relat
ive to Londonand your destinatio
n is Ploted  at the top of the s
ceen to show your Position at a
glance.The    distance you are fr
om your       destination is show'
n in the      RANGE readout"

1220 PRINT ;''"     "; FLASH 1;"P
RESS ANY KEY TO CONTINUE"
1230 IF INKEY$="" THEN GO TO 123
0
1235 CLS : PRINT "To land succes
sfully your speed must equal 0,y
our altitude must equal 0,and yo
u MUST be on the  runway"
1240 PRINT '"Flashing instructio
ns will help you to fly the Plan
e safely.               GOOD LUC
K!"
1250 PRINT ; INK 6;;''"TO REPEAT
 INSTRUCTIONS PRESS 'R'  PRESS A
NY OTHER KEY TO BEGIN"
1270 IF INKEY$="r" THEN CLS : GO
 TO 1080
1280 IF INKEY$="" THEN GO TO 127
0
4000 RETURN

5000 PAUSE 40: CLS : PRINT "You
stalled at ";s;" K/M/H and      c
rashed -USELESS": GO TO 5200
5020 PAUSE 40: CLS : PRINT "Your
 altitude is ";a;" meters so you
 have crashed -USELESS": GO TO 5
200
5040 PAUSE 40: CLS : PRINT "You
ran out of fuel  so you have cra
shed-USELESS": GO TO 5200
5060 PAUSE 40: CLS : PRINT "You
over shot the runway by ";ABS w;
" meters  so you have crashed -U
SELESS": GO TO 5200
5080 PAUSE 40: CLS : PRINT "You
tried to land ";l;" miles short
of the   runway   so you have cr
ashed-USELESS": GO TO 5200
5100 PAUSE 40: CLS : PRINT "You
have missed the runway completel
y by ";pe;" degrees and have cra
shed -USELESS": GO TO 5200
5200 PAUSE 300: CLS : PRINT "Ano
ther flight?(y/n)": IF INKEY$="y
" THEN GO TO 1: STOP
5300 GO TO 1
6000 LET k=INT f/3
6092 LET sc=INT (110*(w/tr)): LE
T k=(2*k)+sc
6100 PAUSE 100: CLS : PRINT "Wel
l done!You scored ";INT k;" Poin
ts out of a Possible 100": STOP
8100 STOP
9500 INK 7: BORDER 2: PAPER 0: C
LS : PRINT "WHICH AIRPORT?": PRI
NT "0)Istanbul"'"1)Chicago"'"2)M
ilan"'"3)Moscow"'"4)New York"'"5
)Port Stanley"'"6)Oslo"'"7)Tel A
viv"'"8)Delhi"'"9)Toronto"

9505 INK 2: PAPER 7
9510 IF INKEY$="1" THEN GO TO 96
00
9520 IF INKEY$="2" THEN GO TO 96
20
9530 IF INKEY$="3" THEN GO TO 96
40
9540 IF INKEY$="4" THEN GO TO 96
60
9550 IF INKEY$="5" THEN GO TO 96
80
9560 IF INKEY$="6" THEN GO TO 97
00
9565 IF INKEY$="0" THEN GO TO 97
20
9570 IF INKEY$="7" THEN GO TO 97
40
9575 IF INKEY$="8" THEN GO TO 97
60
9580 IF INKEY$="9" THEN GO TO 97
80
9585 GO TO 9510
9600 CLS : LET l=4235: LET e=15:
 LET w=700: LET pe=170: PRINT AT
 18,16;"CHICAGO": RETURN
9620 CLS : LET l=581: LET e=4: L
ET w=700: LET pe=35: PRINT AT 18
,16;"MILAN": RETURN

9640 CLS : LET l=1549: LET e=9:
LET w=640: LET pe=-10: PRINT AT
18,16;"MOSCOW": RETURN
9660 CLS : LET l=3500 : LET e=13
: LET w=750: LET pe=170: PRINT A
T 18,16;"NEW YORK": RETURN

9680 CLS : LET l=7406 : LET e=24
: LET w=440: LET pe=110: PRINT A
T 18,16;"PORT STANLEY": RETURN
9700 CLS : LET l=722 : LET e=5:
LET w=500: LET pe=-30: PRINT AT
18,16;"OSLO": RETURN

9720 CLS : LET l=1562 : LET e=9:
 LET w=480: LET pe=35: PRINT AT
18,16;"ISTANBUL": RETURN

9740 CLS : LET l=2230 : LET e=11
: LET w=650: LET pe=40: PRINT AT
 18,16;"TEL AVIV": RETURN

9760 CLS : LET l=5203 : LET e=18
: LET w=510: LET pe=34: PRINT AT
 18,16;"DELHI": RETURN
9780 CLS : LET l=3728 : LET e=14
: LET w=550: LET pe=-150: PRINT
AT 18,16;"TORONTO": RETURN
```

Everyone wants to get their hands on The Fuller Gift Pack.

The Fuller gift pack has got to be the computer bargain of the decade; no wonder everyone wants to get their hands on one.

This super gift pack contains the fantastic Fuller Master Unit, the ultimate unit that provides all the requirements for the serious games enthusiast. The Fuller Master Unit incorporates both the Fuller Box and Orator; its amazing features include a speech synthesiser, an independent sound synthesiser, a joystick interface, an amplifier for standard Spectrum sound, a cassette interface and a hi-fi socket. The pack will also include a Master Unit demo tape and full documentation.

As well as the Master Unit this Gift Pack also contains a de-luxe Joystick, with suction adhesion, a firm smooth action and a moulded hand grip incorporating twin firing buttons.

Once your Spectrum has been added to the Fuller Gift Pack you will have an extended computer system that is second to none.

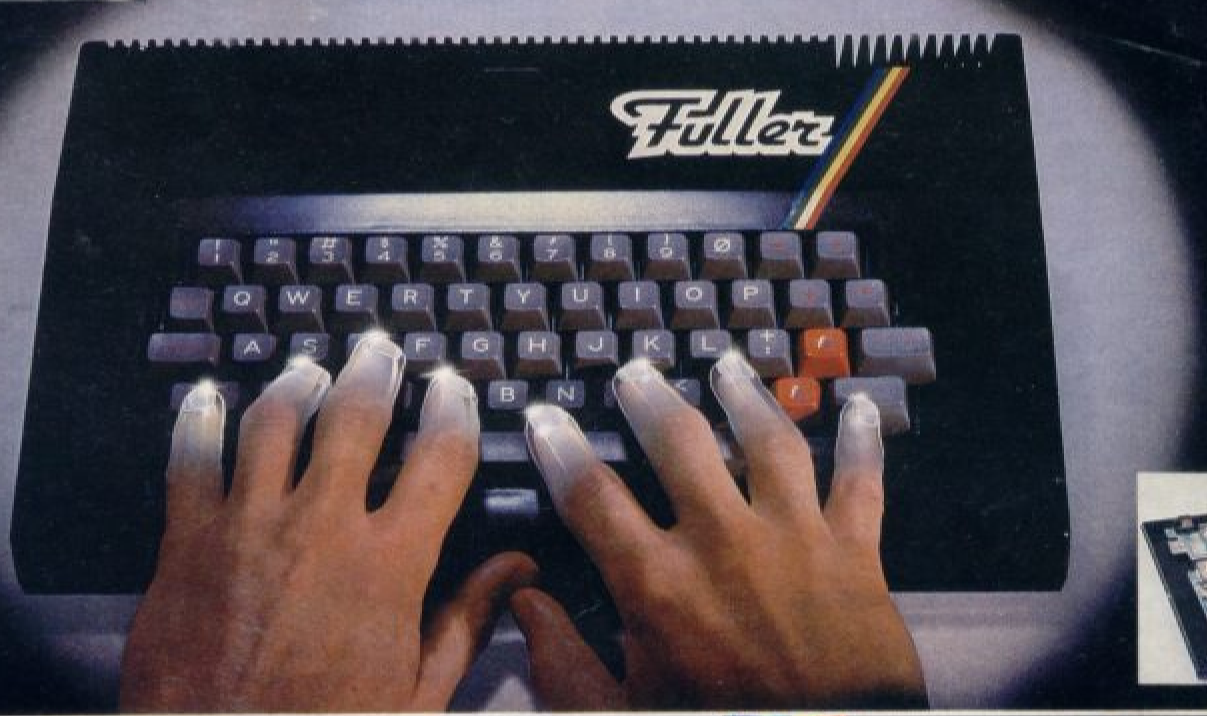
Now you can pit your mind against the four unique, exciting computer games included in the pack. These games have been especially adapted for this superb system and have added arcade sound effects and speech to bring out the very best in your Spectrum.

The Fuller Gift Pack saves you £20 on the purchase price of buying these items separately.

£79.95
inclusive of post and packaging

Power to your Fingertips.

All keys on the FDS keyboard contain all the Graphic Characters of the ZX Spectrum



Transform your Spectrum with the FDS Keyboard and feel the newly acquired power flow from your fingertips as you enter data with lightning fast accuracy.

The Fuller FDS Keyboard has been especially designed for the professional user. This elegant, slim line, desk top system has a keyboard that inclines toward you and contains all the graphic characters of the ZX Spectrum, plus, additional function keys, shift keys on either side and a space bar. With all the entry keys double size, lightning fast, accurate data entry is now made easy. Other amazing features of the FDS include:–

Gold plated key switch contacts, guaranteed for 10^6 operations. 4 cursor control keys, 2 functions for direct entry into E mode.

Based on the popular FD42 system, the Fuller FDS is simple to install; it requres no soldering or technical knowledge. For the user who is reluctant to install his computer circuit board inside the FDS, a buffer is available (£8.75 + 80p&p) which simply plugs into the expansion port and connects directly to the keyboard.

£49.95

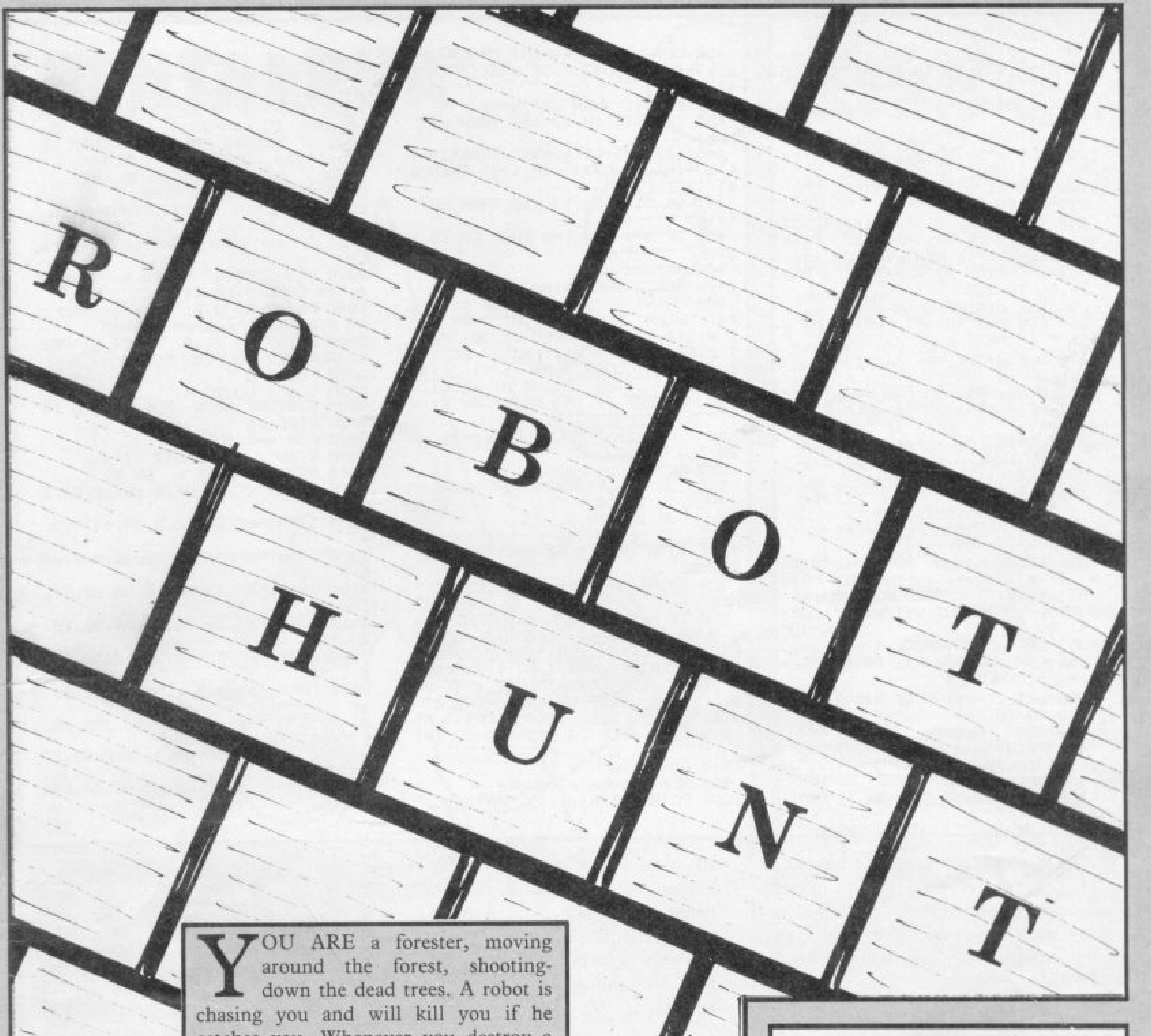
Microdrive interface 1 easily adapted to fit inside the case

Fuller MICRO SYSTEMS

Fuller Micro Systems
The ZX Centre, 71 Dale Street,
Liverpool 2. Tel: 051-236 6109.

At Fuller Micro Systems we strive to maintain the highest quality in both research and development, ensuring our products are the very best. We have now built a brand new mail order department to ensure our service is also the very best. Our stock includes a full range of components and kits that will make the most of your Spectrum, all our products enjoy an excellent reputation for reliability and carry a world wide one year guarantee.

Please supply me with............ FDS Keyboard(s).
I enclose a cheque/PO payable to .... Fuller Gift Pack(s).
Fuller Micro Systems for ....or debit my credit card.
Card No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ ☐ Barclaycard
☐ Access Card
Signature ....
Name ....
Address ....
....
Fuller Micro Systems, The ZX Centre,
71 Dale Street, Liverpool 2. Tel: 051-236 6109



YOU ARE a forester, moving around the forest, shooting-down the dead trees. A robot is chasing you and will kill you if he catches you. Whenever you destroy a tree several carnivorous trees will spring up in its place and they will kill you if you shoot them. Move yourself with I, J, K and M and direct your gun with A, W, S and Z.

Robot Hunt was written for the Spectrum by S Hirst of Hull, North Humberside.

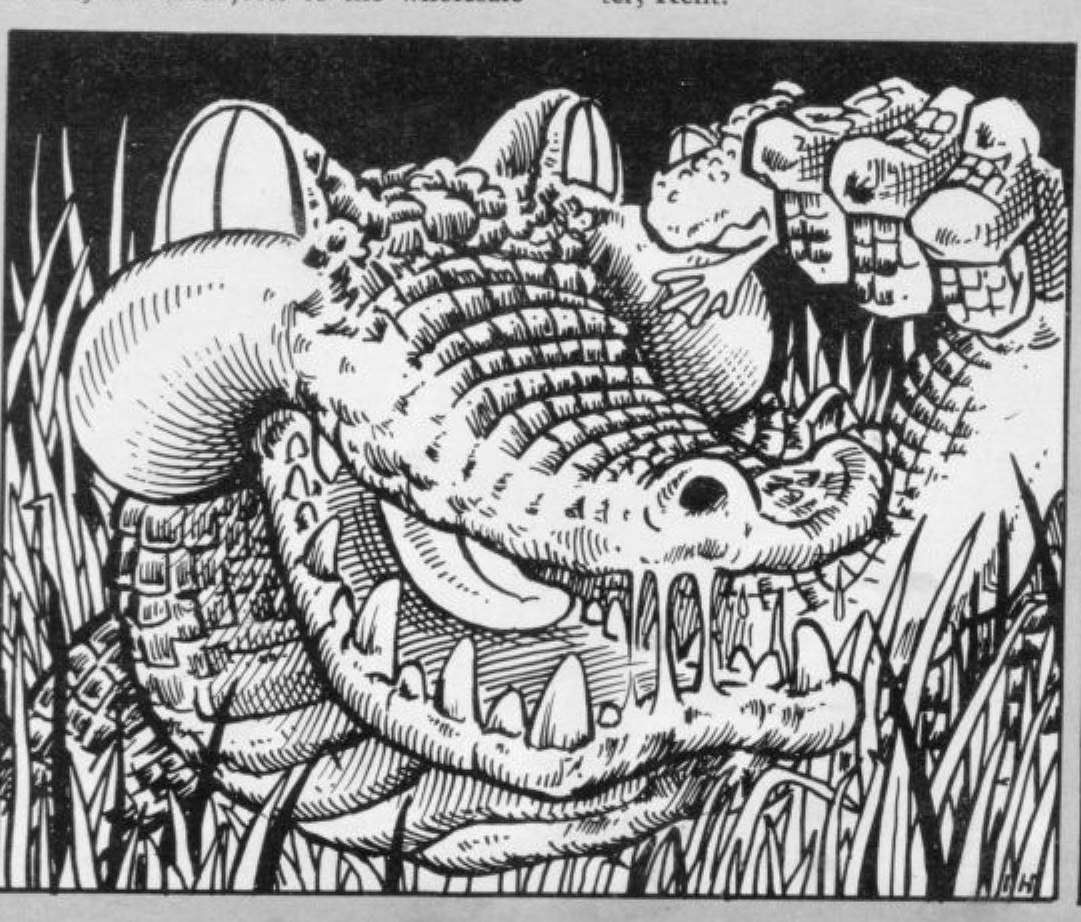
```
  1 REM    ** ROBOT HUNT **
  3 LET hscr=0
  4 LET hit=0
  5 LET man=5: LET m$="ccccc"
 10 LET x=15: LET y=10: LET a=1
5: LET b=10
 11 LET scr=0: LET mo=0
 15 BORDER 5
 17 CLS
 20 PRINT AT 0,0;"ccccc   ROBOT
 HUNT   scr - 00000"
 30 FOR i=1 TO 20: BEEP .1,RND*
30
 31 LET ty=RND*20+1: LET tx=RND
*31: IF ATTR (ty,tx)<>56 THEN GO
 TO 51
 32 PRINT AT ty,tx; INK 4;"i":
NEXT i
 35 GO TO 55
 50 FOR i=1 TO 21: BEEP .1,RND*
30
 51 LET ty=RND*20+1: LET tx=RND
*31: IF ATTR (ty,tx)<>56 THEN GO
 TO 51
 52 PRINT AT ty,tx; INK 4;"i":
NEXT i
 55 LET tots=0
 80 LET h=INT (RND*20)+1: LET l
=INT (RND*31): IF ATTR (h,l)<>56
 THEN GO TO 80
 81 LET hi=h: LET li=l: LET mo=
0
 82 PRINT AT h,l;" ";AT hi,li;"
"
 90 PRINT AT 0,27;"(5*isp)";AT
0,27; OVER 1;scr
 99 REM   man control
100 LET x=x+(INKEY$="k")*(x<31)
-(INKEY$="j")*(x>0)
110 LET y=y+(INKEY$="m")*(y<21)
-(INKEY$="i")*(y>1)
113 IF ATTR (y,x)=58 THEN LET h
it=58: LET x=a: LET y=b: GO TO 5
30
115 IF ATTR (y,x)<>56 THEN LET
x=a: LET y=b
120 PRINT AT y,x;"c"
123 IF x<a THEN PRINT AT b,a;"
";AT y,x;"b": BEEP .05,23: PRINT
 AT y,x;"a": BEEP .05,10
125 IF x>a THEN PRINT AT b,a;"
";AT y,x;"e": BEEP .05,23: PRINT
 AT y,x;"d": BEEP .05,10
130 IF b<>y THEN PRINT AT b,a;"
";AT y,x;"f": BEEP .05,23: PRIN
T AT y,x;"g": BEEP .05,10
135 LET a=x: LET b=y
140 LET a$=INKEY$: IF a$="w" OR
a$="s" OR a$="a" OR a$="z" THEN
GO TO 150
143 GO SUB 400
145 GO TO 100
150 IF a$="a" THEN GO TO 200
155 IF a$="s" THEN GO TO 250
160 IF a$="w" THEN GO TO 300
165 IF a$="z" THEN GO TO 350
170 LET a$="": GO TO 100
199 REM   shoot
```

