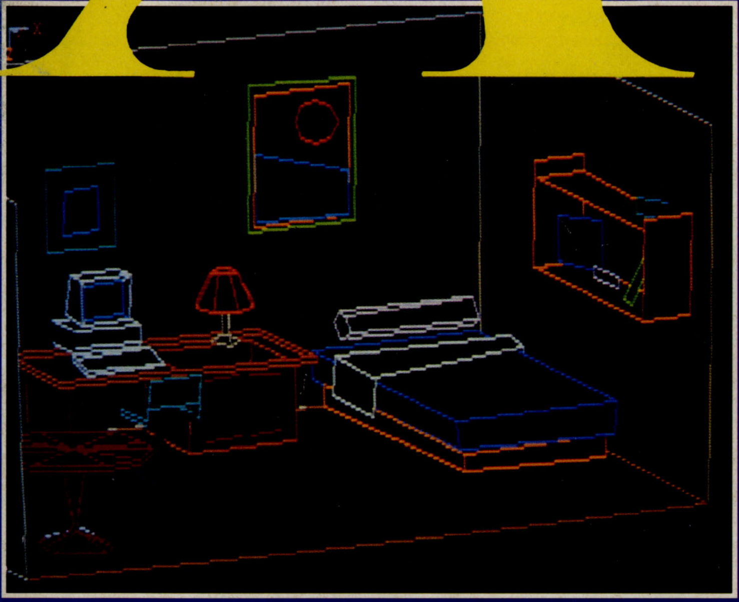
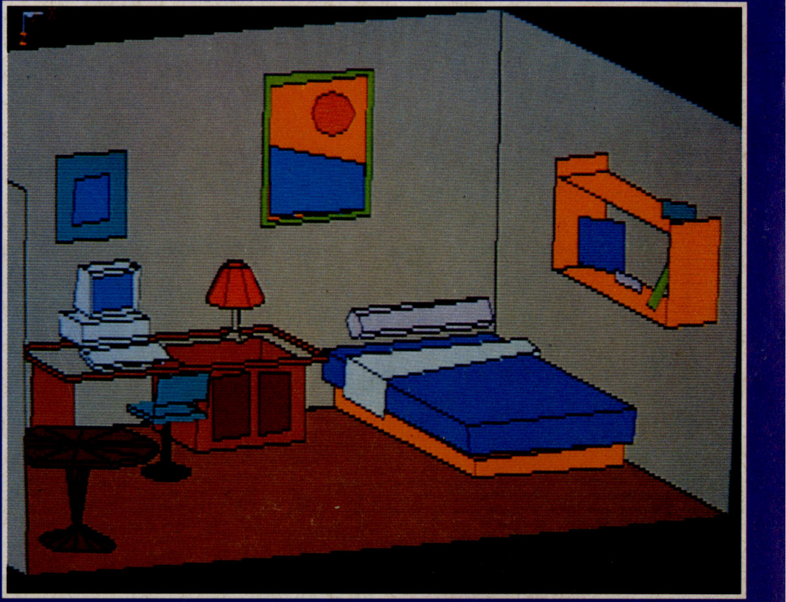
£1.50 ONLY

YOUR AMIGA

February/March 1989
£1.50



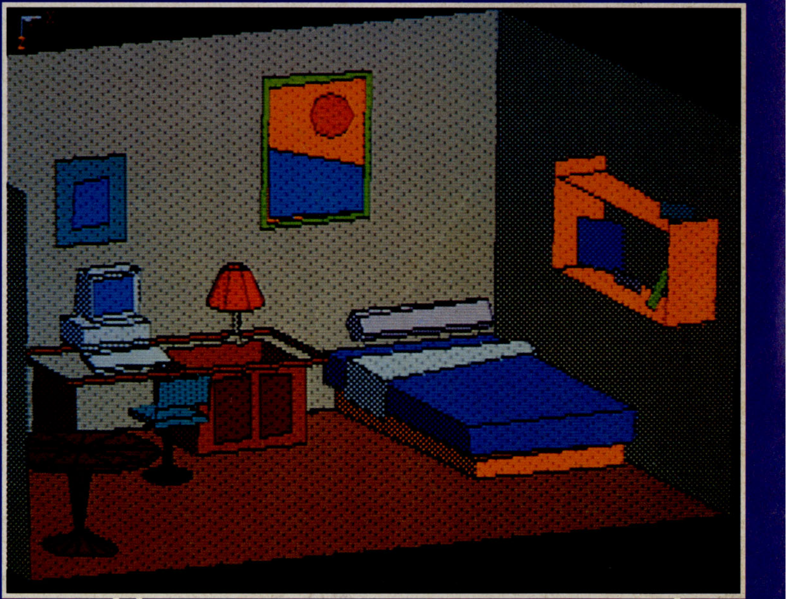


WE REVIEW

DESIGN 3D

FORMS IN FLIGHT

AEGIS 3D



Computer Aided Design

Deluxe Photolab – picture processing

Eliminate Viruses with VIP

REVIEWED

Captain Blood

Pioneer Plague

Sorcery +

Midi Magic

SERIES

CLI for beginners

First words in C

68000 programming

Amiga Music

LAN COMPUTER SYSTEMS

SHOW ROOM OPEN MON TO SAT 10.30am to 5.30pm
ONE OF EUROPES LARGEST AMIGA CENTRES
WE ONLY SELL GENUINE U.K. SPEC AMIGA

LAN AMIGA SOFTWARE PACK

AMIGA 500

Free 20 programmes condensed onto 2 disks for your convenience.