```
 200 BEEP .01,-5: FOR i=1 TO x: IF ATTR (y,x-i)=56 THEN PRINT AT y,x-i;"-": NEXT i: FOR i=1 TO x: PRINT AT y,x-i;" ": NEXT i: GO TO 100
 203 LET hit=ATTR (y,x-i)
 205 FOR n=1 TO i-1: PRINT AT y,x-n;" ": NEXT n
 208 LET x1=x-i: LET y1=y
 210 GO TO 500
 250 BEEP .01,-5: FOR i=1 TO 31-x: IF ATTR (y,x+i)=56 THEN PRINT AT y,x+i;"-": NEXT i: FOR i=1 TO 31-x: PRINT AT y,x+i;" ": NEXT i: GO TO 100
 253 LET hit=ATTR (y,x+i)
 255 FOR n=1 TO i-1: PRINT AT y,x+n;" ": NEXT n
 258 LET y1=y: LET x1=x+i
 260 GO TO 500
 300 BEEP .01,-5: FOR i=1 TO y-1: IF ATTR (y-i,x)=56 THEN PRINT AT y-i,x;"!": NEXT i: FOR i=1 TO y-1: PRINT AT y-i,x;" ": NEXT i: GO TO 100
 303 LET hit=ATTR (y-i,x)
 305 FOR n=1 TO i-1: PRINT AT y-n,x;" ": NEXT n
 308 LET y1=y-i: LET x1=x
 310 GO TO 500
 350 BEEP .01,-5: FOR i=1 TO 21-y: IF ATTR (y+i,x)=56 THEN PRINT AT y+i,x;"!": NEXT i: FOR i=1 TO 21-y: PRINT AT y+i,x;" ": NEXT i: GO TO 100
 353 LET hit=ATTR (y+i,x)
 355 FOR n=1 TO i-1: PRINT AT y+n,x;" ": NEXT n
 358 LET y1=y+i: LET x1=x
 360 GO TO 500
 399 REM   Pacman
 420 PRINT INK 1;AT h,l;"h";AT hi,li; INK 0;" "
 422 BEEP .01,0
 425 LET hi=h: LET li=l
 430 LET h=h+(h<y)+INT (RND*2)-(h>y)-INT (RND*2): IF h>21 THEN LET h=hi
 433 IF h<1 THEN LET h=hi
 435 IF ATTR (h,l)<>56 THEN LET h=hi
 440 LET l=l+(l<x)+INT (RND*2)-(l>x)-INT (RND*2): IF l>31 THEN LET l=li
 445 IF ATTR (h,l)<>56 THEN LET l=li
 450 IF h=y AND l=x THEN GO TO 1000
 460 RETURN
 499 REM   explosion
 500 PRINT AT y1,x1; OVER 1;"(isp)": BEEP .2,-20: PRINT OVER 1;AT y1,x1;"(91)": BEEP .1,30: PRINT OVER 1;AT y1,x1;"(92)": BEEP .1,4: PRINT OVER 1;AT y1,x1;"(197)": BEEP .1,45: PRINT AT y1,x1;"(94)": BEEP .1,15: PRINT AT y1,x1;" ": BEEP .4,-15
 505 IF hit=57 THEN LET scr=scr+20: GO TO 80
 510 LET scr=scr+5
 515 LET tots=tots+1: IF tots>19 THEN GO TO 50
 520 FOR n=1 TO 4
 522 LET ty=RND*20+1: LET tx=RND*31: IF ATTR (ty,tx)=60 THEN GO TO 522
 524 PRINT AT ty,tx; INK 2;"j": NEXT n
 530 IF hit=58 THEN PRINT AT y,x; OVER 1;"(isp)": BEEP .2,-20: PRINT OVER 1;AT y,x;"(91)": BEEP .1,30: PRINT OVER 1;AT y,x;"(92)": BEEP .1,4: PRINT OVER 1;AT y,x;"(197)": BEEP .1,45: PRINT AT y,x;"(94)": BEEP .1,15: PRINT AT y,x;" ": BEEP .4,-15: GO TO 1010
 540 GO TO 90
 999 REM   gobble gobble
1000 PRINT AT hi,li;" ";AT h,l; FLASH 1;"h": BEEP 2,-25: PRINT AT h,l; FLASH 0; INK 0;" "
1010 LET man=man-1
1020 PRINT AT 0,0;"(5*isp)";AT 0,0;m$(1 TO man+2)
1025 IF hit=58 AND man>0 THEN LET hit=100: GO TO 90
1030 IF man>0 THEN GO TO 80
2000 REM   end of game
2020 FOR i=10 TO 20: FOR n=5 TO i: BEEP .02,n: NEXT n: NEXT i
2025 IF scr>hscr THEN LET hscr=scr
2030 CLS : PRINT AT 10,8; FLASH 1;"** GAME OVER **"
2040 PRINT AT 0,21;"score=";scr;AT 0,0;"high score=";hscr
2060 PRINT AT 21,2; PAPER 6;"PRESS ANY KEY TO PLAY AGAIN"
2070 PAUSE 0
2080 CLS : GO TO 4
9020 RESTORE 9070: FOR m=1 TO 10
9030 READ p$
9040 FOR n=0 TO 7
9050 READ a: POKE USR p$+n,a
9060 NEXT n: NEXT m: GO TO 1
9070 DATA "a",24,24,8,60,10,24,38,32
9080 DATA "b",24,24,8,24,44,8,8,8
9090 DATA "c",8,8,8,28,42,8,20,20
9100 DATA "d",24,24,16,60,80,24,100,4
9110 DATA "e",24,24,16,24,52,16,16,16
9120 DATA "f",8,8,8,30,40,24,20,4
9130 DATA "g",8,8,8,60,10,12,20,16
9140 DATA "h",24,126,153,255,60,126,90,219
9150 DATA "i",16,84,41,178,93,170,90,24
9160 DATA "j",16,40,84,56,84,170,16,16
```

TASTY TOADS

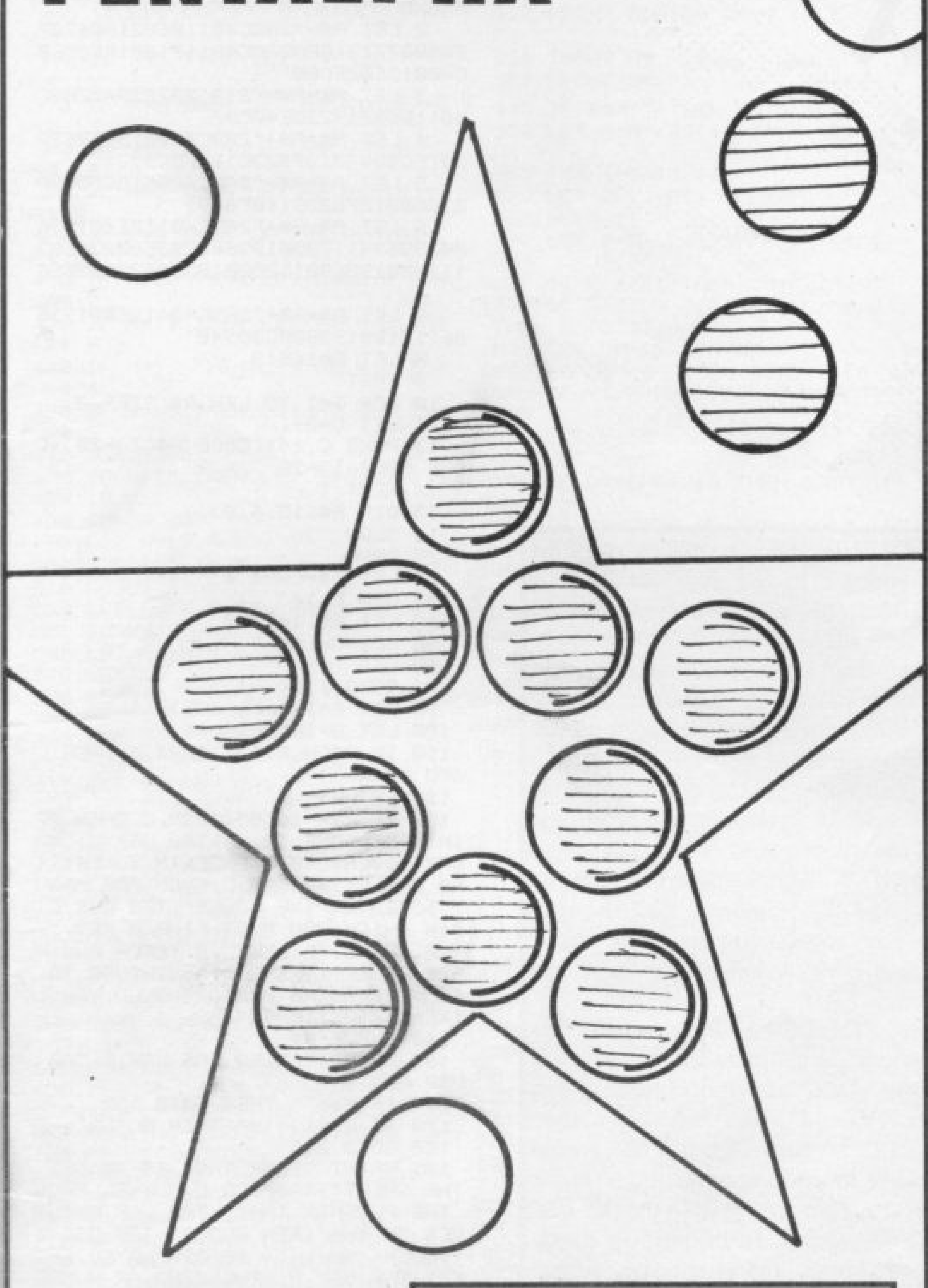
YOU ARE the ravenous toad-eating crocodile at the bottom of the screen. Press 'P' to fire a venom blast which will gulp down one of those tasty toads passing overhead. The Society for the Protection of Toads, which objects to the wholesale slaughter, has erected a forcefield. Firing while a toad is behind it will result in all the frogs escaping, thus ending the game.

Tasty Toads was written for the 1K ZX-81 by Andrew Brewster of Rochester, Kent.



```
   1 REM "T"
   2 LET C=0
   3 LET A=0
   4 RAND
   5 FOR A=1 TO 5
   6 CLS
  10 LET M=18
  20 LET F=2*(INT (RND*8))
  30 FOR B=0 TO 20
  35 PRINT AT 6,0;"*********FORCE*FIELD************"
  40 PRINT AT 19,15;"(i.:96:i.)";AT 18,14;"(9y:3*sp:9t)";AT F,B;"(i.:96:i.)";AT F+1,B;"(9t:i-:9y)"
  50 IF M<F THEN LET M=18
  60 IF INKEY$="P" OR M<18 THEN PRINT AT M,16;"."
  65 IF B=15 AND M=F THEN GOTO 150
  70 IF INKEY$="P" OR M<18 THEN GOSUB 120
  80 PRINT AT F,B;" "
  90 NEXT B
 100 NEXT A
 120 PRINT AT M,16;" "
 125 LET M=M-2
 130 RETURN
 150 PRINT AT M,B;" gulp ";AT M+1,B;"      "
 160 PAUSE 50
 170 NEXT A
```

PENTALPHA



PENTALPHA is a version of an ancient Egyptian solitaire game which L Gabb of Northfield, Birmingham has converted for use on the 16K ZX-81. The aim is to enter the nine counters on the 10 points of a pentagram. A piece may be entered on any unoccupied point and must then be moved two points in a straight line to another unoccupied point.

The intervening point may be occupied or unoccupied. If no more moves are possible before all the counters have been placed, the game is lost. Moves should be entered by pressing the letter corresponding to the point where you wish to enter your counter, followed by the letter corresponding to the point to which you wish to move it.

```
  10 DIM P(10)
  20 LET J$="ICEHBDJFAGIC"
  30 CLS
 100 PRINT AT 0,0;"*Pentalpha*";
TAB 15;"a";TAB 14;". .";TAB 13;"
.   ."
 120 PRINT TAB 6;"b. . .c. . .d.
 . .e"
 130 PRINT TAB 9;". .          . ."
 140 PRINT TAB 10;"f.         .g"
 150 PRINT TAB 9;".    .   .   ."
 160 PRINT TAB 8;".       .h.
."
 170 PRINT TAB 7;".    .       .
 ."
 180 PRINT TAB 6;"i.";TAB 23;".j
"
 190 PRINT
 200 PRINT TAB 6;"move enter mov
e to"
 500 FOR M=1 TO 9
 505 FOR I=2 TO LEN J$-1
 510 IF NOT (P(CODE J$(I)-37) OR
 (P(CODE J$(I-1)-37) AND P(CODE
J$(I+1)-37))) THEN GOTO 590
 520 NEXT I
 530 PRINT AT 12+M,0;"YOU CANNOT
 MOVE AND HAVE FAILED"
 540 GOTO 910
 600 LET A$=""
 610 LET B$=""
 620 PRINT AT 12+M,11;"
   "
 630 PRINT AT 12+M,7;M;TAB 13;"?
"
 640 LET A$=INKEY$
 650 IF A$<"A" OR A$>"J" THEN GO
TO 640
 660 PRINT AT 12+M,13;A$
 670 IF P(CODE A$-37) THEN GOTO
1001
 680 GOSUB 1500
 685 LET A=X
 686 LET B=Y
 690 PRINT AT A,B;"O"
 720 PRINT AT 12+M,20;"?"
 730 LET B$=INKEY$
 740 IF B$<"A" OR B$>"J" THEN GO
TO 730
 750 PRINT AT 12+M,20;B$
 760 IF P(CODE B$-37) THEN GOTO
1001
 770 FOR I=2 TO 11
 780 IF A$=J$(I) AND B$<>J$(I-1)
 AND B$<>J$(I+1) THEN GOSUB 1000
 790 NEXT I
 795 LET P(CODE B$-37)=1
 800 GOSUB 1500
 810 PRINT AT A,B;CHR$ (CODE A$+
128);AT X,Y;"O"
 820 NEXT M
 900 PRINT AT 1,0;">WELL DONE<"
 910 PRINT AT 0,18;"Press r to r
un";TAB 19;"or s to stop"
 920 LET A$=INKEY$
 930 IF A$="R" THEN RUN
 940 IF A$="S" THEN STOP
 950 GOTO 920
1000 LET P(CODE A$-37)=0
1001 PRINT AT 12+M,11;"INVALID M
OVE";AT A,B;CHR$ (CODE A$+128) A
ND B$<>""
1002 FOR I=1 TO 50
1003 NEXT I
1010 GOTO 505
1500 IF B$="" THEN LET T$=A$
1510 IF B$<>"" THEN LET T$=B$
1520 IF T$="A" THEN LET X=0
1530 IF T$>"A" AND T$<"F" THEN L
ET X=3
1540 IF T$="F" OR T$="G" THEN LE
T X=5
1550 IF T$="H" THEN LET X=7
1560 IF T$="I" OR T$="J" THEN LE
T X=9
1570 IF T$="A" OR T$="H" THEN LE
T Y=15
1580 IF T$="B" OR T$="I" THEN LE
T Y=6
1590 IF T$="C" THEN LET Y=12
1600 IF T$="D" THEN LET Y=18
1610 IF T$="E" OR T$="J" THEN LE
T Y=24
1620 IF T$="F" THEN LET Y=10
1630 IF T$="G" THEN LET Y=20
1640 RETURN
```

Phoenix Tower

YOUR QUEST begins as you find yourself in the dark on the first floor of a fifteen-level tower. Your ultimate goal is to climb the tower and find the Great Jewel of Light. On the way you must find keys, armour and many other objects which are essential to your progress.

Opposition is in the form of weird and wonderful creatures which appear from the gloom and which must be destroyed. With luck you will climb the tower, find the jewel and find the ring which tells you how to leave the tower. You may even be helped by the Great Wizard.

Enter "INFO" to obtain a complete vocabulary list, including the code numbers of certain vital objects. Commands should be entered when "NEXT?" is displayed. To use an object, enter its code number.

The program was written for the 16K ZX-81 by Colin Hunniford of Berkhamsted, Herts, and is the first adventure game to be published in *Sinclair User*. We hope that, as well as providing enjoyment, it will also provide hints and inspiration for those interested in writing adventure games.

```
   1 REM MMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMM146 charactersMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMM
   2 LET A$="2A0C4011B6021906207
EC680772310F92A0C4011210019ED5B0
C4001D602EDB0"
   3 LET A$=A$+"2121032239402A0C
4011B60219220E40C9"
   4 LET A$=A$+"2A0C400615C50620
237EC6807710F923C110F2C9"
   5 LET A$=A$+"2A0C400615C50620
23368010FB23C110F4C9"
   6 LET A$=A$+"2A0C40110E001936
00233634112000193603233680233603
112000193680112000193606232336 86
"
   7 LET A$=A$+"2A0C40110E001936
86112100193600C3D940"
   8 LET C=16513
   9 FAST
  10 FOR I=1 TO LEN A$ STEP 2
  11 LET C=C+1
  12 POKE C,16*(CODE A$(I)-28)+C
ODE A$(I+1)-28
  14 NEXT I
  15 DIM A$(15,6,6)
  16 SLOW
  20 RAND
  25 LET D$=CHR$ 0
  30 DIM E(14)
  40 LET E(1)=9999
  50 LET C=16514
  60 LET T=0
  80 GOTO 130
  90 LET A=INT (RND*6)+1
 100 LET B=INT (RND*6)+1
 110 IF A$(N,A,B)<>CHR$ 0 THEN G
OTO 90
 120 RETURN
 130 IF USR 16583<>USR C THEN PR
INT "WELCOME TO :";TAB USR C;"MU
RKY ADVENTURE 2:PHOENIX TOWER";T
AB USR C;TAB USR C;"YOU ARE ABOU
T TO ENTER THE TOWER";TAB USR C;
"IN WHICH YOU MUST FIND A KEY,";
TAB USR C;"A JEWEL,A TORCH AND M
ANY OTHER";TAB USR C;"THINGS TO
SURVIVE.";TAB USR C;"WOULD YOU L
IKE INSTUCTIONS?";
 140 INPUT B$
 150 PRINT " ";B$;TAB USR C;TAB
USR C;
 160 IF B$="" THEN GOTO 130
 170 IF B$(1)="Y" THEN GOSUB 185
 180 GOTO 220
 185 PRINT "YOUR TASK IS TO GET
THE GREAT";TAB USR C;"JEWEL FROM
 THE PHOENIX THAT";TAB USR C;"LI
VES ON THE 15TH FLOOR";TAB USR C
;"WHEN  next !  IS DISPLAYED,ENT
ER";TAB USR C;"THE WORD OR NUMBE
R YOU WANT TO ";TAB USR C;"USE.Y
OU WILL LEARN THE RULES OF";TAB
USR C;"THE GAME AS YOU PLAY.";TA
B USR C;"good luck"
 190 PRINT TAB USR C;TAB USR C;"
WORDS:MOVE,HELP,BASH,UP,DOWN,";T
AB USR C;"TAKE,DROP,INFO,SAVE,SN
EEZE.";TAB USR C;TAB USR C;"1=ST
RENGH","2=TREASURE"
 195 PRINT TAB USR C;"3=JEWEL","
4=SPELL BOOK";TAB USR C;"5=KEY",
"6=SWORD";TAB USR C;"7=RING","8=
LIGHT";TAB USR C;"9=SHIELD","10=
WATER";TAB USR C;"11=WAND";TAB U
SR C;"13=ARMOUR","14=CLUB";TAB U
SR C;TAB USR C;
 196 PRINT "PRESS A KEY AND WAIT
"
 200 IF INKEY$="" THEN GOTO 200
 210 RETURN
 215 REM here the levels are set
 220 FAST
 240 FOR N=1 TO 15
 250 FOR I=2 TO 8
 260 GOSUB 90
 270 LET A$(N,A,B)="■"
 280 GOSUB 90
 290 LET A$(N,A,B)="G"
```