FREE MOUSE MAT *FREE* DUST COVER *FREE* JOYSTICK

£100 OF SOFTWARE WITH YOUR AMIGA

YES YOU CHOOSE THE SOFTWARE YOU WANT WITH YOUR AMIGA (£100 RRP) IF YOU WISH TO ORDER A TV MODULATOR THEN YOU WILL HAVE **£75.00** TO SPEND ON SOFTWARE

£419 INC. VAT

LAN CHRISTMAS PACK

SAVE £180 OFF RRP

A500 computer
Mouse controller
Dust cover
Mouse mat
Joystick
T.V. modulator
10 blank disks
Disk storage box

Ports of call
3 Stooges
Barbarian
Terrorpods
Wizz Ball
ECO
Obliterator

ALL THIS FOR £499.00 inc. VAT

Available only while stocks last.

★ MONITORS ★

1084 S Stereo High Res Col	£199.00+VAT
if purchased with computer	£189.00+VAT
Philips 8833 monitor	£219.00+VAT
if purchased with computer	£209.00+VAT
High resolution monochrome monitors from	£69.00+VAT

★ ACCESSORIES ★

A501 RAM expansion	£129.00 inc. VAT
Mouse mat	£4.95 inc VAT
Amiga dust cover	£4.95 inc VAT
Philips/1084 dust cover	£4.95 inc. VAT
3½" disk drive cleaning kit	£7.95 inc VAT
TV modulator	£24.15 inc VAT
10 D/S 3.5 disks	£14.95 inc VAT
20 D/S 3.5 disks in 80 capacity lockable box	£29.95 inc VAT

★ DRIVES ★

Cumana CAX354 with on/off switch	£99.95 inc VAT
Supra 20mb hard disk	£400.00 inc VAT

★ PRINTERS ★

Citizen 120D	£99.00+VAT
Panasonic KXP 1081	£145.00+VAT
Star LC 10 Printer	£189.00+VAT
Star LC10 Colour printer	£233.00+VAT
Nec P2200 24 Pin Printer	£279.00 + VAT
Epson LX 800	£179.00+VAT
256K Printer Buffer	£160.00+VAT
Printer Cable	£13.00+VAT
Box of Paper 2000 Sheets	£13.00+VAT
Printer Stand	£9.95+VAT
Printer Stand	£26.00+VAT
Printer Ribbons from	£3.00+VAT
New Star LC24-10	£299.00+VAT
Nec Multisink II	£469.00 + VAT

★ SPECIAL OFFER ★

STAR LC10 COLOUR
IN STOCK
STAR LC10 COLOUR PRINTER £233+VAT

OKIMATE 20 COLOUR PRINTER
£139.00 inc. VAT

★ AMIGA B2000 ★

Internal Genlock	Phone
8Mb RAM Board Polulated to 2Mb	Phone
B2000	£849.00
2MB RAM BOARD	£399.00
XT Bridge Board	£399.00
20 MB hard disk	£399.00

A501 RAM EXPANSION NOW IN STOCK

APPLICATIONS SOFTWARE

The Works All For £69.00 inc. VAT

A Wordprocessor + Database and Spreadsheet

The Works is A Three in One Pack

Scribble 2 is the Word Processor
Analyse 2 is the Spreadsheet
Organise is the Database

LANSOFT CLUB

The Club for St and Amiga users everywhere all prices include VAT

★ Special Introductory Prices ★
★ Membership — £20 — (per year) ★
★ Save up to 80% off Games Software ★
★ Up to 30% — Off Application Software ★
Special Club Prices on Cumana Disk Drive £89.00 inc VAT

01 597 8851
1063 HIGH RD,
CHADWELL HEATH,
ROMFORD, ESSEX, RM6 4AU

HOW TO ORDER

Enclose letter with cheque, postal order or credit card number for amount including VAT and delivery charge. Credit card holders may order by telephone. Contact sales desk. Dispatch normally in 24 hours.

Prices current at time of going to press. Subject to availability.

All prices exclusive of VAT + delivery unless otherwise stated

Official Orders from Educational Establishments, Local Authorities and Government departments welcome. Contact Education Department.

FAX: 01-590 6057. Telex: 995548.
Telex: 995548.

TV TUNER

Turn your monitor into a TV. 12 Programmes selection complete with all cables + aerial switchable between TV + computer.

ONLY £55.00 + VAT
if ordered with monitor
£50.00 + VAT

Editor: Stuart Cooke

Deputy Editor: Fin Fahey

Group Editor: Mark Webb

Ad Manager: Paul Kavanagh

Ad Copy Control: Andrew Selwood

Origination: Project 3

Designer: Kim Goodhew

Your Amiga is a bi-monthly magazine appearing on the third Friday of every second month. The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the written consent of the company. ©1989

Distribution SM Distribution, 6 Leigham Court Road, London SW16 2PG. Printed by Redwood Web Offset, Trowbridge, Wiltshire. Origination: Project 3, Whitstable, Kent.

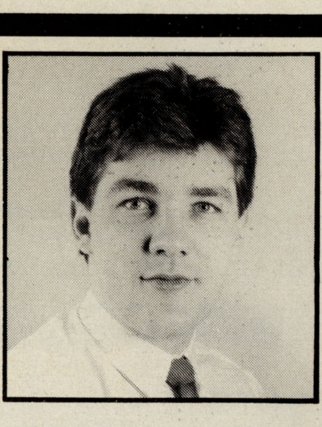
Your Amiga, Argus Specialist Publications Limited, Number 1, Golden Square, London W1R 3AB. Telephone: 01-437 0626 Telex: 8811896. Opinions expressed in reviews and the opinions of the reviewers are not necessarily those of the magazine. While every effort is made to thoroughly check items published for errors we cannot be held responsible for any mistakes that do occur. We regret that we are unable to answer readers letters on an individual basis.

Subscriptions are available from: Infonet Ltd, 5 River Park Estate, Berkhamsted, Herts, HP4 1HL. Rates are as follows: UK £9.00, Europe £11.80, Middle East £11.90, Far East £13.00, Rest of the World £12.10.

Above rates are for surface mail. For airmail rates please contact Infonet at the above address. US subscriptions by Wise Owl Publications, 4314 West 238th Street, Torrance, CA90505 USA.

This month's cover produced using *Design 3D*.

ARGUS
PRESS
GROUP



Welcome

■ **Commodore's Christmas advertising campaign** certainly seems to have worked. The number of boxes that have been moved from the shops, not to mention at the Commodore show, tell their own story. There's an interesting story of a well known distributor selling 10 times the number of Amigas at the Commodore show as he did a certain other computer at the Atari show — this all bodes well for Amiga in the future.

No doubt many of you reading this are new to the world of Amiga and to this magazine — well, you certainly have a treat in store.

Starting to use your Amiga computer can be a difficult and frustrating time, we have heard of many people who have simply given up and resigned their computer to just a games machine. OK, so the games are great, but you're hardly using the Amiga to its full potential.

We at *Your Amiga* think that its a shame to underuse your computer's potential, hence the editorial stance of the magazine.

Of course, you'll find all the best games reviewed, together with all the latest productivity programs but we also offer advice on programming and using your computer. Our CLI, C and 68000 series have proved exceptionally popular and we will of course continue to bring you more of the same.

Don't forget that our Guru is around to answer your queries through the pages of the magazine. If you have a problem write to him, he can't reply on an individual basis but he will cover as much as he can each issue.

Have fun with your Amigas.

Stuart Cooke

Contents

FEATURES

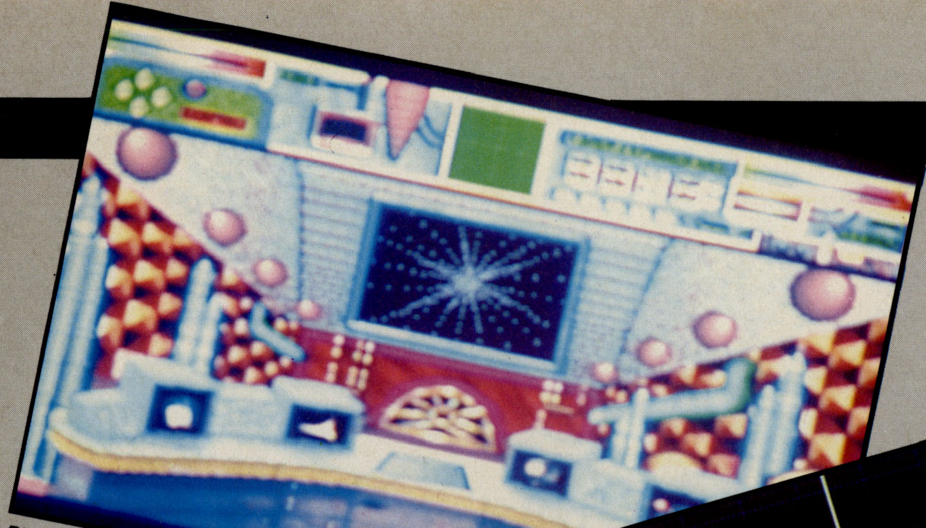
YOUR AMIGA

ENTERTAINMENT

PROGRAMMING

REGULARS

CONTENTS



Pioneer Plague

Statistics

Points -- 1952
Polygons -- 238
1 Vertex 0
2 Vertices 0
3 Vertices 209
4 Vertices 29
>4 Vertices 0
Default Color 0

Aegis 3D

Which is the only Great Lake that is entirely within the USA?