```
 300 NEXT I
 310 GOSUB 90
 340 IF N=1 THEN LET A$(N,A,B)=C
HR$ 5
 350 IF N=2 THEN LET A$(N,A,B)=C
HR$ 14
 360 IF N=5 THEN LET A$(N,A,B)=C
HR$ 11
 370 IF N=6 THEN LET A$(N,A,B)=C
HR$ 13
 380 IF N=7 THEN LET A$(N,A,B)=C
HR$ 10
 390 IF N=8 THEN LET A$(N,A,B)=C
HR$ 9
 400 IF N=9 THEN LET A$(N,A,B)=C
HR$ 8
 410 IF N=10 THEN LET A$(N,A,B)=
CHR$ 7
 420 IF N=11 THEN LET A$(N,A,B)=
CHR$ 6
 430 IF N=13 THEN LET A$(N,A,B)=
CHR$ 4
 450 IF N=15 THEN LET A$(N,A,B)=
CHR$ 3
 460 GOSUB 90
 470 IF N=15 THEN LET A$(N,A,B)=
CHR$ 18
 480 GOSUB 90
 490 LET A$(N,A,B)="S"
 500 FOR I=1 TO 5
 510 GOSUB 90
 520 LET A$(N,A,B)=CHR$ 17
 550 GOSUB 90
 560 LET A$(N,A,B)=CHR$ 15
 570 NEXT I
 580 NEXT N
 590 LET N=1
 600 LET L=1
 610 GOSUB 90
 615 LET Y1=A
 620 LET Y2=B
 630 SLOW
 640 FOR I=0 TO 5
 650 NEXT I
 660 PRINT TAB USR C;TAB USR C;T
AB USR C;"YOU ARE ON THE GROUND
FLOOR OF";TAB USR C;"THE DARK TO
WER"
 665 PRINT TAB USR C;"YOU ENTER
THROUGH AN OPEN DOOR";TAB USR C;
"THE DOOR SLAMS SHUT BEHIND YOU"
;TAB USR C
 666 REM input routine
 670 IF USR C=0 THEN GOTO 4900
 675 IF E(1)<200 THEN PRINT "YOU
ARE FEELING TIRED";TAB USR C;
 680 PRINT "NEXT ? ";
 690 INPUT B$
 700 PRINT B$;TAB USR C;
 705 IF B$="" THEN GOTO 680
 710 IF B$(1)="M" THEN GOTO 915
 720 IF B$(1)="H" THEN GOTO 1840
 725 IF B$(1)="U" THEN GOTO 1220
 730 IF B$(1)="B" THEN GOTO 1490
 734 LET B$=B$+" "
 735 IF B$( TO 2)="DO" THEN GOTO
1270
 740 IF B$(1)="T" THEN GOTO 1660
 750 IF B$( TO 2)="DR" THEN GOTO
2005
 755 IF B$="SAVE " THEN GOTO 445
0
 760 IF B$(1)="I" THEN GOTO 2100
 770 IF B$( TO 2)="SN" THEN GOTO
880
 780 FOR I=1 TO 14
 790 IF STR$ I+CHR$ 0=B$ THEN GO
TO 830
 800 NEXT I
 810 GOSUB 190
 820 GOTO 670
 830 IF E(VAL B$)>0 THEN GOTO VA
L B$*100+3000
 840 PRINT "SORRY,CANNOT DO THAT
"
 850 GOTO 670
 880 IF T>0 THEN GOTO 1155
 885 PRINT "YOU HAVE JUST SNEEZE
D"
 890 LET E(1)=E(1)-1
 900 IF RND>.6 THEN GOTO 1620
 910 GOTO 670
 911 REM moves
 915 IF T>0 THEN GOTO 1155
 920 PRINT "WHICH WAY? (N,E,W,S)
:";
 930 INPUT B$
 935 PRINT B$
 940 IF B$="" THEN GOTO 4600
 950 LET A=A+(B$(1)="S")-(B$(1)=
"N")
 960 FOR I=3 TO 14
 970 LET E(1)=E(1)-E(I)
 980 NEXT I
 990 LET B=B+(B$(1)="E")-(B$(1)=
"W")
1000 IF E(1)<1 THEN GOTO 1440
1006 IF A>6 OR A<1 OR B>6 OR B<1
 THEN GOTO 1110
1009 LET D$=A$(L,A,B)
1010 IF D$="■" THEN GOTO 1110
1020 IF Y1=A AND Y2=B THEN GOTO
1200
1030 LET Y1=A
1040 LET Y2=B
1050 IF D$=CHR$ 18 THEN GOTO 181
0
1060 IF D$=CHR$ 15 THEN GOTO 113
0
1065 IF D$=CHR$ 17 THEN GOTO 176
0
1070 GOSUB 3220
1090 IF D$=CHR$ 3 OR D$=CHR$ 15
THEN LET T=1
1100 GOTO 670
1110 IF USR C+USR 16583<>USR C T
HEN PRINT "BONK.YOU HAVE HIT A W
ALL";TAB USR C;
1111 LET A=Y1
1112 LET B=Y2
1120 GOTO 915
1125 REM monsters
1130 LET X=INT (RND*7)*8+1
1140 PRINT TAB USR C;"HERE IS A"
;" WOLF   N EAGLE  TRICORD MUSHM
AN TROG    NUMPIC  GROG   "(X TO
 X+7);TAB USR C;
1150 GOTO 1090
1155 PRINT "WHAT ABOUT THE MONST
ER?"
1190 GOTO 670
1200 PRINT TAB USR C;"PARDON?";T
AB USR C;
1210 GOTO 915
1215 REM stairs
1220 IF T>0 THEN GOTO 1155
1225 IF A$(L,A,B)<>"S" THEN GOTO
 1330
1230 IF E(5)<1 THEN GOTO 1350
1235 IF L=15 THEN GOTO 4430
1240 LET L=L+1
1250 PRINT "YOU HAVE GONE UP THE
 STEPS"
1260 GOTO 665
1270 IF T>0 THEN GOTO 1155
```

```
1275 IF A$(L,A,B)<>"S" THEN GOTO
 1330
1280 IF E(5)<1 THEN GOTO 1350
1290 LET L=L-1
1300 PRINT "YOU HAVE GONE DOWN T
HE STEPS"
1310 IF L=0 THEN GOTO 1370
1320 GOTO 665
1330 PRINT "NO STAIRS HERE"
1340 GOTO 670
1350 PRINT "YOU NEED A KEY"
1360 GOTO 670
1365 REM escaped
1370 PRINT TAB USR C;TAB USR C;"
WELDONE";TAB USR C;("YOU GOT THE
 JEWEL AND " AND E(3)>0);TAB USR
 C;"£";E(2);" WORTH OF TREASURE
AND YOU";TAB USR C;"GOT AWAY WIT
H ";E(1);" SP$"
1380 PRINT TAB USR C;TAB USR C;"
WOULD YOU LIKE ANOTHER GAME? ";
1390 INPUT B$
1400 PRINT B$;TAB USR C;
1410 IF B$="" THEN GOTO 1380
1420 IF B$(1)="Y" THEN RUN
1425 CLS
1430 RAND USR 16601
1435 REM died
1440 PRINT TAB USR C;"WHAT A PIT
Y,YOU HAVE DIED";TAB USR C;("BUT
 YOU STILL GOT THE JEWEL AND " A
ND E(3)>0);TAB USR C;"YOU GOT £"
;E(2);" WORTH OF TREASURE"
1450 FOR I=0 TO 9
1460 IF USR 16563 THEN NEXT I
1485 GOTO 1380
1486 REM fight
1490 IF T<1 THEN GOTO 4700
1495 LET MS=INT (RND*2000)
1500 LET M=INT (RND*MS)
1505 IF E(13)>0 THEN LET M=M-INT
 (RND*100)
1510 LET Y=INT (RND*E(1))
1515 IF E(9)>0 THEN LET Y=Y+INT
(RND*E(1))
1520 LET X=INT (RND*6)*6+1
1530 PRINT TAB USR C;"OUCH  KICK
 KNOCK BATTERBASH  PUNCH "(X TO
X+5)
1540 LET E(1)=E(1)-M
1550 IF E(1)<1 THEN GOTO 1440
1560 LET MS=MS-Y
1570 IF MS<1 THEN GOTO 1590
1580 GOTO 1500
1590 PRINT TAB USR C;"YOU KNOCKE
D THAT MONSTER A MILE"
1600 LET T=0
1605 IF A$(L,A,B)=CHR$ 15 THEN L
ET A$(L,A,B)=CHR$ 0
1610 GOTO 670
1620 PRINT TAB USR C;" AND DRAWN
 ATTENTION TO YOURSELF"
1625 LET T=1
1630 GOTO 1130
1640 REM take
1660 IF T>0 THEN GOTO 1155
1670 LET D$=A$(L,A,B)
1675 IF D$=CHR$ 15 OR D$="■" OR
D$=CHR$ 0 OR D$="S" OR D$=CHR$ 1
7 OR D$=CHR$ 18 THEN GOTO 1740
1676 PRINT "OK"
1680 IF D$="G" THEN GOTO 1720
1690 LET E(CODE D$)=E(CODE D$)+1
1700 LET A$(L,A,B)=CHR$ 0
1710 GOTO 670
1720 LET E(2)=E(2)+INT (RND*1000
)
1730 GOTO 1700
1740 PRINT "NOTHING HERE TO TAKE
"
1750 GOTO 670
1760 PRINT TAB USR C;"YOU HAVE F
ALLEN INTO MUSH"
1770 LET E(1)=E(1)-INT (RND*100)
1780 IF E(1)<1 THEN GOTO 1440
1790 PRINT TAB USR C;"BUT YOU AR
E ALL RIGHT"
1800 GOTO 670
1810 PRINT "HERE IS A WIZARD AND
 HE GIVES";TAB USR C;"YOU AN OBJ
ET AND HE SAYS:"
1820 LET A$(L,A,B)=CHR$ (INT (RN
D*11)+4)
1825 LET D$=A$(L,A,B)
1830 IF D$=CHR$ 12 THEN GOTO 182
0
1835 GOTO 1070
1836 REM help
1840 IF RND>.7 THEN GOTO 1990
1950 PRINT "SORRY,NO HELP HERE"
1960 LET E(1)=E(1)-10
1970 IF E(1)<1 THEN GOTO 1440
1980 GOTO 670
1990 GOTO 1820
2000 GOSUB 195
2004 REM drop
2005 IF T>0 THEN GOTO 1440
2010 PRINT "WHICH NUMBER WILL YO
U DROP? ";
2020 INPUT B$
2025 PRINT B$;TAB USR C;
2030 FOR I=3 TO 14
2040 IF STR$ I=B$ THEN GOTO 2080
2050 NEXT I
2060 GOSUB 195
2070 GOTO 2010
2080 IF E(VAL B$)<1 THEN GOTO 45
00
2085 LET E(VAL B$)=E(VAL B$)-1
2090 GOTO 670
2095 REM info
2100 PRINT TAB USR C;"YOU HAVE £
```

```
";E(2);" OF TREASURE";TAB USR C;
" AND ";E(1);" SP$ AND YOU ARE O
N";TAB USR C;"LEVEL ";L
2110 GOTO 810
2120 SAVE "PHOENIx"
2130 RUN
3100 PRINT "YOU HAVE ";E(1);" SP
$"
3110 GOTO 670
3200 PRINT "YOU HAVE £";E(2);" O
F GOLD"
3210 GOTO 670
3215 REM objects
3220 PRINT TAB USR C;"HERE IS A"
;TAB USR C;("PIT OF MUSH" AND D$
=CHR$ 17);("WISE WIZARD" AND D$=
3220 PRINT TAB USR C;"HERE IS A"
;TAB USR C;("PIT OF MUSH" AND D$
=CHR$ 17);("WISE WIZARD" AND D$=
CHR$ 18);("SOLID WALL" AND D$="■
");("SILVER SWORD" AND D$=CHR$ 6
);("GOLD RING" AND D$=CHR$ 7);("
TORCH" AND D$=CHR$ 8);("POSH SHI
ELD" AND D$=CHR$ 9);("BUCKET OF
WATER" AND D$=CHR$ 10);("SILVER
WAND" AND D$=CHR$ 11);("SUIT OF
HEAVY ARMOUR" AND D$=CHR$ 13);("
WOODEN CLUB" AND D$=CHR$ 14);("S
TAIR CASE" AND D$="S");("GRAND P
HOENIX GUARDING A JEWEL" AND D$=
CHR$ 3);("NASTY LOOKING MONSTER"
 AND D$=CHR$ 15);("HOARD OF TREA
SURE" AND D$="G");("BOOK OF SPEL
LS" AND D$=CHR$ 4);(" LOAD OF NO
THING" AND D$=CHR$ 0);("GOLDEN K
EY" AND D$=CHR$ 5);TAB USR C;
3230 RETURN
3240 REM light
3300 PRINT TAB USR C;"THE JEWEL
LIGHTS UP";TAB USR C
3305 PRINT TAB USR C;"NORTH";TAB
 USR C
3310 LET D$=A$(L,A-(A>1),B)
3315 GOSUB 3220
3320 PRINT TAB USR C;"SOUTH";TAB
 USR C
3325 LET D$=A$(L,A+(A<6),B)
3330 GOSUB 3220
3335 PRINT TAB USR C;"EAST";TAB
USR C
3340 LET D$=A$(L,A,B+(B<6))
3345 GOSUB 3220
3350 PRINT TAB USR C;"WEST";TAB
USR C
3355 LET D$=A$(L,A,B-(B>1))
3360 GOSUB 3220
3365 GOSUB 196
3370 GOTO 670
3400 IF T>0 THEN GOTO 1155
3405 PRINT TAB USR C;"THE SPELL
BOOK MAKES SOMETHING";TAB USR C;
" FOR YOU"
3410 LET E(1)=E(1)-INT (RND*E(1)
)
3420 GOTO 1820
3500 GOTO 5100
3600 IF T<1 THEN GOTO 4700
3610 PRINT TAB USR C;"YOUR SWORD
 KILLED THE MONSTER"
3620 GOTO 1600
3700 PRINT TAB USR C;"TO GET OUT
 OF THE TOWER,YOU MUST";TAB USR
C;"GO DOWN THE STEPS ON THE BOTT
OM";TAB USR C;"FLOOR"
3710 GOTO 670
3800 PRINT TAB USR C;"YOU SWITCH
 ON THE LIGHT AND SEE:"
3810 GOTO 3305
3900 GOTO 5100
4000 IF A$(L,A,B)=CHR$ 3 AND T>0
 THEN GOTO 4020
4010 GOTO 5100
4020 PRINT TAB USR C;"WATER HAS
PUT OUT THE PHOENIX"
4030 GOTO 1600
4100 PRINT TAB USR C;"WITH A WAV
E OF YOUR WAND,YOU ARE";TAB USR
C;"ON THE NEXT LEVEL"
4110 IF L<15 THEN LET L=L+1
4120 GOTO 1600
4300 GOTO 5100
4400 IF T<1 THEN GOTO 4700
4405 PRINT TAB USR C;"YOU SURE B
ASHED THAT MONSTER"
4410 GOTO 1600
4430 PRINT "THESE STEPS DO NOT G
O UP"
4440 GOTO 670
4445 REM place save
4450 IF USR 16583<>USR C THEN PR
INT "START TAPE RECORDER AND PRE
SS";TAB USR C;"NEWLINE WHEN READ
Y.";TAB USR C;TAB USR C;"rem loa
d |xx|";TAB USR C;
4460 INPUT B$
4470 SAVE "Xx"
4480 GOTO 670
4500 PRINT "NOTHING TO DROP"
4510 GOTO 670
4600 PRINT TAB USR C;
4610 GOTO 920
4700 PRINT "NOTHING HERE TO BASH
"
4710 GOTO 670
4800 REM scroll off last words
4900 PRINT TAB USR C;
5000 IF PEEK (PEEK 16396+PEEK 16
397*256+1)<>128 THEN GOTO 4900
5010 GOTO 675
5100 PRINT "YES,YOU HAVE IT"
5110 GOTO 670
```

Helpline

Our resident expert, Andrew Hewson, has again been busy answering queries. In this section he has collected all his advice on particular subjects to give you the best help in making the ownership of a Sinclair machine more fulfilling.

PEEK, POKE are explained

EVERY MONTH a selection of queries from readers on the difficulties they experience are answered in *Sinclair User*. Here is a selection of those which offer advice which will be of universal interest.

• *Please address problems and queries to Andrew Hewson, Helpline, Graham Close, Blewbury, Oxfordshire.*

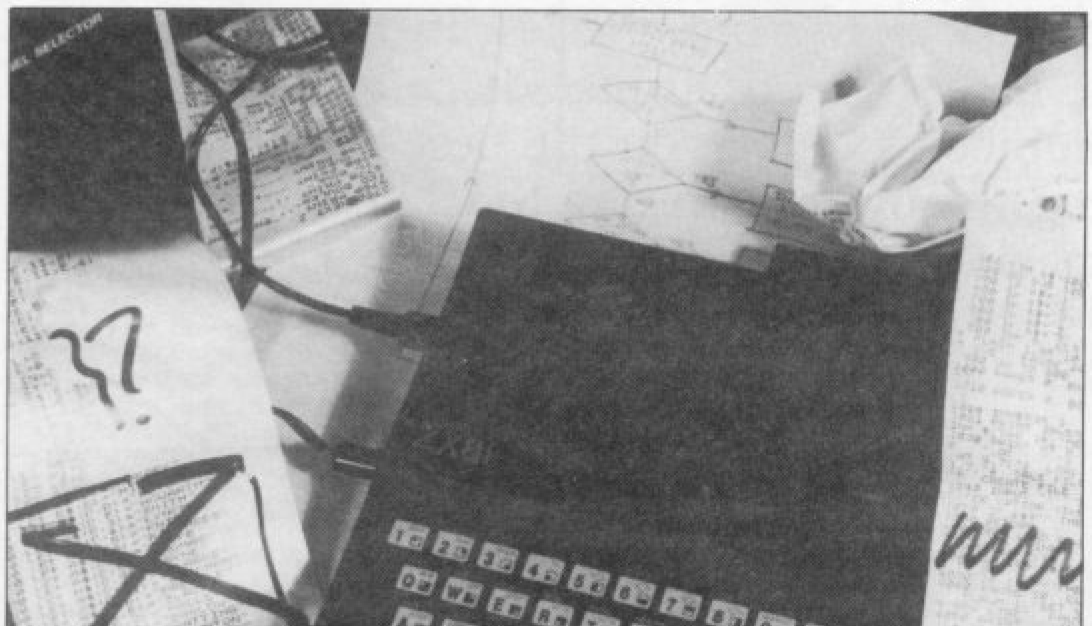
David Anthes of Bridgport writes: **My ZX-81 has a bug. When I POKE 57 into various addresses as per page 163 of the manual, I get weird effects. Sometimes the machine crashes but Sinclair claims there is nothing wrong with it.**

When you enter PRINT PEEK address, your computer PRINTs a positive integer number between 0 and 255 inclusive. That is the number which your machine holds in the byte at location "address". The value of "address" must be a positive integer between 0 and 32767 on the 16K ZX-81 or Spectrum.

POKE is the complementary command to PEEK. It puts a number into a location. The form of the command is:

POKE address, number

Where address is a positive integer in the range 0 to 32767 — 16K machines — and number is a positive integer in the range 0 to 255 it is a dangerous command because POKEing the wrong location can cause the machine to crash for any one of a hundred reasons. PEEKing is a means of looking at what your computer is doing, whereas POKEing is a way of putting a spanner in the works; if you put the spanner in the wrong place or the wrong spanner in the right place you can cause havoc.



John Hawes of Glamorgan wants to PEEK the ZX-81 display file. He asks: **Is it possible to discover whether a given character is at a given position in the display of the 16K ZX-81?**

There are two golden rules to remember when manipulating the ZX-81 display. They are:

The ZX-81 display file must always contain at least 25 bytes filled with the NEWLINE character, decimal 118.

The 1K ZX-81 display file contains only the characters which have been PRINTed or PLOTted, plus sufficient spaces to fill any gap between the left-hand side of the screen and the character in question. The 16K ZX-81 display file contains 25 lines of 32 characters unless SCROLL has been used since the last CLS. The following routine PRINTs the character at row R, column C of the 16K ZX-81 display file:

```
10 FOR I=0 to 5
20 PRINT "ABCDEFGHIJKLM
   NOPQRSTUVWXYZ";
30 NEXT I
100 LET D=PEEK 16396+256*
   PEEK 16397
110 INPUT R
120 INPUT C
130 PRINT AT 21,0; CHR$ PEEK
   (D+33*R+C)
```

Peter Bankes of Debenham asks: **Is it possible to poke the Spectrum to get caps lock?**

The caps lock condition is stored in bit 4 of FLAGS2 at address 23658 in the system variables area. When bit 4 is set, all entries will appear in capitals. Hence a program can determine whether caps lock is set by checking the status of that bit and altering it if required.

The condition of bit 4 may be checked from Basic in a somewhat cumbersome fashion as illustrated by:

```
10 IF INT(PEEK 23658/8)=2*INT
   (INT(PEEK 23658/8)/2) THEN
   PRINT "CAPS LOCK NOT SET":
   GOTO 30
20 PRINT "CAPS LOCK SET"
30 STOP
```

To set bit 4 from Basic and hence turn on the caps lock enter

POKE 23658, PEEK 23658+8

To turn if off again, enter

POKE 23658, PEEK 23658-8

Alternatively you may wish to use the ROM routine which "toggles" the caps lock. The routine is located at address 4317 (10DD hexadecimal) and successive calls of the form

RAND USR 4317

turn the caps lock on and off.

Machine speak

MANY PEOPLE have had problems with machine code. John Stevens of Hammersmith, London writes: **I am trying to learn how to write machine code programs but I am finding it difficult to understand the meaning of some of the words which are used. Can you explain as fully as possible what is the difference between a bit and a byte, and between a register and a variable?**

A bit is the fundamental building block of computer memory and can exist in only one of two states. The two states can be thought of as representing ON or OFF; TRUE or FALSE; YES or NO; UP or DOWN; MALE or FEMALE or any other pair of logically opposite conditions. The mechanism by which a computer memory works is not really important to us but in the Sinclair computers the state of a bit is memorised by setting a microscopic solid state switch either ON or OFF as appropriate.

The usual notation is to think of one state as the ZERO state and the other as the ONE state. A bit is considered to be set when it is in the state representing ONE and to be re-set otherwise. That notation allows us to speak of a given pattern of bits in terms of its binary equivalent and by converting the binary number to a decimal each bit pattern can be given an exceptional positive integer number.

For example, consider eight bits of which the right-most four are set and the left-most four are re-set as illustrated in table one. The binary pattern of the eight bits can be converted to a decimal if it is remembered that, in a binary number, the right-most column is the units column; the next column to the left again is the fours column and so on, doubling at each move to the left. The decimal equivalent of 00001111 is therefore:

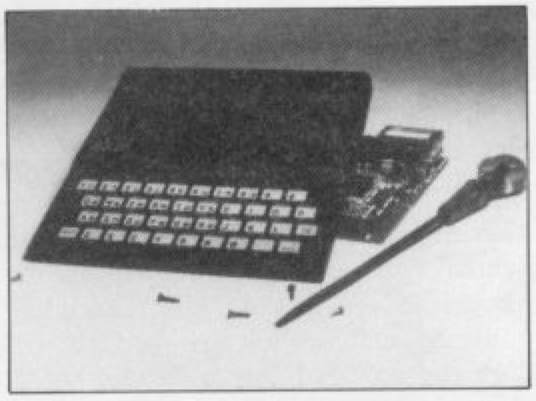
$$0*128+0*64+0*32+0*16+$$
$$1*8+1*4+1*2+1*1=15$$

Obviously it is inconvenient to refer to bits as the right-most or the third from the right and so the convention is adopted of numbering the bits from the right, starting at zero as shown in table one. When that convention is used the number of each bit is also the power to which 2 must be raised to give the value of the column. That is:

$$2^{\text{bit number}} = \text{column value}$$

Bit 3, for example, is in the eights column because $2^3 = 8$.

I chose to consider a group of eight bits together because of the Z-80A microprocessor at the heart of the Sinclair computers is designed to operate on eights bits at a time. The term 'operate' covers all the types of task which the Z-80A can perform directly, such as addition, subtraction, rotation, logical



AND, and the like. Thus although a bit is the fundamental unit of computer memory, bits are usually manipulated together in groups of eight, so a group of eight bits is called a byte — pronounced bite.

There are 256 ways of arranging the contents of a group of eight bits. The first is 00000000, the second is 00000001, the third is 00000010. Thus each of the bytes in RAM can be used to hold a single positive whole number lying between 0 and 255 inclusive by setting or re-setting the eight bits in the byte according to the binary equivalent of the number.

The Z-80A does not alter the contents of memory directly when it is executing a program; rather it copies the contents of a location in memory into one of several special locations in the microprocessor called a register and then operates on the contents of the register. The Z-80A is a powerful microprocessor because it has many registers and so it can hold several numbers at once, thereby reducing the need to make time-consuming transfers between the processor and memory.

Most of the registers have one or more special features. The most important one is the 'a' register or accumulator, so-called because the results of most arithmetic or logical instructions are accumulated in the 'a' register. Some instructions use a second register as a second source of data together with the 'a' register.

For example the instruction:

add a,b

means add the contents of the 'a' register to the contents of the 'b' register and leave the result in 'a'.

Thus a register is a dedicated location in the microprocessor which has specific attributes and functions. A variable is a location or group of locations in RAM which are used by a particular program. If the program is written in Basic or another high-level language, the variable is given a name and all references to the variable are made using the name.

The next question, from Alan Bermingham of London, follows from the previous one. He asks:

What do the following programs do — an assembler, a disassembler, an interpreter, a compiler?

A machine code routine consists of a sequence of instructions which the Z-80A understands directly with no need for prior interpretation. The simpler

Switch setting	**Off**	**Off**	**Off**	**Off**	**On**	**On**	**On**	**On**
State	**Re-set**	**Re-set**	**Re-set**	**Re-set**	**Set**	**Set**	**Set**	**Set**
Binary pattern	0	0	0	0	1	1	1	1
Bit number	7	6	5	4	3	2	1	0

Table 1. Three ways of representing a group of eight bits of which the four left-most are re-set and the four right-most are set. The bit number is shown on the bottom line.

```
10  FOR I=23296 TO 23325
20  INPUT Z$
30  IF Z$="S" THEN STOP
40  PRINT Z$;" ";
50  LET Z$(1)=CHR$(CODE Z$(1)—7*(CODE Z$(1) 57))
60  LET Z$(2)=CHR$(CODE Z$(2)—7*(CODE Z$(2) 57))
70  POKE I,16*CODE Z$(1)+CODE Z$(2)—816
80  NEXT I
```

Table 3. A Spectrum program to load 30 two-character hexadecimal codes into the printer buffer.

instructions are held in one byte of memory but the more complicated instructions can occupy as many as four bytes.

Generally, the instructions are executed in the order in which they are encountered, although there are exceptions. The Z-80A keeps a note of from where the next instruction is to come by means of a special register pair called the program counter. Thus if the location pointed to by the program counter contains the number 128 in decimal — 80 in hexadecimal — the Z-80A will add the contents of the 'a' register to the contents of the 'b' register and leave the result in the accumulator, because 128 is the decimal machine code instruction for

add a,b

The decimal or hexadecimal codes

> 'A disassembler is of use when analysing code written by somebody else'

for all the 600 or so instructions in the Z-80A instruction set are difficult to remember and so for that and other reasons machine code programs are almost always written using an assembler program. An assembler converts instructions like add a,b to the correct code. It also allows the programmer to name variables, add comments and give labels to various points in the program and to call subroutines using the labels. A good assembler will have other facilities as well, all aimed at making the programmer's job as straightforward as possible.

A disassembler performs the opposite function to an assembler; it converts a sequence of numbers into a sequence of mnemonics which are easier to understand than the original code. A list of the more important mnemonics is given in the Sinclair manuals in Appendix A. A disassembler is of use when analysing code written by somebody else to discover how it works.

The output from an assembler is a program which the microprocessor can understand directly because it consists of machine code instructions. In contrast, a program written using an interpreter, such as Sinclair Basic, is held in RAM in more or less the form in which it was entered by the programmer.

Interpreters are high-level languages which bear little or no relationship to the instruction set of the processor on which they are run. Every time the program is executed, however, each line must be analysed by the processor before the required action can be taken. The principal disadvantage of the system is that the programs can be slow to execute, because the processor spends most of its time determining what each program line means.

A compiler circumvents the problem by analysing each program line once only and then storing a sequence of machine code instructions which are equivalent to the original program. Thus the speed of a machine code program is obtained without losing the convenience of a high-level language. The machine code produced by a compiler can be somewhat tortuous and inflexible and so when efficiency is essential an assembler is used instead.

Hex code	Assembler code	Comment
2A 65 5C	LD HL,(23653)	Set HL to address of spare space
44	LD B,H	Copy HL
4D	LD C,L	to BC
3E OO	LD A,O	Set A to O
77	LD (HL),A	Copy A to memory
7E	LD A,(HL)	Copy memory back to A
FE OO	CP O	Compare result
CO	RET NZ	Return if error
3E FF	LD A,255	Set A to 255
77	LD (HL),A	Copy A to memory
7E	LD A,(HL)	Copy memory back to A
FE FF	CP 255	Compare result
CO	RET NZ	Return if error
A7	AND A	Clear carry flag
23	INC HL	Increment HL
ED 72	SBC HL,SP	Subtract stack pointer from HL
C8	RET Z	Return if finished
A7	AND A	Clear carry flag
ED 7A	ADC HL,SP	Add stack pointer to HL
O3	INC BC	Increment BC
18 E7	JR — 25	Repeat for next memory location

Table 2. A Spectrum machine code program to test each memory location between the beginning of the spare space—marked by the STKEND system variable—and the bottom of the stack.

Memory growth

NICHOLAS KENNEDY of Belfast, Geoffrey Hulme of Stoke, and Gareth Rieley of Nottingham, are interested in adding RAM to their ZX-81s or Spectrums. They ask questions like: **What are the advantages and problems of using a 64K RAM on the ZX-81? Can 16K Spectrum programs be loaded into the 48K machine? Can 16K ZX-81 RAMs be used on the Spectrum? Why is there a difference in price between add-on RAMs for the Model One and Model Two Spectrums?**

In the face of such a barrage I contacted Stephen Adams, who specialises in dismantling Sinclair computers for fun, for the latest information. Most people know that 64K of total memory — i.e., ROM and RAM — is the most which can be used with the Z-80 microprocessor without special paging facilities. In the ZX-81 the bottom 8K of memory is reserved for the ROM — the set of machine code instructions which give the machine its character. In the unexpanded machine the next 8K is absent and then there is 1K of RAM at addresses 16K to 17K.

When a 16K RAM pack is added it displaces the 1K of RAM to occupy addresses 16K to 32K. The implication of that design is that Sinclair intended originally to release a new ROM with enhanced facilities to occupy addresses 0 to 16K.

In fact, it has never done so but other manufacturers have stepped in to fill the gap with graphics units and assemblers; 16K and 32K memories are also available which occupy addresses 16K to 32K and 16K to 48K respectively.

In either case, all the extra memory is available directly to the Basic system; 64K RAMs are available of which 48K lies between addresses 16K and 64K and is available for normal use. Another 8K replaces the "missing" 8K of ROM but that part of memory can be PEEKed and POKEd only from Basic. The final 8K is present but unused, which means presumably that the manufacturer also has some future application in mind.

The memory situation for the Spectrum is equally complicated but for rather different reasons. In this machine the ROM occupies 0 to 16K and the

Continued on page 110

BOOK A TEST FIGHT TODAY!

You'll have a REAL fight on your hands when you tackle our new games.

They're all-action games designed to push your senses to new limits. Take "Brain Damage"; trapped in your computer's micro-circuits you'll need all your resources to defeat the CPU's electron panzers ... and if they don't get you, the Rogue Programs will.

When you get your breath back you'll be running for your life again in "Exterminator". Transported into the future you'll find that earth is a very different place. Robots rule the world and the human race is almost extinct, there's only one family left. You have to save them ... but hurry you don't have much time.

Sitting in defense command, your hands poised at the controls, you scan the sky for activity knowing that it will shortly erupt. "Armageddon" is about to begin ... only YOU can save the last six cities.

If you're still in control of your senses you can try your hand at rescuing the miners trapped on Titan ... easy you think! Ride your ship through the mini-meteor storm and pick up the men. But you didn't reckon on the corporation spaceships making this "Mission Impossible".

Book your test flight today. Buy any of Silversoft's new games and find out if you're really in control.

To: Silversoft Ltd. London House, 271/273 King St, London W6

Please send me ....

....

I enclose a cheque/postal order for: ....

Please debit my ACCESS No ....

Name ....

Address ....

....

....

All games run on any ZX-SPECTRUM and cost £5.95, write today for full details of our complete range of software.

Selected titles are available at W.H. SMITHS, BOOTS, JOHN MENZIES, and at all good computer stores.

Continued from page 108

standard 16K RAM occupies the remaining area up to 32K. The easiest way to add more memory is to return the machine to Sinclair with a suitable cheque and it will do the job, but that means doing without your machine for a time.

Add-on memory is available from other manufacturers but Sinclair has re-designed the circuit board completely since its original launch and so it is necessary to determine whether your machine is an early version — model one — or a re-designed version — model two.

Most model one machines have grey keys, whereas model two has light blue keys. To be certain which model you have you should look into the machine through the slot in the rear where the edge connector is situated.

If you have a 16K model one you will see a vacant 16-pin socket on the printed circuit board lying to the left of the slot and about two centimetres from the rear of the machine.

If your machine is a 48K model one the 16-pin socket will not be vacant but will act as the connector between the main printed circuit board and a subsidiary board on which the extra memory and decoding chips are mounted.

In that case you will also see seven circular capacitors, each rather smaller than a halfpenny piece, mounted on the subsidiary board in a line facing the rear of the machine.

The 16-pin socket is not present on the model two because there is space available on the re-designed main printed circuit board for the extra memory and decoding chips. Instead, the very large Z-80 microprocessor chip is clearly visible through the slot, lying approximately below the 9 key on the keyboard.

The reason for the price difference for add-on memory between model one and model two machines is apparent. To extend a model one the necessary chips must be mounted on a suitable board to be located above the main board. To extend a model two the necessary chips need only to be inserted in the sockets already provided on the main board.

It is not possible to plug a ZX-81 RAM pack directly into a Spectrum, although at least one company now markets a small adaptor to allow you to do so, thus giving 32K of RAM with a 16K ZX-81 RAM pack or 48K of RAM with a 64K ZX-81 RAM pack.

Programs written for a 16K or 32K Spectrum should work without modification on the 48K machine and it is safe to assume that all 16K programs available commercially will run on the larger machines. Of course, the converse is not true; it is most unlikely that a program written for 48K would function on a smaller machine.

Henry Evanson of Bromborough writes: **I wish to use the 48K software packs but the first I purchased will not load. How can I check that I have received a 48K Spectrum and not a 16K machine by mistake?**

The answer is to check the high byte of the P-RAMT pointer at address 23733 by entering:

PRINT INT (PEEK 23733/4)-15; "K"

The machine sets the contents of 23733 to 255 — or 127 on the 16K Spectrum — when it is plugged-in and the line above performs a simple calculation and PRINTs 16K or 48K as appropriate.

At any particular time a user may run out of memory. Michael Fawcett has a problem with his ZX-81 in that regard. He writes: **When I get near to the end of entering a particular program the cursor disappears and whenever a key is pressed subsequently, a letter disappears. What is wrong?**

Many ZX-81 users will recognise in Fawcett's remarks that he is running out of memory. The only guaranteed solution is to buy an add-on RAM pack, although a certain amount of space can be saved by deleting REM statements and shortening PRINT statements. The Spectrum, incidentally, emits a low-pitched buzz when memory is short.

A check on the amount of memory left can be made at any time by entering the following REM at line 1:

1 REM RESERVE SOME SPACE

and POKEing the following numbers:

33, 0, 0, 57, 237, 91, 28, 64,
167, 237, 82, 68, 77, 201

into locations 16514, 16515 . . . 16527 (on the ZX-81).

The equivalent locations on the Spectrum are 23760, 23761 . . . . . 23773 and the numbers to be POKEd are the same except that 101, 92, should be substituted for 28, 64.

In each case a short machine code routine is loaded into the space reserved in the REM statement and then the amount of memory left can be PRINTed by entering:

PRINT USR 16514 — 24; "BYTES"

or in the case of the Spectrum:

PRINT USR 23760 — 24; "BYTES"

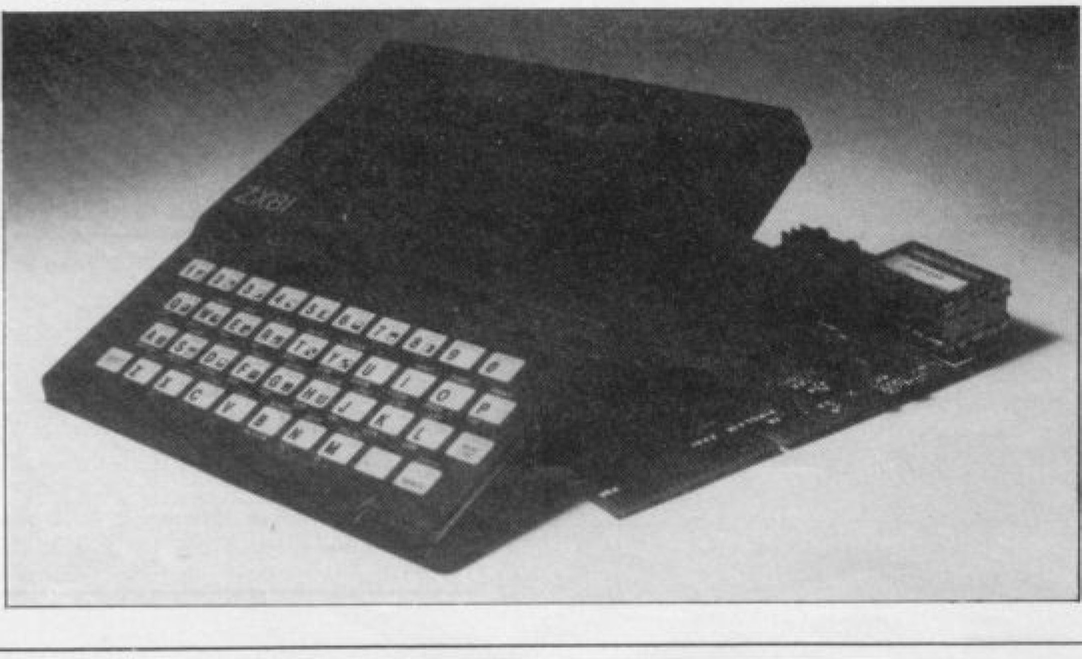
Robert King of London asks: **I have a program which works well on my friend's Spectrum but always crashes on mine. I suspect a fault in the RAM. Have you a program which checks each RAM location in turn?**

Checking RAM involves setting every bit of every byte, checking that it remains set, and then re-setting every bit and checking that it then remains re-set. Setting every bit in a byte is equivalent to POKEing 255 into that byte. Similarly, re-setting every bit is equivalent to POKEing in 0.

Obviously it is not possible to POKE numbers into every memory location while the machine is running, because the computer will crash, but a partial check can be made by testing every location in the spare area of memory.

The program in table two runs such a check. It can be loaded using an assembler or using the simple hexadecimal loader listed in table three.

The program checks every location up to the bottom of the stack and returns the address at which it stops—about 32575 in the 16K machines and 65343 in the 48K machine when they are working correctly.



Storing data in string arrays

UNDERSTANDING data storage and arrays is something which puzzles many readers.

Norman Disley of Cheshire has a collection of about 50 music cassettes which he wishes to catalogue on a Spectrum. He writes: **Each tune would require six or more entries — title name, soloist, and as each cassette contains about 30 titles the collection would use almost all of the 9999 lines of the Spectrum, leaving little room for later insertions. What can I do?**

Several people have written outlining the same idea of using the 9999 program lines available to store up to 9999 items of information. The technique has the merit of simplicity in use but there is no possibility of writing a program to search, sort or PRINT the information.

The conventional technique is to store the information in a numeric or string array. The length of a numeric array is 4+2* number of dimensions +5* total number of elements.

Thus, for example, the two-dimensional array N(2, 5) which contains 10 elements occupies 4+2*2+5*10=58 bytes.

A string array occupies 4+2* number of dimensions + total number of elements and so Zs (2, 5) uses 4+2*2+10=18 bytes.

Disley wishes to store string data and so allowing 10 characters per entry, six entries per tune, 30 tunes per cassette and 50 cassettes, he would need to declare a string array:

Z$(50,30,6,10)

This would occupy

4+2*4+50*30*6*10=90,012 bytes

There are about 10,000 bytes available to the user in the 16K Spectrum and about 42,000 bytes in the 48K machine. Clearly Disley will not be able to hold all the data in the machine at the same time in that manner. The problem would still arise even if he were to use program lines, as each character in the line occupies one byte and there is an overhead of six bytes per line.

Space could be saved by using fewer than 10 characters per entry, perhaps by using abbreviations. Alternatively, the six entries per tune of 10 characters each — 60 characters in all — could be combined into a single entry of, say, 30 characters so that less space was wasted on trailing blanks.

A more sophisticated solution would be to hold all the information in one long string array and to use a second, numeric, array to point to the location in the string array at which each entry began. The slicing technique described in *ZX Spectrum Basic Programming* could then be used to select a given entry. The following program demonstrates the method:

```
10 DIM P(101): REM DECLARE
   POINTER ARRAY
20 LET PZ=1
30 DIM Z$ (1000): REM DECLARE
   STRING ARRAY
40 INPUT "ENTER DATA"; A$
50 IF P(PZ)+LEN A$>1000 GOTO
   100
60 LET P(PZ+1)=P(PZ+LEN A$)
70 LET Z$ (TO P(PZ+1))=Z$
   (TO P(PZ))+A$
80 LET PZ=PZ+1
90 IF PZ<102 THEN GOTO 40
```

'I want to enlarge the array in a program'

John Brookes of Bromsgrove writes: **The books are generally easy to read but none of them explains arrays sufficiently clearly for us wooden-headed types. Can you help?**

Most tasks performed by computers comprise reading information into memory, manipulating the information according to a program of stored instructions, and writing the information out of memory.

The information stored in memory must be organised in some convenient fashion so that the person who writes the program of instructions can do his job. Most high-level languages allow the programmer to declare variables of various kinds so that different types of information can be stored and treated in appropriate ways.

The ZX-81 and the Spectrum each allow six types of variables — numeric with a single character name; numeric with a multiple character name; control for a FOR-NEXT loop; string; string array; numeric array.

The first two types are identical in use, as they can each store only a single positive or negative number. Most programmers try to give a name to a variable which reminds them of the information it holds. For example, a bank account program might hold the current balance in a variable called BALANCE.

In many programs similar information is to be stored concurrently and in that situation the program is also likely to become unnecessarily cumbersome because the same operation must be performed on many variables, each with a different name and therefore requiring a separate piece of code.

The bank account program might be required to store the amount spent using each of 30 cheques in a cheque book. The amount spent using the first cheque could be stored in a variable called CHEQUE. The same variable could not be used to store the value of the second cheque because only one value can be stored in a numeric variable.

The act of entering the second value over-writes the first value, causing it to be lost, hence the programmer must think of a new name for the second cheque. Most programmers would run out of inspiration before they had named all 30 cheques.

One way of naming all 30 which would not require too much effort would be to call them CHEQUE1, CHEQUE2, CHEQUE3. That is the idea of an array. Sinclair arrays are restricted to single-character names

```
10 PRINT AT 0,11;"CHEQUEBOOK"
20 PRINT AT 2,0;"ENTER CURRENT
   BALANCE"
30 INPUT BALANCE
40 DIM C(30)
50 PRINT AT 2,0;"ENTER EACH
   CHEQUE IN TURN"
60 FOR I=1 TO 30
70 PRINT AT 4,0;"CURRENT
   BALANCE= ";BALANCE
80 INPUT C(I)
90 LET BALANCE=BALANCE-C(I)
100 NEXT I
110 CLS
120 PRINT "CHEQUE NO.","VALUE"
130 FOR I=1 TO 30
140 PRINT I,C(I)
150 IF PEEK 16442-3 THEN SCROLL
160 NEXT I
```

Table 1. A simple cheque book program.

only but that is a small price to pay for the flexibility they provide.

Table one shows a simple cheque book program which uses an array, C, of length 30 to hold the value of each cheque as it is entered. The program also shows the use of another variable, I, to count through the array selecting each element of the array in turn. When all the cheques have been entered the program prints the value of each in turn.

Simon Smith of Battersea, London has a more complex question concerning arrays. He writes: **I have a ZX-81 and recently bought a 64K memory to replace the 16K RAM pack I had previously. I now want to enlarge the array in a program which indexes my record collection to make use of the extra memory space. Is there any way of doing so, short of copying the data to another array?**

Defining a new array and copying the data across using a FOR-NEXT loop would certainly be the easiest way of dealing with the problem. The disadvantage is that the new array can occupy only the space which is not used by the old array; hence when the old array is deleted — by DIMensioning it to zero — the memory space it occupies remains unused. If the old array was 12K bytes long and a further 32K bytes of memory was added, the new array could occupy only 32K of the total of 44K bytes available.

There is no Basic command to make an array grow to fill the remaining memory space and so I have written the machine code routine listed in table two to do the job. The routine can be loaded into a REM statement forming the second line in a ZX-81 program using a hexadecimal loader. For example:

```
10 REM XA$
20 REM AT LEAST 108
   CHARACTERS
30 FOR I=16523 to 16630
40 INPUT Z$
50 IF Z$="S" THEN STOP
60 PRINT Z$;"  ";
70 POKE I,16*CODE Z$+CODE
   Z$(2)–476
80 NEXT I
```

To load the machine code routine, run the loader and enter each pair of hexadecimal codes in turn. Be very careful not to make mistakes because the program makes no error checks.

To use the routine to double the size of a string array called A$ make the first program line a REM statement containing an "X" followed by the name of the array, i.e., A$ as in the loader. Then POKE the factor by which you want the array to grow into the first byte of the REM statement. In that case the array is to grow by a factor of two so you should enter:

POKE 16514,2

Then call the machine code routine by entering:

RAND USR 16523

Table 2. A ZX-81 routine to enlarge an array dynamically.

Hex code	Assembler code	Comment	Hex code	Assembler code	Comment
3A 82 40	LD A,(16514)	Return if	09	ADD HL,BC	Set HL to
FE 01	CP 1	parameter	38 2B	JR C,43	new length.
D8	RET C	is zero.	D1	POP DE	
21 83 40	LD HL,16515	Set CH-ADD.	E5	PUSH HL	
22 16 40	LD (16406),HL		A7	AND A	
CD 1C 11	CALL LOOKVARS	Find array.	ED 52	SBC HL,DE	Set BC to
D8	RET C	Return if	44	LD B,H	increase
C0	RET NZ	numeric	4D	LD C,L	in length.
7E	LD A,(HL)	Return	E1	POP HL	
FE 80	CP 128	if	E3	EX (SP),HL	Set HL to
D8	RET C	string.	E5	PUSH HL	address of
23	INC HL	Set A to	19	ADD HL,DE	end of array.
23	INC HL	number of	23	INC HL	
23	INC HL	dimensions.	CD 9E 09	CALL MAKE SPACE	Enlarge array.
7E	LD A,(HL)		E1	POP HL	
CB 27	SLA,A	Multiply by	D1	POP DE	
06 00	LD B,0	two and	73	LD (HL),E	Insert new
CB 10	RL B	add one,	23	INC HL	length.
3C	INC A	transfer	72	LD (HL),D	
4F	LD C,A	to BC.	23	INC HL	
2B	DEC HL	HL points to	23	INC HL	
56	LD D,(HL)	length.	4E	LD C,(HL)	Set BC to
2B	DEC HL	DE is	23	INC HL	first
5E	LD E,(HL)	old length.	46	LD B,(HL)	dimension.
E5	PUSH HL	Save.	11 00 00	LD DE,0	Calculate
D5	PUSH DE		EB	EX DE,HL	new first
EB	EX DE,HL	Set DE	3A 82 40	LD A,(16514)	dimension.
A7	AND A	to old	09	ADD HL,BC	
ED 42	SBC HL,BC	data length.	3D	DEC A	
EB	EX DE,HL		FE 00	CP 0	
21 00 00	LD HL,0	Calculate	20 FA	JR NZ,-6	
3A 82 40	LD A,(16514)	new data	EB	EX DE,HL	
A7	AND A	length.	72	LD (HL),D	Store new
19	ADD HL,DE		2B	DEC HL	first
38 33	JR C,51	Jump on overflow	73	LD (HL),E	dimension.
3D	DEC A		C9	RET	
FE 00	CP 0		E1	POP HL	If overflow
20 F8	JR NZ,-8		E1	POP HL	then return.
			C9	RET	

Both the total length of the array and the size of the first dimension will grow by the factor specified at address 16514.

The routine makes a number of error checks. It will detect if the named variable does not exist or is not an array. It also ensures that there is sufficient room in RAM to enlarge the array by the factor specified.

The routine makes use of two ROM routines. The first is LOOKVARS which is located at 4380 decimal — 111C hexadecimal.

That routine finds the location in the variable area of the variable pointed to by CH-ADD and returns the address in the HL register pair. The second ROM routine, MAKE SPACE, inserts BC bytes at the address pointed to by HL.

The operation of the routine can be checked by adding the following lines to the hexadecimal loader:

```
1000 DIM A$(2,3)
1010 LET A$(1)="ABC"
1020 LET A$(2)="DEF"
1030 GOSUB 2000
1040 POKE 16514,2
1050 RAND USR 16523
1060 GOSUB 2000
1070 STOP
2000 LET W=PEEK
     16400+256*16401
2010 FOR I=W TO W+21
2020 PRINT I,PEEK I
2030 NEXT I
2040 PAUSE 32768
2050 CLS
2060 RETURN
```

Execute the test program by entering:
RUN 1000

The program declares an array A$ and PRINTs the contents of the first 22 bytes of the variable area where the array is held.

The user can note the contents of the display and compare it to the format of an array as shown on pages 173 and 174 of ZX-81 Basic Programming.

The program then doubles the size of the array and displays the first 22 bytes again. The user will see that the total length of the array has increased appropriately and that the size of the first dimension has doubled.

With a little care the routine can be adapted to run on the Spectrum. The ROM routines LOOKVARS and MAKE SPACE are located at .10418 and 5717 respectively. CH-ADD is at 23645 — 5C5D hexadecimal.

It is also necessary to alter the addresses at which the scale factor and the name of the array are stored.

Number crunch on Sinclair

A QUESTION often posed by those writing an educational program who have incorporated a number of problems into their program and want to present, say, half of them in one session, is asked by Steven Maltby of Northampton. **How do you tell the ZX-81 to think of five random whole numbers between 1 and 10 without the risk of repetition?**

The obvious answer to his question is a routine of the form:

```
10 DIM A(5)
20 LET A(1)=1+INT(RND*10)
30 PRINT A(1),"";
40 FOR I=2 TO 5
50 LET R=1+INT(RND10)
60 FOR J=1 TO I-1
70 IF A(J)=R THEN GOTO 50
80 NEXT J
90 LET A(I)=R
100 PRINT A(I);"";
110 NEXT I
```

Array A is used to store each selection and if a repetition arises it is discarded and a fresh random number is drawn. The routine works well for small samples but becomes progressively slower if the number of samples is increased.

To see the effect, try changing lines 10, 20, 40 and 50 as follows, to simulate the effect of drawing all 52 cards from a pack in random order:

```
10 DIM A(52)
20 LET A(1)=1+INT(RND*52)
40 FOR I=2 TO 52
50 LET R=1+INT(RND*52)
```

Each selection tends to take longer than the previous one because the chance of choosing a number which has appeared previously increases. In choosing the final number, the ZX-81 must select the correct number out of 52 possibilities, which can take a very long time.

A much more elegant solution is to use a routine which mimics the act of shuffling a pack of cards, i.e., create an array containing all 52 possibilities in order and then randomise the locations of each element of the array in turn. A suitable routine of this form for Maltby's problem is:

```
10 DIM A(10)
20 FOR I=1 TO 10
30 LET A(I)=I
40 NEXT I
50 FOR I=1 TO 10
60 LET R=1+INT(RND*10)
70 LET S=A(R)
80 LET A(R)=A(I)
90 LET A(I)=S
100 NEXT I
110 FOR I=1 TO 5
120 PRINT A(I)
130 NEXT I
```

Lines 10 to 40 set up the array, lines 50 to 100 swop each element in turn with another element selected at random — the equivalent of shuffling cards — and lines 110 to 130 PRINT the first five elements of the randomised array.

Alan Sheldon of Aylesbury asks: **It would appear that numbers in the program area of memory are followed by additional information which does not appear in listings. Is that so and if so why?**

Sheldon is correct as can be seen by entering the Spectrum program listed in table one. The program will also work on the ZX-81 if line 15 is altered to read:

```
15 LET S=16509
```

Line 5 is a dummy line, the purpose of which is to allow the user to study the appearance of numbers in programs. When the program is RUN it looks at the contents of the first 21 bytes in the program area and displays them on the screen. The results for the Spectrum are shown in table two.

The first two bytes contain the line number (5) and the next two bytes specify the length of the remainder of the line (11 bytes). The next four bytes hold the character code for the first line of the program:

LET A=1

The character codes vary slightly between the two machines. For example, the code of the letter 'A' is 65 on the Spectrum and 38 on the ZX-81 although the code for the 'LET' is 241 on both machines. The full list of character codes is given in appendix A of the manual supplied with each computer.

On the Spectrum the next byte contains 14. That is not the code for the end of a line, as might be expected, but instead it is described in appendix A of the manual as "number". In fact, the byte acts as a signal to the LIST and other commands to ignore the byte and the contents of the five locations which follow it. Hence there is no indication in listings of the program that those additional locations are used. The line is terminated by the next byte which contains 13 — the ENTER character.

On the ZX-81 the character codes are different but the effect is the same. The location containing the code for '1' is followed by six "hidden" bytes, which do not appear in program listings.

Some clue as to the purpose of those hidden bytes can be gained by replacing line 5, the dummy line, by another line. Try, for example, RUNning the program with

```
5 LET A=2.7
```

as the dummy line. The characters for the number "2.7" occupy three bytes, not one as for the number "1", but again the number is followed by six hidden bytes. A few minutes' experimentation will show that whenever a number appears within a program six hidden bytes follow.

The reason for the use of the hidden bytes is that the ZX-81 and the Spectrum do not store and manipulate numbers in the character form in which they are displayed. They are converted into a "calculation" format and all additions, multiplications and so on are undertaken on the numbers in this format. When

```
 5 LET A=1
10 PRINT "BYTE"; TAB 6;
   "CONTENTS"; TAB 16;
   "CHARACTER"
15 LET S=PEEK 23635+256*PEEK
   23636
20 FOR I=S TO S+20
25 PRINT I;TAB 8;PEEK I;TAB
   20;CHR$ PEEK I
30 NEXT I
```

Table 1. A Spectrum program which looks at the first 21 bytes of the program area.

Byte	Contents	Character
23755	0	?
23756	5	?
23757	11	?
23758	0	?
23759	241	LET
23760	65	A
23761	61	=
23762	49	1
23763	14	?
23764	0	?
23765	0	?
23766	1	?
23767	0	?
23768	0	?
23769	13	

Table 2. The first 15 lines of the screen display produced when the Spectrum program in Table 1 is executed.

the result of a calculation is PRINTed it must be converted into characters for display on the screen. Similarly, the character form of a number entered by the user must be converted to the calculation format before a calculation can be executed.

All such conversions take time. To accelerate the execution of programs the conversion to calculation format is undertaken immediately a number in a program line is entered from the keyboard. The resulting five-byte form is stored in the hidden bytes. The use of this technique enables a considerable saving to be made in the time taken to execute a program, particularly if numbers are included within FOR loops, in which case the same conversion would otherwise be undertaken many times. Of course, the time taken to deal with a program line entered from the keyboard is lengthened but not to an unacceptable extent.

Hugo Cassidy of Dorchester asks: **Can you explain the method of encoding numbers on the Spectrum?**

Given that it is necessary to convert numbers from decimal to binary, it is logical to use a binary format which is efficient and therefore fast for the computer to use. Two separate formats are used on the Spectrum, a special format for integers, or whole numbers, lying in the range −65535 to 65535 and a floating point format for all other numbers. The ZX-81 uses the floating point format only.

The integer format is the simplest to understand and so I shall explain it first. A suitable number, N, is converted to the five-byte form by setting the first and fifth bytes to zero and using the second byte to indicate the sign of the number, 0 for positive, 255 for negative. If the number is positive the value is stored in the third and fourth bytes as:

Third byte=N—256*INT (N/256)
Fourth byte=INT (N/256)

If N is negative the two bytes contain:

Third byte=65536—N—256*INT ((65536—N)/256)
Fourth byte=INT ((65536—N)/256

The principal advantage of the use of integer format is that for positive integers the third and fourth bytes are in the form the Z-80A microprocessor uses when addressing locations in memory. Commands such as PEEK and POKE are executed much faster than they would otherwise be if the more complex floating point form were used to store the addresses to which they refer. The format also enables the calculator routines in the ROM to execute much more quickly when calculations involving integers only are performed.

The program in table one can be used to inspect the positive integer form by varying the first line. For example, entering:

5 LET A=47

will show that 47 is held as 0,0,47,0,0. The negative version cannot be inspected using this program because all numbers are stored in their positive form in the hidden bytes. If a number is preceded by a negative sign it is negated when the line is executed.

The program in table four gives the five-byte form of any number, positive or negative, entered from the keyboard. The program PRINTs the contents of the first item in the variable area, that is the number N entered by the user from the keyboard, because it is the first variable declared in the program.

Note that the program should be initiated by entering RUN rather than GOTO 10 because doing so will cause the variables area to be CLEARed, thus ensuring that N is the first variable.

The floating point form is designed to provide the computer with a systematic method of retaining as much accuracy as possible in any given calculation. Some numbers cannot be specified completely in decimal form. The fraction one-third in decimal form consists of 1.3 followed by an infinite number of threes so that expressing it as 1.3333, for example, is almost, but not exactly, correct. The same problem occurs when binary arithmetic is used.

The solution is to retain only the most significant digits at each stage in a calculation. Provided more significant digits are retained than are required in the answer then in all but the most exceptional circumstances the calculated result will be accurate enough for practical purposes.

The program listed in table five calculates and PRINTs the floating point form of a number entered by the user. The line numbers have been set so that it can be placed in memory at the same time as the inspection program in table four. By entering the same number into both programs the user will see that the calculation is correct.

The program has two parts. The first stores the sign, S, of the number, X, entered by the user. It then multiplies the absolute value of X successively by 2 until the result exceeds 2 raised to the power 31 or 21474383648. The number of multiplications executed is stored in N. The new value of X then lies necessarily between 2 to the power of 31 and 2 to the power 32 and so the integer part of the number can be stored exactly in 31 bits.

Thus by discarding the fractional residue the number can be stored in four bytes, each containing eight bits with one bit left over to hold the sign of the number. The four bytes together are called the mantissa.

The second part of the program calculates the values held in each of the four bytes and stores them in the variables A, B, C and D and then PRINTs the variables. An adjustment is made to the value of A depending on the sign of the original number. In effect A is less than 128 for positive numbers and greater or equal to 128 for negative numbers.

The fifth byte of the floating point form is used to store the exponent, that is the number of times that the manitissa must be divided or multiplied by 2 to place the decimal point in the correct position in the number.

The program calculates that number using N, the number of multiplications made originally. The result is adjusted by adding 160 so that numbers greater than or equal to 128 and numbers less than one have exponents less than 128.