Kevin Geog.

Trivial Pursuit

MIDI Magic © 1987 René Vega

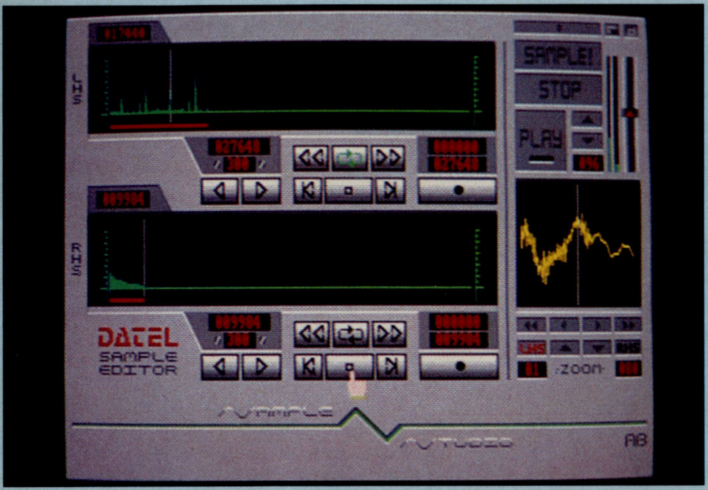
Thu Sep 22 09:40

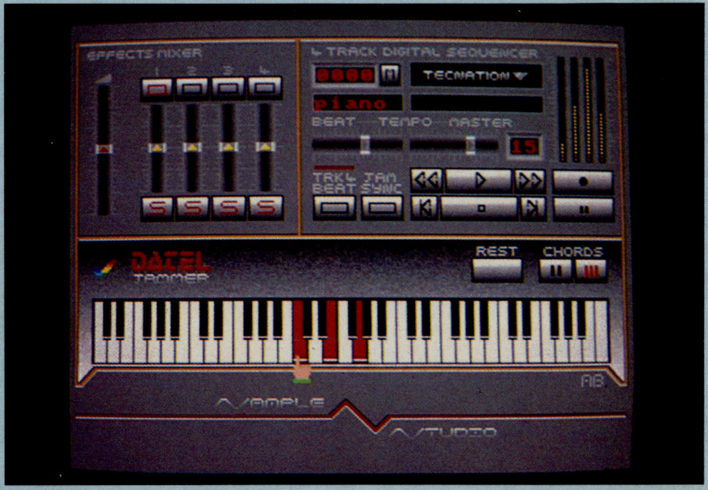
Measure Beat Pulse
00001 01 001 120

MIDI Magic

Sorcery Plus

DATEL ELECTRONICS





□ AMIGA PRO SAMPLER STUDIO + DATEL JAMMER

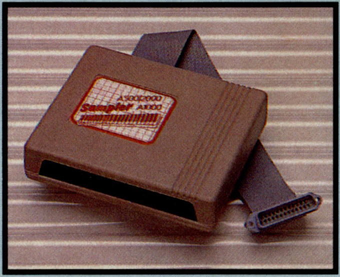
- A top quality sound sampling system at a realistic price.
- 100% machine code software for realtime functions.
- HiRes sample editing.
- Realtime frequency display.
- Realtime level meters.
- Files saved in IFF format.
- Adjustable manual/automatic record trig level.
- Variable sample rate & playback speed.
- Seperate scroll line waveform windows plus zoom function with Edit windows for fine accurate editing.
- 3D shot of sound waveform. Wave editor to design your own waveforms or adjust existing ones.
- Microphone & line input 1/4" Jack & Din connections.
- Software files can be used within other music utilities.

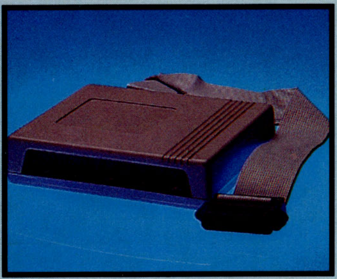
To complement the Sample Studio the Datel Jammer gives you a 5 octave keyboard to play & record your sampled sounds.

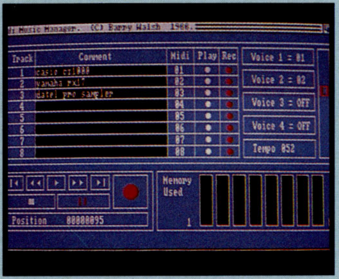
FEATURES:-

- 4 track sequencer up to 9999 events.
- Tempo & Beat controls.
- Mixer Controls on Instruments.
- Load & Save sequence.
- Works on standard IFF file sounds.

ONLY £69.99 PLEASE STATE A500/1000/2000







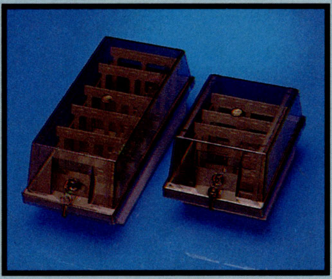
Midi Music Manager. (C) Barry Walsh 1988.

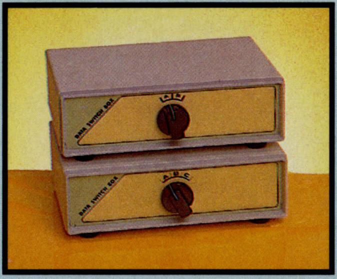
Track	Comment	Midi	Play	Rec
1	casio cz1000	01	●	●
2	yamaha rx17	02	●	●
3	datel pro sampler	03	●	●
4		04	●	●
5		05	●	●
6		06	●	●
7		07	●	●
8		08	●	●

Voice 1 = 01
Voice 2 = 02
Voice 3 = OFF
Voice 4 = OFF
Tempo 052

Position 00000095

Memory Used 1





□ MIDIMASTER

- Full Midi Interface for A500/1000/2000 (please state model).
- Compatible with most leading Midi packages (including D/Music).
- Midi In - Midi Out x3 - Midi Thru.
- Fully Opto isolated.
- No need to pay more - Full Midi standard.

ONLY £34.99

□ MIDI CABLES

- Top quality.
- 3 metre length.

ONLY £6.99 PAIR
UNBEATABLE VALUE

□ MIDI MUSIC MANAGER

A TRULY PROFESSIONAL MIDI PACKAGE AT A REALISTIC PRICE

- Play sampled sounds on Amiga from any Midi track.
- Full dubbing - listen to one track while recording another.
- Works with many Midi interfaces including Datel Midi Master (see Ad).
- 8 realtime Midi tracks for record/playback.
- Adjustable track length - limited only by available memory.
- Works with standard IFF files.

ONLY £39.99

□ DISC/STORAGE BOX OFFERS

- DD40 holds 40 3.5" discs. Lockable.

ONLY £6.99

- DD80 holds 80 3.5" discs. Lockable.

ONLY £8.99

□ DISCS

- 3.5" DS/DD.
- Top quality, bulk packed.

ONLY £22.99
FOR 25 DISCS

□ DATA/SWITCH BOXES

- A/B type connect two printers to one computer or vicé-versa.
- Centronics connections or RS232 Serial connections (25 pin). Please state.

ONLY £24.99

- ABC type connect three printers to one computer or vice-versa.
- Centronics or RS232 connections.

ONLY £34.99

□ PRINTER LEADS

- 25 pin 'D' to 36 way Centronics parallel lead. 1.2m length.
- A500 or 1000, please state.

ONLY £8.99

□ ROBOTARM

FULL FUNCTION - 5 AXIS MOVEMENT

- Explore the fascinating science of Robotics with this full feature Robot Arm.
- Human like dexterity - with 5 Axis of movement it is so versatile. It can manipulate small objects with amazing ability.
- Easily controlled using 2 joysticks (any 9 pin type) or connect to your Amiga with our Interface + Software to give Computer/Robotic control (see Interface offer).
- Comes with Accessories including 'Finger' Jaws, Magnetic Attachment, Shovel Scoop, 4 Stabilizing Suction Base Legs, etc.
- Uses 4 HP2 batteries (not supplied) to power motor movement so uses no computer power.
- Self contained, ready to use (except batteries, joysticks).

ONLY £49.99

□ INTERFACE OFFER

- Unique Software/Hardware package to allow you to interface your Amiga with the Robotarm.
- Train mode allows you to store & then repeat movement sequences.
- Very easy to use.
- This Interface is not needed to use the Robotarm but interfacing with your Amiga has great possibilities.

ONLY £24.99
COMPLETE WITH CABLES.