```
10 INPUT N
20 PRINT N;" ";
30 LET A=PEEK 23627+256*PEEK
   23628
40 FOR I=A+1 TO A+5
50 PRINT PEEK I;" ";
60 NEXT I
70 GOTO 10
```

Table 3. A Spectrum program to inspect the five-byte form of number entered by the users.

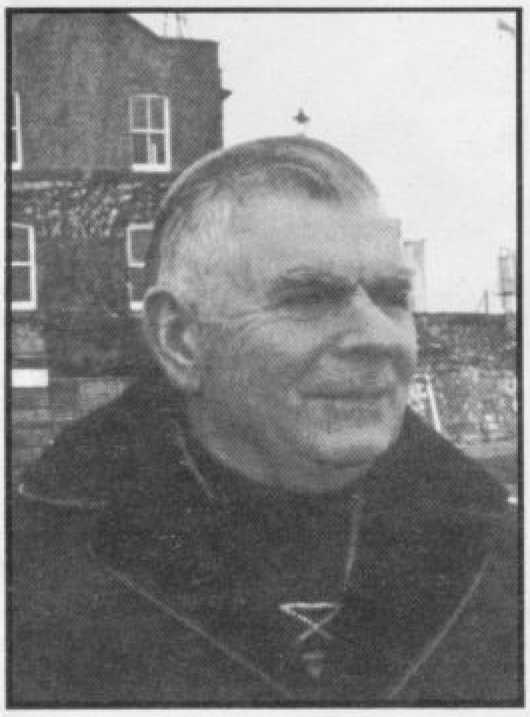
To run the program on the ZX-81 change line 30 to read:

```
30 LET A=PEEK 16400+256*PEEK
   16401
```

```
210 LET N=O
220 INPUT X
230 LET S=SGN X
240 LET X=ABS X
250 LET X=2*X
260 LET N=N+1
270 IF X<2147483648 THEN GOTO
    30
280 LET A=INT (X/16777216)
290 X=X—16777216*A
300 LET B=INT (X/65536)
310 LET X=X—65536*B
320 LET C=INT (X/256)
330 LET D=X—256*C
340 PRINT "EXPONENT="; 160—N
350 PRINT "MANTISSA=";A-
    128*(S=1);" ";B;" ";C;" ";D
```

Table 4. A Spectrum program to calculate and PRINT the floating point form of a number entered from the keyboard.

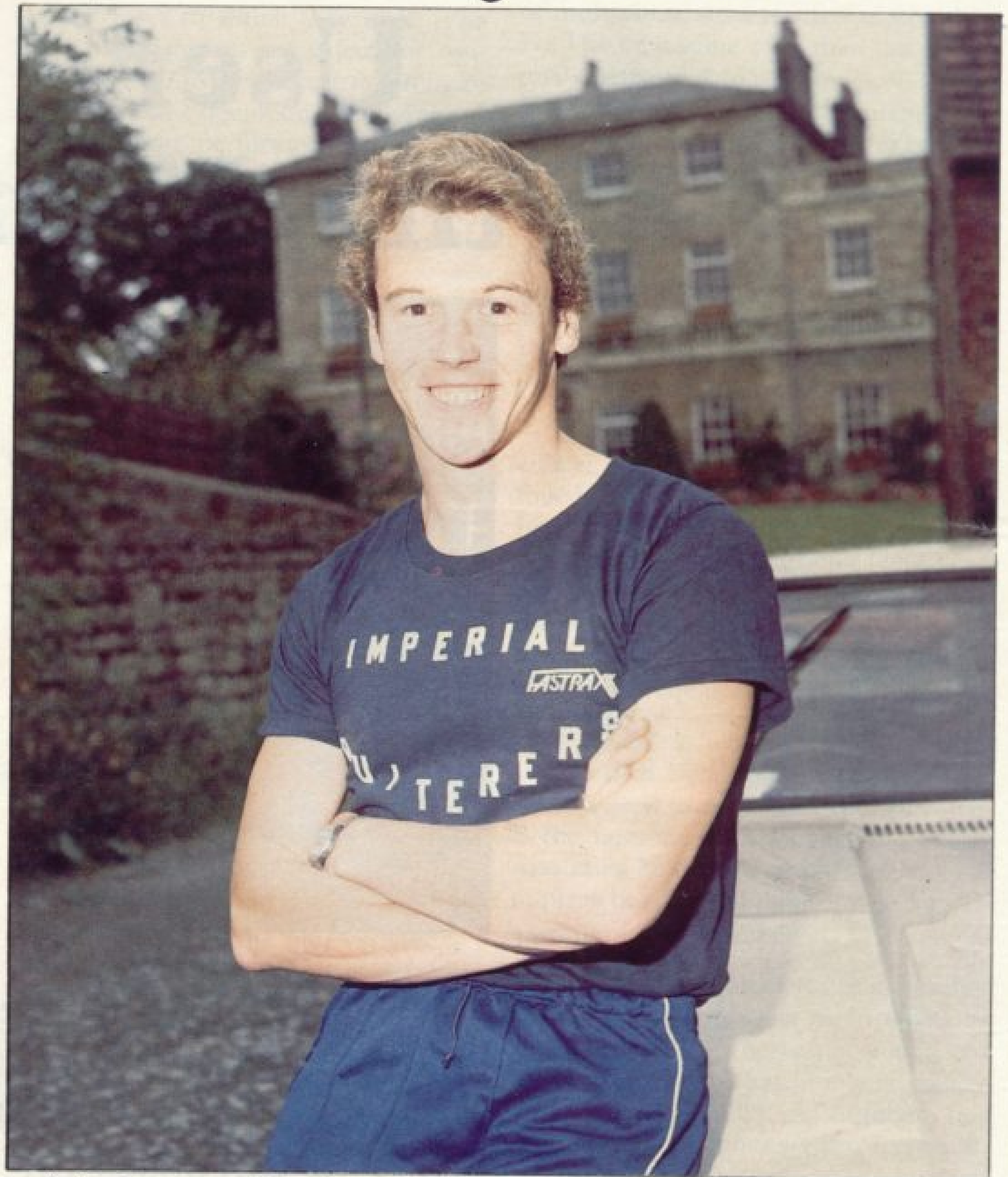
User of the Month



During the last 12 months we began a new series in which we interviewed interesting users. We reprint some of them to indicate the wide variety of purposes for which the Sinclair machines are used.

Stephen Sowerby is a member of the Olympic pentathlon squad. Claudia Cooke discovers how he makes time for training

Leading athlete's quest for gold is boosted by ZX-81



INTERNATIONAL athlete Stephen Sowerby swears by his ZX-81. Without it, he says, finding the time to train is much more difficult. At 28, Stephen hopes to make his Olympic debut as one of the four-man modern pentathlon squad in the 1984 Games in Los Angeles.

On top of that, he runs two businesses near his home in Yorkshire and spends what precious spare time he has on his hobbies of photography and wine and beer making. He says:

"I'm impressed with the little beast — the ZX-81. I bought it last December with the idea of upgrading it as soon as I had mastered it but I don't see any need now. It does everything I want it to do".

Before buying his computer, Sowerby had had no contact with anything more complicated than a pocket calculator but he decided that anything which could speed his office workload would be invaluable.

"First, I waded through the book and made it to work. Then I wrote my first program, a simple one for VAT checking which has been fantastic. I found that it was really easy if you used common sense. Of course, there were headaches along the way but I sat down and scratched my head and it all came out right.

"I am pleased to say that I have never bought a commercial program. I have written 10 of my own; the longest is for helping me to do the accounts. The amount of time it saves is incredible. I buy the computer magazines and use some of the programs in them. Apart from being fun they can be really helpful, too, because sometimes you discover you have been using five lines where someone else has discovered a way to say the same thing in only one line".

Running his business, as well as fitting in his sports activities, means that Sowerby is highly-motivated to do everything in as short a time as possible and the computer has helped.

"If I have all my work done by a reasonable time in the afternoon I can leave and fit in more training. So the computer is helping me in my sport, too". Modern pentathlon is a gruelling combination of sports, the aim being to find the best all-rounders. It embraces swimming, running, fencing, shooting and show jumping.

"Swimming was always my main sport at school. I won at county level and was in the national top 10 when I was 17. Then I went to Carnegie College, Leeds to train as a physical education teacher for three years and I stopped competitive swimming.

"My first job was as a teacher of PE, swimming and outdoor pursuits at Daniel Stewarts and Melville College, Edinburgh, where David Wilkie went to school. I thoroughly enjoyed it but after a year I had the chance to swim competitively again with my old club in Leeds and so I left teaching to give myself more time.

"I was trying to be selected for the Commonwealth Games and that meant four or five hours in the water every day. Then I entered the national biathlon — swimming and running. I had entered it and won while I was still at college and that time I won it again".

Sowerby is modest about that achievement, saying that although he was always the best runner at school, he assumed that was simply because he was fit from all the swimming.

The result of his success in the biathlon was that the well-known pentathlete, Jim Fox, suggested he should try modern pentathlon. He had never fenced before, never used a gun, and scarcely ridden a horse but after only four weeks' training, he entered his first pentathlon in Cambridge and finished second, despite the presence of seven international modern pentathletes.

Since then he has not looked back. He has risen to international status and travels the world to compete in events as far afield as the United States—three trips already—and several countries in the Eastern bloc.

"I love the travelling and there are so many places I know I would never have seen had it not been for the competitions there. We were in Poland at the start of the crisis and I learned a good deal from that.

"They absolutely revere their sportsmen. There were food shortages and queues everywhere but we had meat at every meal and the autograph hunters were all round us".

Sowerby is the only member of the four-strong Olympic squad who does not train full-time but he sees that as no bad thing.

"I have all the practice I need and in a way I think my job helps, because it takes my mind off things. If you are just training all the time you think only about your sport and it doesn't necessarily do you any good. You just become bored waiting for the next training session.

"I am glad I tried the modern pentathlon because I am so much happier now. You have to be good at all five events but if you don't do so well in one aspect, it's not the end of the world. When I was swimming, one race might make or break me.

"Now I know I am the worst fencer in the squad but I am training hard and my running and swimming are strong enough to compensate for it".

The businesses which Sowerby runs stem from the enterprise of his parents, Dennis and Elsie, who formed their joinery company at Knaresborough 30 years ago. As the business flourished, so more and more customers needed glass, so he formed Knaresborough Glass and now runs both companies with the help of his father.

Set in a small yard off the main street in the sleepy Yorkshire market town, with offices in a building known as the hen hut, it is a far cry from the glamour of the international sports circuit but Sowerby, who last year married a fellow member of his swimming club, is happy to return to his hen hut and sit crouched over his computer.

Since buying the ZX-81 he has also bought an Olivetti printer driven by a Memopak I/F which he keeps hidden under the dK'tronics keyboard.

"The printer has been a marvellous help, too. I have written a word processor program and I use it for all my letters, for final demands, invoices, everything really. I begin to wonder what I did without it".

Sowerby also sees potential for computers in the world of sport. Already they are used widely for results, as well as for much of the administrative work, but he cites a need for them in sports such as fencing. Each fencer's sword has a button on the end which lights at certain points of contact with the opponent and is controlled by a box mechanism. "That box mechanism often fails and as there is only one company making fencing equipment, there is not much we can do but a computer to operate that mechanism could prove far more reliable and I am sure someone could make money producing it".

'I have written a word processor program and I use it for all my letters. For final demands, invoices, everything really. I begin to wonder what I did without it'

Sowerby feels the project is a little ambitious for him, although he maintains that if you want to do something, you can usually do it even when, as in his case, it seems as if there will never be the time.

"I do a great deal with my time but I enjoy it all, otherwise I wouldn't do it. I will go on as a pentathlete until I stop enjoying the training. When it becomes a real chore and you don't want to do it, that's the time to stop.

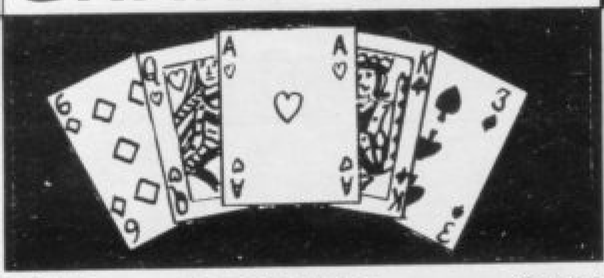
"Many people seem to think I must be sacrificing a great deal for sport but I don't see it as a sacrifice, because I enjoy it.

"The only things I miss are smoking 40 cigarettes a day and probably being overweight by the age of 18. So where's the sacrifice?"

Financially, Sowerby has to be self-supporting in his sport and that can mean sacrificing some expense — the Spectrum, for example, for which he sees no need in his office but on which he loves playing games for its colour and its extra graphics.

Any purchase not essential to him stays in the shop but that could be why both his business and his sporting career are proving so successful.

GRAND SLAM



NOW PLAY BRIDGE ON YOUR OWN WITH

MICROBRIDGE

THE PERFECT TUTOR FOR IMPROVING BRIDGE PLAY.

PRAISED BY EXPERTS AS THE IDEAL COMPANION FOR BEGINNERS AND ADVANCED PLAYERS.

Microbridge allows you to play Contracts all the time, at your own pace, in your own way, against a formidable defence.

Microbridge gives you an infinite variety of Contracts to practise your card play, in a way never before possible.

Microbridge contains special features to help you analyse your play as it progresses.

Microbridge does not get upset or lose its temper! It develops your confidence and skill in a remarkable way.

PLEASE SEND ME YOUR MICROBRIDGE CASSETTE FOR THE SPECTRUM 48K

NAME ______________

ADDRESS ______________

____

I ENCLOSE MY CHEQUE/P.O. FOR £7.99 PAYABLE TO 'STELLAR SOFTWARE' 144, PAMPISFORD RD, SOUTH CROYDON, SURREY CR2 6DA

BRIDGE SOFTWARE

SPECTROGRAPHICS

(for Sinclair Spectrum 48k)

"A complete graphics toolkit...very easy to use... documentation clear and concise"
(Personal Computer World)

For Teachers For Games Enthusiasts For Artists For Everyone!

Exploits the Spectrum's superb graphics potential to the full:

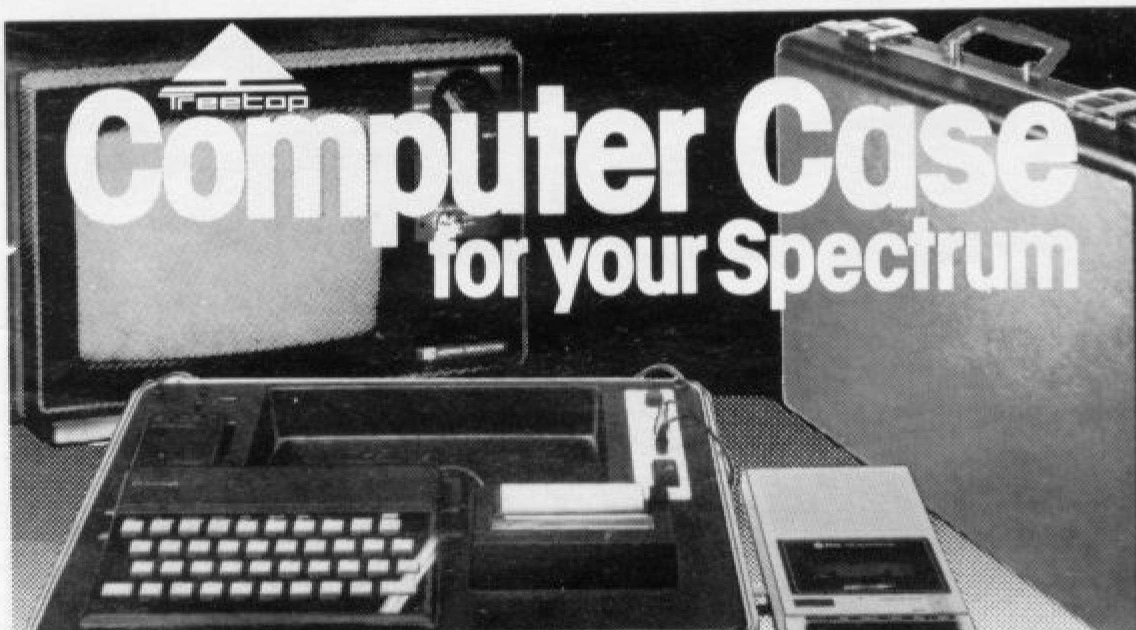
- 11 graphics procedures
- Hi- and lo-res sketch pads
- 8 sizes of text
- a mini-text editor
- colour WASH and PAINT
- create your own UDG characters
- SAVE displays on tape or PRINT on ZX Printer
- RELOAD displays into your own programs

send £6.90 for cassette and manual

graphics software for ZX81 also available

send s.a.e. for our full list of games, statistics, graphics and scientific software!
Full money-back guarantee : Trade enquiries welcome

BRIDGE SOFTWARE **Dept SA , 36 Fernwood, Marple Bridge, Stockport, Cheshire, SK6 5BE, England.**



- **Solves storage problems**
- **Computer angled for convenient use**
- **Avoids lengthy setting up**
- **Hides most of wiring**
- **Case and console moulded in black grained ABS plastic**

Treetop Designs, 61 Widmore Road, Bromley. Kent. BR1 3AA.

£44.95 (plus p & p)

Smart executive type case, purpose built, with locks, special hinges and lift-off lid.

Internal console allows neat installation of computer, power supply and printer in fitted recesses. Provision for housing tape recorders up to 28cm x 15cm.

Aluminium switch panel eliminates constant removal of plugs for both save/load and power off facilities and provides sockets for tape recorder leads, to allow operation in convenient position shown.

All internal wiring and leads supplied to allow easy plug-in installation.

Case lid fitted with shock absorbing foam to protect components when closed and with retaining straps for neat storage of mains and T.V. aerial leads.

To: Treetop Designs
Freepost, Bromley, Kent. BR1 3UZ

Please supply ________ cases at £47.45 each (incl. p&p and insurance)

My cheque/P.O. for £ ________ is made payable to Treetop Designs

PLEASE PRINT NAME AND ADDRESS

Mr/Mrs/Miss ______________

Address ______________

Please allow up to 28 days for delivery.

If you are not satisfied we shall of course refund your money, if you return the case undamaged within 14 days.

Each month we will interview a notable Sinclair user. Claudia Cooke meets our first User of the Month, a retired shipbuilding manager.

Retiring to the sea, the ships and his Sinclairs

ERIC ANSELL had been looking forward to retirement and the chance of a return to his native Isle of Wight. Two years ago, at the age of 65, he moved into a cottage near the sea at Cowes with his wife, also an islander, and intended to enjoy a quiet life of reading, walking and watching television.

"Never in a million years did I think I would learn something so new at my age", he says. Yet in the six months since his son gave him a ZX-81, he has become hooked on home computers, to the extent that he has just bought a Spectrum and a new tape recorder to go with it.

"My son bought a ZX-81 but then he moved to another computer and when he visited me last summer he asked if I would like the ZX-81.

"At first I wondered what I wanted with a computer. I could not see any use for it. I had no interest in computers and I did not realise how much fun they could be".

Ansell, however, is a man who believes in trying anything. "Once my son had left I started fiddling with it and became absolutely hooked in no time. It is the sheer logic of it which appeals to me," he says.

"You program a game from one of the books and it tells you that you have made two errors. Once you find those errors you realise they are so logical and obvious. It just takes common sense".

He admits it was two weeks before he managed to program a game successfully and he has not yet reached the stage of writing a program.

"When I bought the Spectrum two months ago, I found it much easier to start because I already had some knowledge of the ZX-81 and I have become much faster at typing-in now.

"I used to have two pipes of tobacco in the afternoon and perhaps read a little or watch television. Now I become so engrossed I forget all about my pipe and everything else and I usually spend about two hours, four days a week, with my Spectrum".

Ansell had had contact with computers during his work as naval planning manager with Swan Hunter in Newcastle before his retirement but although he did the critical path diagrams for the computer printouts, his contact was indirect.

"I was always rather dubious about large computers at work. If, for instance, something is proceeding satisfactorily on a ship and you put it through the computer to find the best way, you might have to wait five or six days for an answer but you can go and look at the ship and have it corrected in one day".

He spent 35 years working for a local shipbuilding firm at Cowes before moving to Newcastle when the firm closed. He began his career as a naval draughtsman and then became a chartered engineer before working as a naval shipbuilding manager. He loves ships and everything to do with them and says:

"What amazes me about the Spectrum is its ability to produce an almost perfect reproduction of, say, a destroyer, at the push of a few buttons".

> 'Never in a million years did I think I would learn something so new at my age'

Ansell is keen to scotch rumours that computers are for the young, or at least for the under-60s. He sees no reason why that should be so.

"I admit we are not so likely to have contact with them in retirement. I doubt that I should have discovered them if it had not been for my son but I would strongly advise any retired person with a reasonable amount of common sense to buy one straight away and play with it.

"It has given me an interest I never expected to have at my age and my brother-in-law and I spent the whole of Boxing Day together, playing golf with the Spectrum".

It must be said that he is and always has been an avid games player. Apart from being a crossword fanatic, he also subscribes to a weekly puzzle magazine and likes board games as well.

"I remember when my two sons were young, I would buy a new game every Christmas and we would all spend hours with it, sometimes making-up the rules as we went along which, of course you cannot do with a computer", he says.

As for progressing to even more ambitious activities, he is happy with his ZX-81 and his Spectrum, although he says he may consider a 48K Spectrum instead of his present 16K.

The next step, of course, is for him to begin writing programs, something he is characteristically keen to try.

"I realise I need more experience first and I am still enjoying the cassettes I can buy, like Escape and Othello, but I would like to program one or two games and I intend to try soon. I find a great sense of achievement in putting a program on tape and having it correct".

Ansell is a young-looking 67. He wears a smart blue velvet jacket and sits in an armchair within easy reach of a bowl of assorted pipes, as befits any ocean-loving man. On the other side is his Spectrum and across the room there is the presentation tray he received on his retirement from Swan Hunter, listing all the ships on which he worked.

The room is a stark mixture of the old and the new, a combination which Ansell has proved can work well together. All in all, computers have enhanced his life in a way he did not think possible and did not think he desired.

Retirement at Cowes, where he and his wife were brought up, attending the same school and marrying 43 years ago, is something of an idyll and Ansell admits that much as he enjoyed his years in Newcastle, there was never any doubt in his mind as to where he would set up his retirement home.

"I think that had I left the island when I was young, there might not have been the same nostalgia but I was 50 when we moved north and that is just too old to begin putting down new roots".

Now, surrounded by sea, ships and a host of books on both subjects, Ansell and his computers are enjoying themselves thoroughly.

Claudia Cooke interviews a West Midlands doctor who is finding ways of using a Sinclair computer to help educate his children

The smiling Spectrum is Sophie's first choice

SOPHIE DENT has found life enriched since the arrival of a Sinclair Spectrum in her home six months ago. It is not that she has spoken of that enrichment but speaking is altogether a problem, since Sophie is only 18 months old.

Her shrieks of delight are sufficient proof as she hurls herself towards the keyboard for another intense session with one of her two programs. Both were written specially for Sophie by her father, Dr Tom Dent, who shares Sophie's enthusiasm for home computers.

His other children, 10-year-old James and Rhian, aged seven, both fell in love with the Spectrum as soon as it arrived. It soon became apparent that the role of onlooker was too much for little Sophie; she wanted to play, too.

At first her father was not sure how he could write a program for Sophie—for pressing specific keys and refraining from resting an elbow on the others is difficult for any baby—but found the answer with two colourful and instantly-rewarding programs which operate whichever key is pressed.

The first allowed Sophie to produce a smiling baby's face on the screen, something with which she could identify and recognise from an early age. Her father, a 35-year-old general practitioner, explains: "Faces are the first things to which a baby responds and understands. It was a happy, smiling face and Sophie loved it".

The second baby-proof program is called Ghastly Graphics and has proved equally popular with the older children. Again, it operates by random pressing of the keys. Gradually it produces a mass of random graphics which eventually will fill the screen in a colourful pattern. Each graphic is accompanied by a note, making a cheerful tune which further captures Sophie's attention.

Dent admits that at that stage the Spectrum is just another toy to Sophie and an expensive one were it not also used by other members of the family.

He is already looking to the future, when Sophie is three or four, and hopes the Spectrum will be able to teach her to read quicker and more efficiently than either he or his wife, 36-year-old Janet, could do.

"The marvellous thing about a computer is that it has infinite patience. Where anyone else might be bored with a reading lesson after half an hour or so, it will carry on until Sophie has had enough; and when children are really interested in something, they can go on for hours.

"The computer also makes things much more fun; learning becomes a game. I have written two educational programs for James and Rhian and they love them. Yet if I showed them the same facts in a book they would tell me to get lost. Books do not tend to be fun".

One of Dent's educational programs is for learning tables, a tedious task for most schoolchildren. Yet James and his

friend both worked their way through the program during my visit with evident enjoyment, not to mention great accuracy.

The other program is on elementary physics, an explanation of the differences between gases, solids and liquids. A simple explanation of each is followed by a simple question to test the child's comprehension of what has just been read. An incorrect answer takes the child back to the relevant text, which may be read again for a better understanding.

There are no pictures, no tunes—only words. Yet the program has proved popular with the children in a way no physics text book could hope to do.

"If you want to know the answer in a text book, you have to look it up each time at the back of the book but here it tells you immediately whether you are correct. It also means you cannot cheat or skip pages as you can do in a book. It is programmed so that you cannot move on until you have understood and answered a question correctly on one particular section".

The Dent family does not use the Spectrum only for serious programs. They use many of the games programs on the market, too, and Dent was forced to admit that he is usually beaten by James.

He sees the implications of the computer generally as far-reaching. Already the health authority at Walsall, where he works in a group practice, has a computer which can recall children for vaccinations more accurately than human labour could do.

There is also the potential for computers to be used in many other forms of recall in medicine, such as women needing regular cervical smear tests.

Dent feels that if individual practices could have their own computers they could prove invaluable in assessing individual practitioners' performances. He has already written a Basic program as an experiment for patient use. It assumes that the patient is complaining of a stomach pain and asks a series of relevant questions to which the patient responds by pressing a key.

The computer assimilates the various responses so that Dent can press another key and be given one or several possible diagnoses.

I was invited to try the program and, being unable to remember the full details of any stomach pain I might have had, caused a rather confusing diagnosis which ranged from dyspepsia to gall bladder trouble. The point was that it

was fun and I think I would have thought so even if I had a pain at the time. It also seemed more private and I could take my time in pinpointing the exact spot of the pain without worrying that the doctor might become impatient.

The program is not something Dent will use with his patients; it is simply a test for himself at this stage.

"I think a computer could become very useful in this field. Certainly it would save some time if a patient were to answer the questions before going in to see the doctor. It is a fact that a doctor, being only human, cannot ask the same questions without some form of bias".

With a computer, a patient is given no hint of an expected answer, or the possible element of surprise at any given answer, but Dent is quick to point out that the computer is not in itself sufficient because it does not see the patient.

"The computer cannot possibly notice that the patient looks more tired than the previous week, for example, or seems to be depressed about something, but it might have a function in producing, very quickly, a list of possible diagnoses for a given complaint".

Dent, who lives with his family and his Spectrum at Streetly, near Sutton Coldfield in the West Midlands, trained as a doctor in London and has been practising at Walsall for five years. One or two of his friends in the area have their own home computers but he says his children know more computer owners than he does.

"By and large, children seem to love them. If they think a computer is fun and it helps them to learn, I cannot see anything wrong. I certainly do not feel it is indoctrination or anything. When a child becomes bored with the computer, he has only to switch it off and walk away".

'The computer also makes things much more fun; learning becomes a game'

Dent is hesitant about predicting the success of a computer as a replacement for school because of the social implications.

"It is a good teacher but I think the children would miss school friends, football and so on. We hope that Sophie might learn from the computer to read and perhaps to write before she begins school.

"If there were no school in the area which seemed acceptable to us, I suppose we might consider teaching her at home but that is not the case and, anyway, I think we would make that decision with or without the computer".

At the moment the Spectrum is Sophie's toy, one of many. She understands that to press a key will produce something entertaining on the screen. She does not yet know that it can make a difference which key you press and she certainly does not understand that tapes were not designed for unravelling with a little finger and trailing across the floor but given that she is still a baby, her familiarity with computers should make all those things far easier to learn in the next few years.

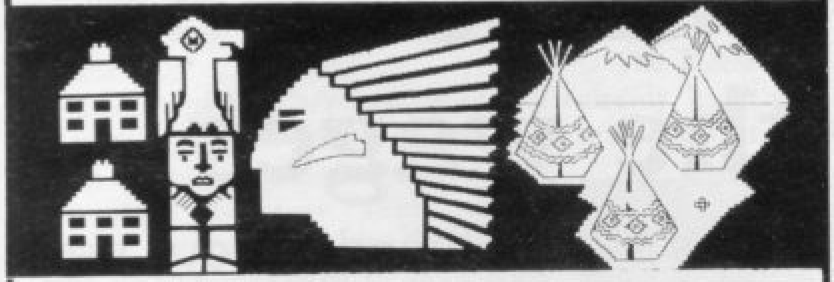
Her seven-year-old sister already enjoys entering programs, with a child's unquestioning acceptance of the repetitive details required in instructing a computer—and she started only six months ago.

The Dent family has found a toy, a teacher and a potential secretary all in one. The next stop will be to buy an interface to enable them to use the Spectrum as a word processor.

With an age range spanning one-third of a century, they have found something from which each can benefit. As a family they have been converted.

If readers would like to nominate people for User of the Month, please write to Sinclair User, *196-200 Balls Pond Road, London N1 4AQ giving name, address and daytime telephone number and the reasons for the nomination.*

SPECTRUM & ZX-81 EDUCATIONAL SOFTWARE



CALPAC LEARNING SERIES

Use our Computer Assisted Learning PACks to help your children with their school work. The programs in the series use moving colour graphics and sound to make learning more enjoyable. Each pack contains four programs and is suitable for use with the 16K or 48K Spectrum. Program notes are supplied

VOLUME 1 (from 6 years) £9.50
Tens and units addition and subtraction with detailed help facilities; powerful, easy-to-use drawing program; English comprehension applied to American Indians.

VOLUME 2 (from 7 years) £9.50
Roman history; spelling tester with easy entry of your own lists; flexible fable-tester; homophones.

VOLUME 3 (from 9 years) £9.50
Nouns, verbs, adjectives and adverbs; tenses of verbs; biology of the flower; long division tutor.

Additional features of the CALPAC LEARNING SERIES include:
*Spelling checkers
*"Help" call up routine
*Easy insertion of subject material of your own choice into the programs

CALPAC C1 O-LEVEL CHEMISTRY £7.50

Four clearly presented revision/tutorial programs. The subject matter has been carefully structured to cover the most important aspects of:
★ Elements, compounds and mixtures.
★ Structure, bonding and properties.
★ Redox, electrolysis and the activity series.
★ Acids, bases and salts.

48K Spectrum and 16K ZX-81 versions of the cassette are available. Please specify which you require.

Our software is available by direct mail from:
CALPAC COMPUTER SOFTWARE
108 Hermitage Woods Crescent
St Johns, WOKING, Surrey GU21 1UF
For further details please telephone: 048 67 2584
We have a demonstration cassette available for retailers or schools

TAKE A BREAK PLAY WITH EDUCATION

New games for secondary level learning and revision

AKADIMIAS FOUNDATION SERIES

Programs especially written by University Academics and teachers for the Spectrum 48K.

Choose between various time based skill levels to answer questions in your subject.

English History