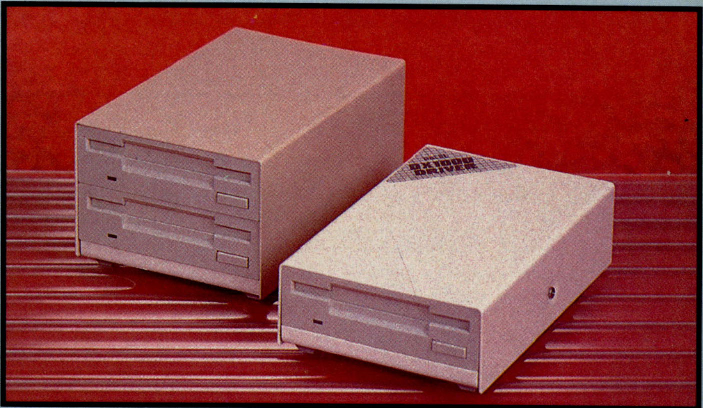
DATEL ELECTRONICS

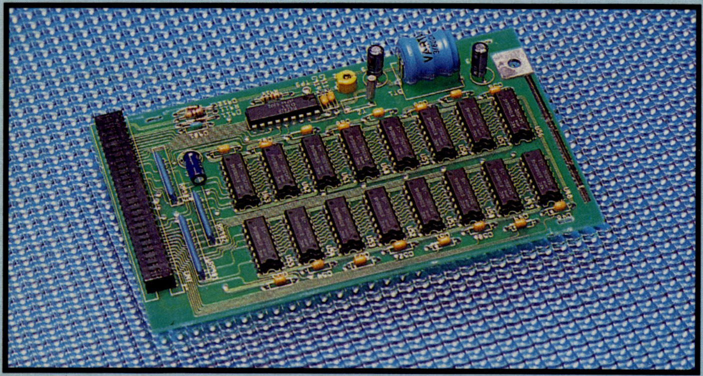
□ EXTERNAL 3.5" DISC DRIVE

- Slimline extra low profile unit - only 6" long!
- Top quality NEC drive mechanism.
- Throughport allows daisy-chaining other drives.
- A superbly styled case finished in Amiga colours.
- Fully compatible.
- 1 meg unformatted capacity.
- Good length cable for positioning on your desk etc.
- Value for money - before you buy a drive please compare the features - this drive has an NEC drive unit & is housed in a superb housing - many units available are built to a price & not a standard. Don't spend a few pounds less & end up with 'rubbish' - & remember you are buying from the manufacturer.
- Complete - no more to buy.
- Single or twin drive models available.

ONLY £89.99 SINGLE DRIVE £149.99 TWIN DRIVE

ADD £5 FOR COURIER DELIVERY IF REQUIRED





□ 512K RAM EXTENSION CARD

- Available with/without calender/clock option.
- Simply plugs internally into A500 slot.
- Switch in/out with switch supplied.
- Fitted in minutes - no soldering etc.
- Accepts 41256 DRams (zero K fitted).
- With calendar/clock onboard time/date automatically booted.
- Battery backed to retain time/date.

ONLY £19.99 FOR STANDARD CARD TO ACCEPT 512K

ONLY £34.99 FOR VERSION WITH CLOCK/CALENDAR

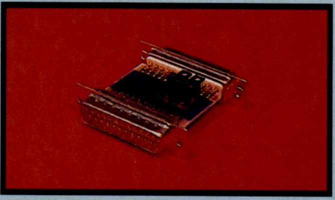
PHONE FOR LATEST FULLY POPULATED BOARD / RAM PRICES.

□ MARAUDER II

QUITE SIMPLY THE BEST DISC COPIER AVAILABLE FOR THE AMIGA (ALL MODELS)

- Superfast disc copier will copy almost any commercial disc.
- Friendly user Interface - Mouse driven throughout.
- Completly compatible with Amiga multitasking system.
- Even decrypts many encoded programs including D. Print/Video/Paint/Music/II etc.
- Supports up to 4 drives simultaneously for multple copies.
- Special 'Strategy Files' cope with even the most advanced protection schemes.
- Now shipping the latest version.
- Fast option - typically around 80 seconds.
- USA's top selling copier.

ONLY £29.99 POST FREE



□ EXTERNAL DRIVE SWITCH

- Switch in/out of external drives.
- Save on memory allocated for drives not currently in use.
- DF1 & DF2 controlled.

ONLY £9.99

□ REPLACEMENT MOUSE

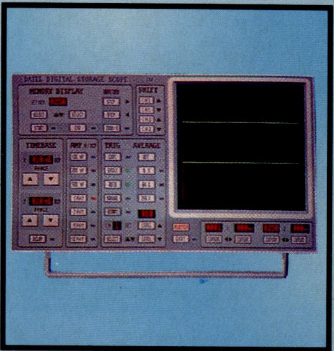
- Fully Amiga compatible.
- Rubber coated ball.
- Optical type.

ONLY £24.99

□ SPLITTER LEAD

- Allows joystick & mouse to be connected to same port.

ONLY £4.99



□ DATA ACQUISITION UNIT

- Turn your Amiga into a sophisticated measuring instrument capable of measuring a wide range of data inputs.
- Sample & display events from microseconds to hours- with amplitudes from milivolts to 50 volts.
- A Hardware/Software package with very high spec. including:-

DIGITAL SCOPE DISPLAY - 2 channel inputs.Manual or continuos display. Timebase 500ms/div to 20us/div- accurate to 5%.

- 6 bit flash conversion gives 2 million samples/sec.

PLOTTER DISPLAY

- Timebase range 1 sec to 10hrs per plot.

All features found on units costing thousands of pounds.

ONLY £89.99

PLEASE STATE A500/1000/2000

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

DATEL ELECTRONICS

**DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE
GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.**

SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324

AMIGA UPDATE

Wooden Digger's Soap

No, this isn't a piece about *Neighbours* but just three of the latest applications of the ubiquitous Amiga. Archeology, soap production and the New Forest tourist centres are all using the Amiga to increase their efficiency.

This season is truly a winter of disk contents for sons of York as archeologists working on the Helserton Project based in North Yorkshire start to use the Amiga for site surveys. The Project covers several acres of moorland which was originally the site of both a Roman and Anglo-Saxon settlement.

Principal archeologist and computer programmer, Dominic Powlesland, developed a database on the Commodore PC60 which represents the site as a 2D map. Using a mouse-driven interface, the user can define an area of the map and zoom in for closer study. Every find within the area is logged for position size and depth, allowing the user to analyse the site in detail or from a wider historical and geological perspective.

Dominic is currently transferring all of this onto a 2000 system which allows the added facilities of including video stills and sequences to further improve this important and revolutionary process.

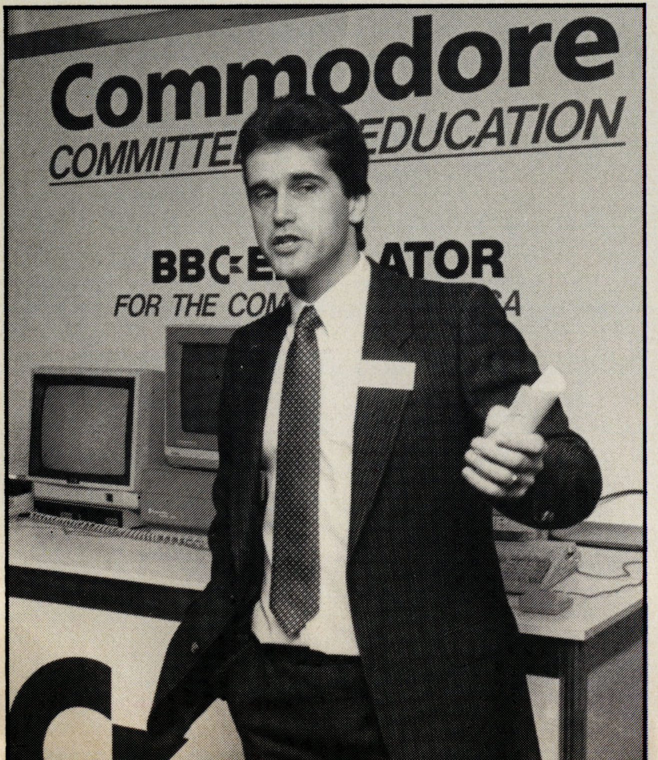
Amiga Collects Acorns

If Commodore is going to make a serious bid for the British Education market, it has to break Acorn Computer's virtual stranglehold, so the announcement of the *BBC Emulator* at the Novotel show forms a landmark in the Amiga assault.

Programmed by Ariadne and marketed by James Associates, the emulator claims to run BBC Basic programs seven times faster than normal and 6502 code just slightly slower than the normal host machine. This means that Amigas can now be networked alongside BBC B but has the added educational advantage that MS-DOS, Amiga and BBC programs can be used — if necessary (and if memory allows) all three can be running concurrently!

Educational establishments can buy the *BBC Emulator* for £39.99 but the street price is £49.99. Not a bad price for a Beeb.

Commodore's Peter Talbot serenades the Education sector



Digita's Jerry Rihill — seems like a nice bow!

Accounting for Commodore

During the Christmas Commodore Show at Hammersmith's Novotel, Digita International announced that Commodore had adopted its *Home Accounts* program for inclusion in a promotional offer to accompany the Amiga 500.

The company's director, Jerry Rihill, also launched a year planner and a rather novel program to convert the Amiga into a typewriter. The planner, *Day-By-Day*, was designed in conjunction with a lecturer in Cognitive Psychology from Exeter University.

E-Type is the typewriter program which may seem a quirky idea, to say the least. Although it can be used for dashing off simple letters and memos, the main advantage of this program is to enable forms to be filled in via the Amiga and its printer.

For more details contact: Digita International, Kelsey House, Budleigh Salterton, Devon EX9 6HJ. Tel: (0395) 45059.

Word's Out

The *Protext* wordprocessor is already available for the PC and Atari ST machines but the new Amiga program is more than another trans-ported version. Tailored to use the Amiga's WIMP environment, Arnor's product features automatic re-formatting, page break lines shown during editing, macro record mode, background printing, line drawing and footnotes. The system is supported by a fast 7000 word spelling checker which will work as the user types in a document.

Probably the best feature of *Protext* is that it is 100% British.

Protext is available from Arnor, Protext House, Wainman Road, Peterborough PE2 0BU. Tel: (0733) 239011.

Chrysalis Hatches

First new software house of 1989 is Chrysalis, son of Teque (*Terramex, Pac-Mania, Munsters*). The Rotherham-based company has been quietly working on its first release, *Prison*, which is now ready to burst onto the 16-bit scene.

Prison, programmed by Michael Hart and Jason Wilson, involves a disconsolate inmate of a penal colony deep in space. News of a dismantled escape craft hidden on the planet gives him new hope but information is hard to come by. To effect an escape, he must search the planet for those who hold the secrets to the locations of the ship's parts.

The search takes the graphic adventurer through nightclubs, rail trips and numerous battles with mutant guards who will challenge anyone's right to escape.