European History
English Literature

Ask us for a catalogue:
Sussex Software
FREEPOST : Devizes SN10 1BR
Wiltshire

SUSSEX SOFTWARE

Every computer needs
CHATTERBOX II
"Listen creep, I am the leader....."

For **ZX81**
SPECTRUM
BBC
TRS 80
APPLE
NASCOM
VIC/PET/64
(Please state)

NEW! EXCLUSIVE! PITCH CONTROL

CHATTERBOX II™ can say anything!

Genuine phoneme synthesis - not just recorded speech - hence unlimited vocabulary. Programmable pitch for more natural intonation (exclusive to Wm Stuart Systems)- solid tone cabinet for quality sound - integral beep/music amplifier. **PLUS** expansion socket for BIG EARS voice recognition system. Full instructions technical notes and software supplied with this outstanding educational unit.
DEALER ENQUIRIES WELCOME

As seen on BBC TV "Computer Programme"

BIG EARS

SPEECH INPUT FOR ANY COMPUTER

Hugely successful Speech Regnition System, complete with microphone, software and full instructions.
BUILT TESTED & GUARANTEED
PLEASE STATE COMPUTER: UK101, SPECTRUM, ATOM, NASCOM2, Vic 20, Micron, ZX80/81, PET, TRS80, MZ80K, APPLE II, BBC MICRO

ONLY £49

ZX81/SPECTRUM

NEW! VIBRATO CONTROL

MUSIC SYNTHESISER (Stereo) +16 LINE CONTROL PORT

Play 3-part music, sound effects, drums etc. Full control of attack, decay and frequency.
Input/Output lines provide control and monitor facility for Home Security, Robot Control, Model Railway etc. etc. Works with or without 16K RAM.

Full instructions/software included.
Add keyboard to make a live performance polyphonic synthesiser!
Note: up to 3 units can be used simultaneously; giving 9 music channels & 48 I/0 lines

AMAZING VALUE AT ONLY £19.50 (KIT) £25.50 (BUILT)

SOFTWARE

THE COMPOSER
Synthesiser Music Programme.
Enter & play 3 part harmony.
Includes demonstrations.
(Spectrum/ZX81) **recommended** £7

TALKING HANGMAN
For Chatterbox!
The classic game claims its victims with a dry vocal accompaniment (Spectrum) £6

ZX ARP/DRUMSEQ
Fascinating synthesiser demonstrations. Generates automatic sequences and plays from keyboard. Some weird effects (Spectrum) £6

CHROMACODE??????
Can you defuse the bomb by cracking the secret combination before time runs out? With Chatterbox voice output (Spectrum) £6

COLOUR MODULATOR KIT £16 BUILT £22
RGB in, PAL/UHF out (not for ZX)

Please add VAT at 15% to prices.
Barclay/Access orders accepted by telephone

All enquiries S.A.E. please

WILLIAM STUART SYSTEMS Ltd
Quarley Down House
Cholderton
Nr. Salisbury
Wiltshire. SP4 ODZ
Tel: 098 064 235

Taking the strain out of calculating the wages

Starting with the ZX-80 Ronald Sims has always found serious uses for his machines. Claudia Cooke talks to him about how they help.

NO-ONE was more pleased than Vera Sims when her husband Ronald invested in his first computer, a ZX-80. For the first time in almost 20 years, her workload was reduced from a day-and-a-half each week to only four hours. Mrs Sims has the job of calculating and paying the weekly wages for the 35 employees of the family automation business.

"I used to do it all in my head, with the help of a ready reckoner and it took me a day and a half. Now my husband has written a program which does it all for me and it takes a few hours. It has been marvellous".

Her husband has since changed to a ZX-81 with a 64K Memopack and the program covers job costing for up to 10 jobs, tax deductions, a variety of individual allowances and more than 30 subtotals and totals.

Mrs Sims, 69, says: "It was a bit strange at first because I didn't know the first thing about computers but I soon got the hang of it and now I'm really pretty quick".

Husband Ronald, 70, interrupts to reveal that the early days were not that simple. "She just wouldn't believe it at first. When the computer produced the figures she used to take them down quickly and work it out herself to make sure. I must say her calculations were as quick as those of the computer.

> 'The computer just cannot make the kind of human errors everyone makes'

"Now she realises that the computer just cannot make the kind of human errors which every person is bound to make at some time or another".

Although the Sinclair is his first computer, Sims is no newcomer to the processes of logic involved. He had worked in the radio industry since 1929 and had become fascinated by 'wirelesses' even before he left school at the age of 17.

"At school we had a physics master who was very keen on the wireless, as it was known in those days. I was charged with his enthusiasm and by the time I left I was one of few people who knew how they worked."

His first job was with a firm selling do-it-yourself wireless kits. People would buy the kits on Friday, he says, and return on Monday to find what they were doing wrong. His job was to advise customers exactly how to assemble them.

"In the early 1930s I realised my technical education was not keeping pace with the changes so I got a job with Siemens in telephone exchange development work. The logic then was the same as the logic of computers today and I was given a good training".

When the second world war broke

out, the company had just begun work on the development of radar and Sims became chief of test gear.

"There was such urgency because of the war that we had to start manufacturing in quantity products which had not been made previously. Being responsible for testing, it was the first time I really became interested in automation and automatic testing."

After the war, Sims left to join Michael Sobell, first as technical assistant to the buyer and then as chief buyer. He stayed with the company for 12 years and says that without Sobell's commercial guidance he could never have started his own business.

Start it he did, 25 years ago, when he detected a gap in the industry for automation. Today his company, Lectromec Controls, is still flourishing in south London, run by his three grown-up sons and his ZX-81.

He is working on a cashflow program for the firm, although he is doubtful whether his Memopack can cope with a sufficient number of invoices.

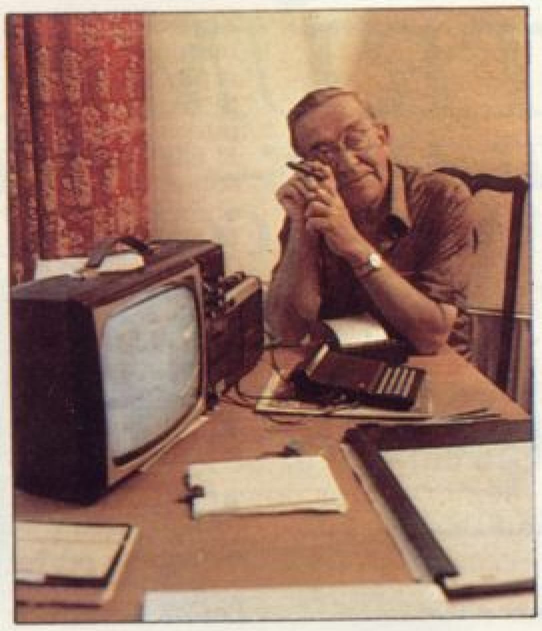
His wages program took many months to compile but the result has been worthwhile. Not only does it do the job much faster, it rules out the necessity for Mrs Sims to have her cash sheets checked by someone else before handing-out the pay.

"If you make a mistake you just look at the picture and know if it's right or wrong", says Mrs Sims. "The only thing is that you lose your brain, because you just look at the picture and copy it all down. I used to have to make so many calculations in my head but now I almost think you could become brainless within a few years."

Sims has one criticism of Sinclair, levelled at the printer. "I think it was very ambitious to produce this printer but I think it leaves a lot to be desired. It prints so badly; I think it's the thermal paper. If you leave it switched on all day, which we might need to do often, the whole thing gets warm and the print blurs".

On the whole, though, he sees his computer as a great boon and not only for work purposes. "We have four grandsons and you should have seen them with all those computer games at Christmas. We couldn't drag them away from the set. We all enjoyed it".

The eldest grandson, aged 12, has already had the ZX-80 bequeathed to him, together with the 16K memory.



He is at least as enthusiastic as his grandfather.

Now Sims is working on a conversation program to use with his grandsons at this year's Christmas gathering. It is a marvellous program so far, tailored individually to each of the four boys so that if one gives the computer his name, it is liable to answer back with the name of his brother.

A selection of 10 answers to each response from the boys appears at random, resulting in some highly-entertaining *non sequiturs,* as well as some startlingly accurate replies.

When asked how I was feeling by the computer, I replied "very hot", only to be told rather cheekily: "You look all right to me". When I agreed with the computer's suggestion, "I hear they call you Podge", it replied curtly. "Stop mucking me about".

Sims hopes to develop the program to around 10 or 15 minutes by Christmas and is gaining evident enjoyment from the work involved.

Aside from the computer, he and his wife are kept busy at their home in Ewell, Surrey. Mrs Sims is a regular bowls player, although sadly out of action during my visit due to a back injury. Her husband likes nothing better than to sit down for an hour or so and play the organ. He is from a musical family and as a child learned to play a variety of instruments, including the piano, violin, flute and piccolo, but his favourite remains the organ. "I particularly love playing classical music. I have always found it relaxing. To return home and get lost in the music for a time is a wonderful way of recovering from work".

'I think it was very ambitious to produce the printer but it leaves a lot to be desired; it prints so badly. If you leave it switched on all day the thing gets warm and the print blurs.'

The Sims also have a narrow boat, moored on the Thames, to which they retreat whenever possible during the summer months to enjoy the sun and the fresh air.

"Mostly we take it on the Thames these days, rather than the canals, but it is a wonderful way of relaxing. You don't have to arrange it all in advance; you just decide to go and that's it. It's a kind of freedom which is very valuable in business because, contrary to popular opinion, you cannot just take-off on holiday overseas whenever it suits you. You never know what will happen until the last moment".

Both are now semi-retired and the father figure wisely attempts to keep his nose out of the family business to a great extent, leaving his sons to run it the way they wish to do.

"I never pushed them into it, or expected too much of them. They all worked for other firms for a few years before joining me but I must admit it is pleasant that they all wanted to join. Now one is the managing director, one is sales director and the third is really installation manager".

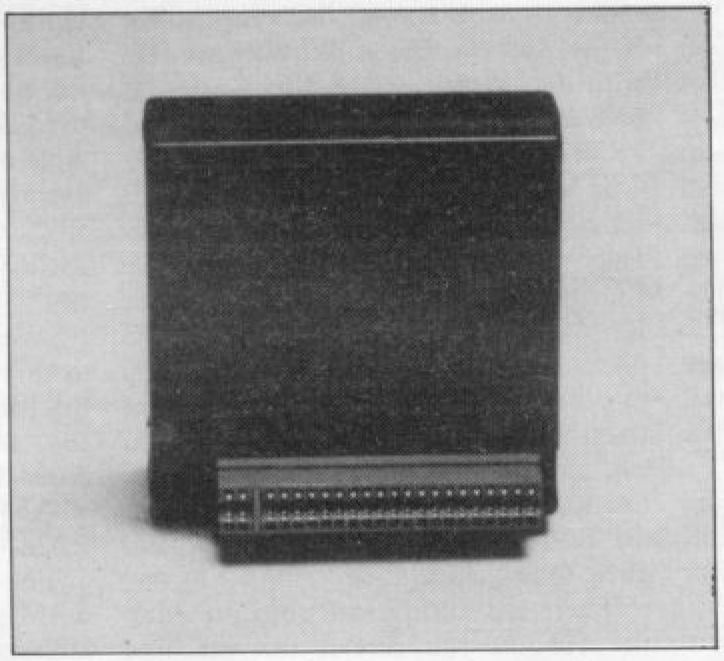
One of his sons has also become a highly-proficient organist, playing in a dance band. His father pours generous praise on his talent.

Married for 43 years, the Sims are a contented couple, although not content to sit back and do nothing. Theirs is an active life, made more so by the advent in their home of the computer.

They certainly defy any theory that computers are for the young. It was Mrs Sims who, having read *Sinclair User,* became the first person to nominate someone for the title of User of the Month — her husband.

She nominated him not only for his wages program but also for the fact that he derives so much enjoyment and relaxation from compiling the programs. His enjoyment of making the programs, she says, is almost greater than his enjoyment of operating them in his computer and although she steers clear of programming, she has learned a new skill at an age when few women expect to do so.

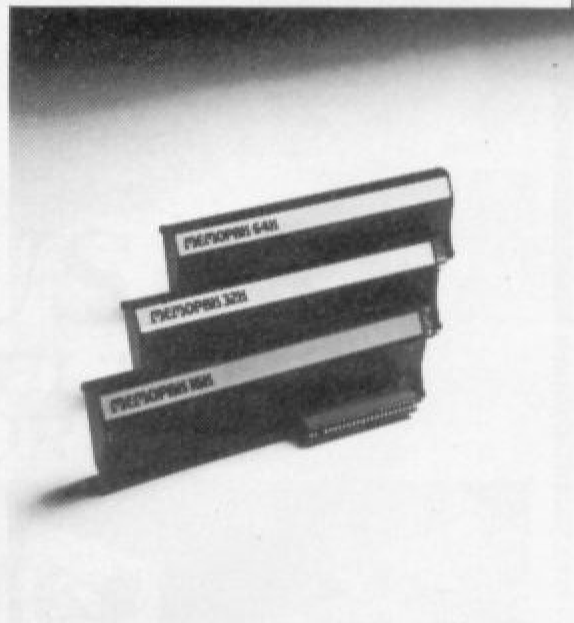
Memory



Expanding memory on both the basic ZX-81 and the 16K Spectrum is a major concern to Sinclair owners. Stephen Adams reviewed the expansions systems available for both machines and the results are re-printed here.

Cheap is not nasty in memory packs

Stephen Adams examines the many ways of expanding the storage of the ZX-81.



THERE ARE so many ZX-81 RAM packs around these days that it is worth looking at what they can offer in the way of more facilities.

First, a little explanation of the ZX-81 memory map and ROM routines would help in understanding the limitations of the ZX-81.

Sinclair engineers, when they designed the ZX-81, did not expect that anyone would need more than 16K of RAM. So they took some short cuts in the design which made it cheaper and easier to access the RAM and ROM. One of the them was to restrict the upper 32K of the memory map to working the screen and nothing else. The other was that the ROM, which is only 8K long, was allowed to repeat itself throughout the memory map unless the RAM was working in that area.

Those decisions allowed them to decode only the top two ADDRESS lines—A14/A15—to determine whether RAM/ROM or RAM with DISPLAY was on at any one time, the address line A15 deciding whether the display was on or not and the display being on only in the top 32K of memory and thus dividing the memory map in half.

The A14 address line divided each 32K section into two quarters, the bottom half of which was ROM and the top half RAM. As they were the only decoding done inside the ZX-81 the 1K of RAM repeated itself all the way through the 16K RAM section.

Memory map of the ZX-81

64K	
	RAM for display—display file
48K	
	ROM appears again
32K	
	1K or 16K RAM
32K	
	1K or 16K RAM
16K	
	ROM appears again
8K	
	True ROM area
0K	

When the 16K RAM pack is added, however, it is switched-off by the use of the $\overline{\text{RAMCS}}$ line on the edge connector and the RAM chips in the RAM pack do all the extra decoding necessary to divide the 16K section into individual bytes.

To expand the memory above 16K required some special decoding in the RAM pack so that it did not interfere with the display, which needs a repeat of the first 16K of RAM at 48K and above in the memory map.

It also required that the ROM be restricted to the first 8K of the memory map. That was done by using the $\overline{\text{ROMCS}}$ line in the same way as the $\overline{\text{RAMCS}}$ was used on the 16K RAM pack. Memotech was the first company to produce a 56K RAM pack. Most of them are now called 64K packs but you can use only 56K and that allows you to use 48K for Basic and 8K RAM where the ROM used to be, between 8K and 16K, for machine code.

There are some restrictions on using the 48K as you cannot run machine code in it and you have to be careful that the display file does not cross the 32K border. Memotech now has 16K and 32K RAM packs in the same boxes as its 64K ones, which are often used by dealers as a second choice to the Sinclair 16K RAM. Memotech has a good reputation for service. The 32K pack allows you to use your 16K RAM pack as well to achieve 48K.

The boxes are made from extruded aluminium and are used as a heatsink for the internal +5V regulator. RAM wobble problems should be familiar to all readers and Memotech is no exception but the company provides a Velcro strip which binds the packs to the ZX-81 and other packs. The instructions are clear and concise in a booklet accompanying each pack but they are expensive.

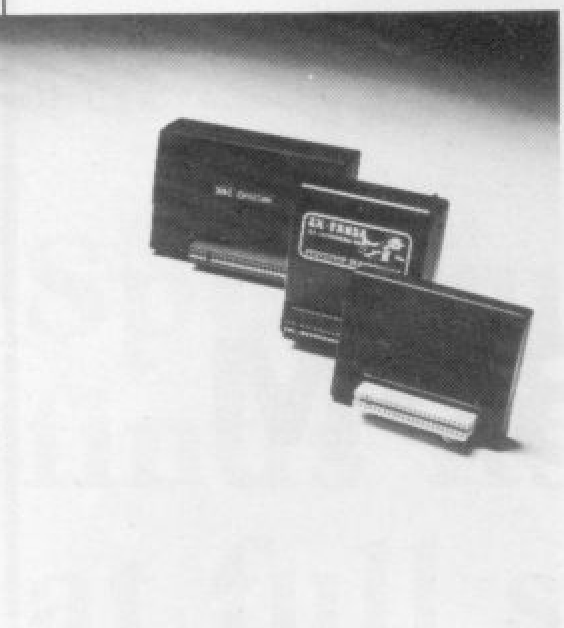
The Sinclair 16K RAM pack has received both complaints and praise. The complaints are from users who have had to experience RAM pack wobble, which occurs when using the Sinclair keyboard. The design of the RAM pack is such that it is not connected firmly to the ZX-81 and the top of the RAM pack rests against the top of the ZX-81 case. Every time the keyboard is used that lifts the RAM pack from the table and shakes it. The edge connector eventually becomes so loose that any slight movement will disconnect it from the ZX-81, corrupting the ZX-81 memory.

Several methods have been produced to stop the wobble — flexible cables so that the RAM pack can be laid flat on the table from dK'tronics and the RAM-LOK from Adapt Electronics which bolts together the ZX-81 and RAM.

The best method, though, is to buy a RAM pack which has been designed to eliminate the wobble. They consist mostly of not allowing the circuit board to be connected to the casing of the RAM pack, so that the board can move up and down inside the case with the movement of the ZX-81. A reliable stiff edge connector is also required. Some, like that from Cheetah, have also been designed to wrap round the back of the ZX-81 so that there is little movement between the two casings. Some manufacturers are also using +5V-only RAMs so that they do not have to provide a power supply from the +9V supply which can vary depending on the equipment used.

Kayde, Camel and Econotech 16K RAMs are good examples of the first type, the first two being cased. The ZX-Panda and the Cheetah are good examples of the second type.

The ZX-Panda can also be used to give 32K by adding a small PCB containing 16K more RAM inside the case. The Audio Computers RAM pack provides a fairly stable 16K RAM pack which has a hook to hold it on to the



back of the ZX81 and can be fitted with an XROM pack which makes it even more useful as it provides EPROM loading and saving facilities. Some of the RAM packs provide a LED — a little red light — but that only provides an indication that there is power to the RAM pack and not whether it is working correctly.

The best choice is obviously a non-wobble RAM pack which is inexpensive. If you can afford it and do not want to use pre-programmed ROMs or other devices, opt for a cheap 64K RAM pack. They take up just as much power as the 16K and will also work with the Sinclair printer — see the table for a list of facilities of the various RAM packs.

Basicare provides a different kind of RAM; you must first buy a base module to use its RAM, called a Persona. It is in 16K and 64K packs for the ZX-81 up to a maximum of 512K. That is divided into banks and you will require other modules to have the banks talk to each other. The maximum memory available at one time is 32K. It is the only system which allows you to run big programs but at the moment there is very little software to run it.

Yet another type of RAM pack is available and that is the battery-backed RAM — DROM, which allows the user to do things which normally would have to be run in from tape — instantly. They can provide storage for routines to side-scroll the screen in any direction, assemblers, new operating systems to replace the Sinclair ROM or even your own favourite Basic program.

A multitude of DROM packs is produced by Camel Products. Two of them work on the ZX-81, the Memic 81 and the Cramic. The Memic 81 contains one or two static 2K by one-byte chips giving 2K or 4K of RAM backed-up by a battery when the power is off to preserve its memory. That occupies any position in 8K-16K area of the memory map. DROM devices allow you to write a machine code or a Basic program and transfer it into the 8K-16K area of the memory map. Machine code can be run from there, saving valuable RAM space, but Basic programs must be uploaded to the Basic area before running the program.

That means that any Basic program is limited to 8K unless the Cramic is used which is 16K long. The routines for the transfers are included in the notes with every pack and consist of machine code routines which must be run by using the USR command.

The Cramic is a special case as the 16K can be switched in or out by a software switch and so a Basic program can be loaded from tape into the Cramic and sealed-off from the ZX-81. Powering-off the ZX-81 then has no effect, as the RAM module has its own battery to keep it operating. The program can be restored by powering-up the ZX-81 and running a machine code program to step back in the Cramic where you ceased. That leads to my first criticism of the notes; nowhere is it mentioned that you should save the machine code program on tape, in case something goes wrong.

The code required is very short but as it needs to be put into memory before using Cramic it would have been better to load it from tape. That routine could be stored in the Memic 81, of course, which is outside the Basic area and the routine loaded into the Basic area from there. The advantage of DROM over ROM is that it can be changed.

The other DROM is available only in kit from from Hunter Electronics and consists of board which plugs into the back of the ZX-81 and provides 2K of battery-backed RAM, with space for three more chips. If you wish, some of them can be turned into ROM sockets to take 2K or 4K EPROMs.

The RAM pack race has now become so cut-throat that a few firms which were doubtful have withdrawn. The firms remaining are giving the customers what they want, depending on price. Cheap is not necessarily nasty any more. So look for the bargains — they are there to be found.

DROMS

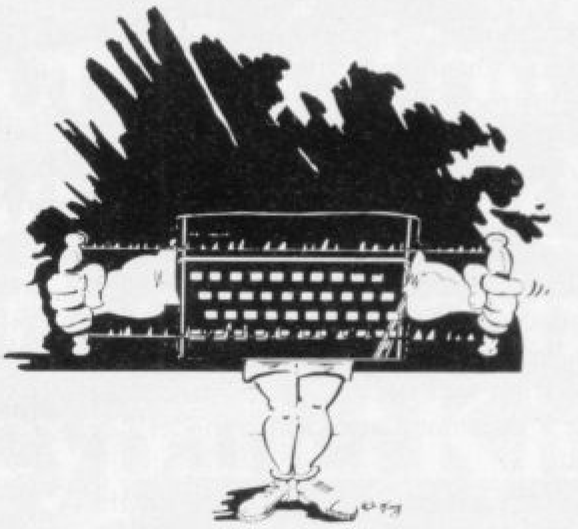
Memic 81 4K	**£34.45**
Cramic 16K	**£91.95**
Basicare 2K	**£39.50**
Hunter 2K	**£19.95**

SUPPLIERS

Basicare Microsystems Ltd, 12 Rickett Street, London SW6.
Cambridge Microelectronics (Camel), 1 Milton Road, Cambridge.
Cheetah Marketing Ltd, 359 The Strand, London WC2.
JRS (Econotech), 19 Wayside Avenue, Worthing, Sussex.
dK'tronics, Unit 2, Shire Hill Ind. Estate, Saffron Walden, Essex.
Audio Computers (Solldisk Ltd), 87 Bournemouth Park Road, Southend-on-Sea, Essex.

RAM pack	Anti-wobble	+5V only	Cased	Price
Basicare 16K	✓		✓	£26.75
Basicare 64K	✓		✓	£76.25
Camel 16K	✓	✓	✓	£20.64
Camel 64K	✓		✓	£80.45
Cheetah 16K	✓		✓	£19.75
Cheetah 64K	✓		✓	£44.75
Econotech 16K	✓			£20.95
dK'tronics 16K	✓		✓	£22.95
dK'tronics 64K	✓		✓	£52.95
Audio Computers	✓	✓	✓	£19.85
Memotech 16K			✓	£29.90
Memotech 32K			✓	£49.95
Memotech 64K		✓	✓	£79.00
Sinclair 16K			✓	£29.95
ZX-Panda	✓	✓	✓	£19.95

Sinclair has been overtaken by other suppliers of upgrades. Stephen Adams reports.



Spectrum finds itself at full stretch

THE SPECTRUM is in four basic forms. Model 1s which have two IC-type sockets into which a printed circuit board is plugged and model 2s which have all the RAM fitted on to the main printed circuit board. There are, of course, two types in each category, the 48K and the 16K.

The 48K cannot be expanded, as all the existing memory space is covered with either ROM or RAM — all 64K of it. The ROM can be switched-out externally to add different ROMs using the ROMCS but the RAM cannot as there is no equivalent RAMCS.

Therefore all memory expansions can be done only to the 16K models. As the computer already contains 16K of RAM, the top 32K of memory area — 32K-64K — is the only space to put it. Most memory expansions fill that space with 32K worth of RAM chips but the East London Robotics 64K add-on — the SP80 — has two sets of 32K RAM which can be switched in and out under a program instruction or from the keyboard.

The kits consist of four memory-decoding chips, except model 1 versions where all the chips are soldered to a PCB, and eight 32K by one-bit chips. Those RAM chips are very sensitive to static electricity and warnings are included in all the instruction sheets. The way to handle them properly is not to remove them from their protective packaging until needed and to keep touching an earthed object such as a radiator or gas pipe occasionally to release the static charge. The static can be caused by nylon or wool in your clothing being rubbed.

Putting-in the chips can be a little difficult and it is recommended that you check that the pins are straight before you insert them. The Fox Electronics kit has the pins already straightened. Also check that none of the pins is bent outside the socket or underneath the chip when you have finished.

The best technique to use is to put all the pins on one side into the socket and then to pull the pins on the other side over the holes in the other side of the socket. Once both sets of pins are resting in the socket holes you can push down gently on the chip to push it into its socket.

The instructions vary from a four-page, step-by-step meticulous description by Delta Research to one page just listing where the chips go by Fuller. Apart from the Fuller instructions all provided sufficient detail to allow you to know where each chip goes and what precautions to take.

As part of its instructions Fox Electronics also includes a sheet showing how to tune the colour on a Spectrum to give better results. It also shows you how to tell whether it is the Spectrum or the TV set causing a problem. Sheets normally cost £1 from Fountain Electronics but are free with its RAM kit, along with a small Basic program to demonstrate the usefulness of 48K.

> 'The 48K cannot be expanded'

The fitting of the Fox kit was the easiest but if you have problems you can send back the kit with the details of the Spectrum at any time in the following 12 months. East London Robotics will fit the chips at an extra cost of £7 by post or £3 by personal visit.

Delta sends a memory test tape which checks all the memory by using a machine code program going through at least all the manufacturer's specifications. The tape can be used to keep an eye on suspect memory faults as it stops with an error which can then be sent back to Delta or Sinclair. The tape normally would cost £3.50 and is a really comprehensive test taking at least eight minutes.

East London Robotics and Delta were the only model 1 boards we could test and both could be fitted easily with less trouble than the model 2.

East London Robotics also does an SP80 kit which can provide 64K of memory in the same sockets as the SP48 (48K) fitted. That is achieved by having two separate banks of 32K, switched by an OUT instruction.

That is possible only because of the 64K RAM chips which are used instead of the 32K RAMs and a massive modification of the decoding chips which plug into the Spectrum. No soldering is required but some care needs to be taken inserting the chips, as three of them are wired together using twisted insulated wire.

A LED is also soldered on to one of the chips to indicate which bank is being used. The only problem is that you can see it only with the top off or looking through the edge-connector hole. The LED lights when it is in bank two and when the machine is turned on the light shows on bank one.

The only problem with the SP80 is that the stack and any program running in it must be in the 16K of memory provided by Sinclair. If it was allowed to go into the top 32K it would be switched-out on the first OUT instruction and the program would have nowhere to go.

Also the machine code stack for return addresses would have to be in the lower 16K area for the same reason. That leaves the user to develop a program which will be able to use banks of memory without having the convenience of Sinclair Basic to cope with it.

Prices of the various kits are detailed, along with an indication of how good the instructions were, guarantee period and number of tests performed on memory after fitting.

It is also possible to add a RAM pack on the back of the Spectrum. A Spectrum-type 32K RAM pack has been produced by Cheetah Marketing Ltd. It fits very snugly on the back of the Spectrum and because it is outside the case it does not matter if it is model 1 or model 2.

It also has an extension piece on the back so that you can plug in Microdrives when they arrive. The RAM pack has no known wobble in use and can be fitted very easily by plugging it into the expansion port. It costs £39.95.

The alternative is to use a ZX-81-type RAM pack with an adaptor to give you another 16K or 32K. To use it you will require an adaptor.

EPROM Services and myself both make adaptors for the 16K RAM pack. Mine can also cope with a 64K RAM pack to give a full 32K by changing a soldered strap on the board or ordering a 64K version. I also produce an Adam II which lets you have a 16K RAM pack on the back of the Spectrum but which also allows devices which used to work in the popular 8K-16K region to work in the 56-64K region of the Spectrum. All the adaptors mentioned cost £9.

Of the RAM packs and kits reviewed. Fox Electronics for kits and Cheetah for RAM packs stand out as good value for money.

Fox Electronics, 141 Abbey Road, Basingstoke, Hampshire. Tel: 0256-20671

Cheetah Marketing, 359 The Strand, London WC2 0HS. Tel: 01-240-7939.

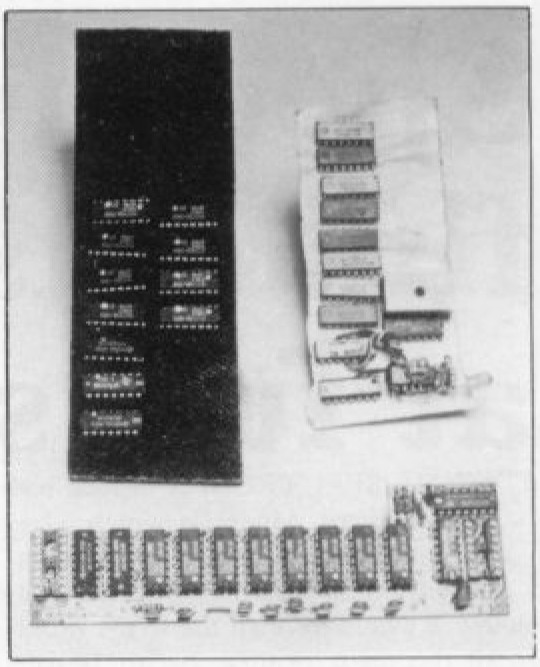
East London Robotics, No. 11 Gate, Royal Albert Docks, London E16. Tel: 01-471-3308.

Delta Research — cheques to Servodata Ltd — 15 Church Street, Basingstoke, Hampshire. Tel: 0635-45373.

Fuller Micro Systems, 71 Dale Street, Liverpool 2.

Eprom Services, 3 Wedgewood Drive, Leeds LS8 1EF. Tel: 0532 667183.

Stephen Adams, 1 Leswin Road, London N16 7NL.



Producer	RAM	Instructions	Model	Guarantee	Tests	Price
Fox	32K	Good	2	12 months	1	£24
ELR	32K	Good	2		1	£23.65
Delta	32K	V. good	1		>100	£33
Delta	32K	V. good	2		>100	£26
Fuller	32K	Poor	2		1	£24.30
ELR	32K	Average	2		1	£23.65
ELR	64K	Average	1		1	£50.65
E'	64K	Average	2		1	£46.65

Advertisement Index