Michael and Jason (Spock) help Chrysalis to break free from Prison

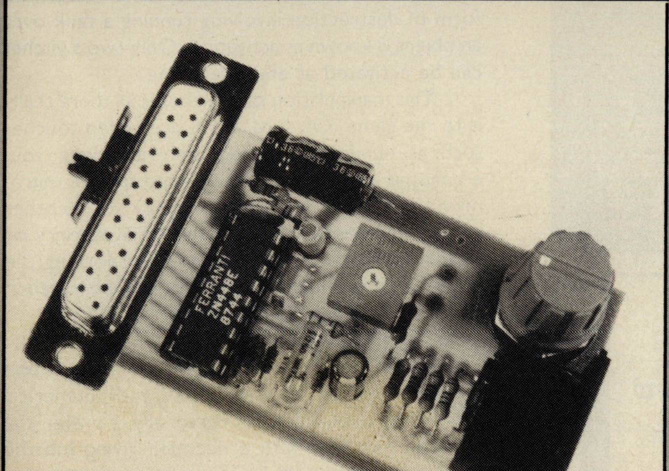
Hard Backup

Hard disks are marvellous additions to any Amiga system but, like any physical storage medium, wear and tear eventualy takes its toll. With most users storing valuable utility programs, a hard disk fault can mean hours of fiddling about re-loading the original programs. The Disc Company has produced a piece of software which can cut this time down to 20 minutes by providing a dump program which stores the contents of any hard disk onto a series of floppy disks.

SuperBack takes approximately 20 minutes to transfer 20 Mb of information which can then be reloaded when needed. The utility will also transfer RAM disk contents to disk and costs £59.95 from Amiga Centre Scotland, GEM, HB Marketing and Microdealer.

Port Sunlight harbours Lever Brothers' new £12 million plant which has the ability to produce over one million bars of soap per day. The process is controlled by a network of DEC minicomputers connected to logic controllers attached to flow controllers, thermometers and the like. The network also interfaces to an array of 20 Amigas which use Nucleus Software's *Dexterity*package to display the production processes in real time. The Amiga system is proving to be a cheaper alternative to large panels of dials and flashing lights.

Down in the New Forest something stirs as tourists find an information station based on two Amiga 2000 machines. The computers, installed by Aspen Interactive, can be used to display the sights and sounds of the Forest and can be used to plan walks as well as showing animated sequences of what each particular activity involves.



ZEN's low cost A/D3 Audio Digitiser

Budget Sampling

ZEN Computer Services' *Audio Digitiser* for the Amiga offers high sampling times for £16.

The digitiser has a fast A/D converter which can actually be made to operate at speeds in excess of the Amiga's maximum sampling rate. Compatible with packages such as *AudioMaster* and *PerfectSound*, the sampler plugs directly into the A1000 but a gender reverser is required for connection to A2000 and A500 models.

The *A/D3 Audio Digitiser* costs £16.00 if supplied in kit form and £25.00 assembled and tested.

The package is produced by ZEN Computer Services, 2 Silver Birch Grove, Swinton, Manchester M27 1FS.

Perfect Sentinel

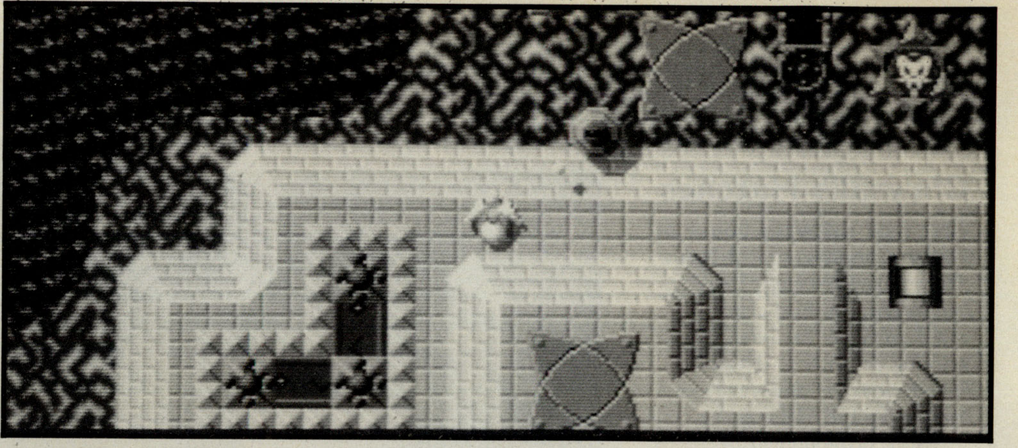
Sentinel Software is changing its name to WordPerfect UK. This underlines the close relationship which Sentinel has developed with WordPerfect Corporation over the past three years in which product sales have soared from around ten units per month to the present level of about 70 units per day.

WordPerfect is a powerful wordprocessor available across a wide variety of computers and much of *Your Amiga* is originally submitted on this product.

For more information contact WordPerfect UK, Hellington House, New Zealand Avenue, Walton on Thames, Surrey KT12 1PY. Tel: (0932) 231164.

FUSION

Gordon Hamlett can't normally mend a fuse. What hope has he of saving the world from Fusion?

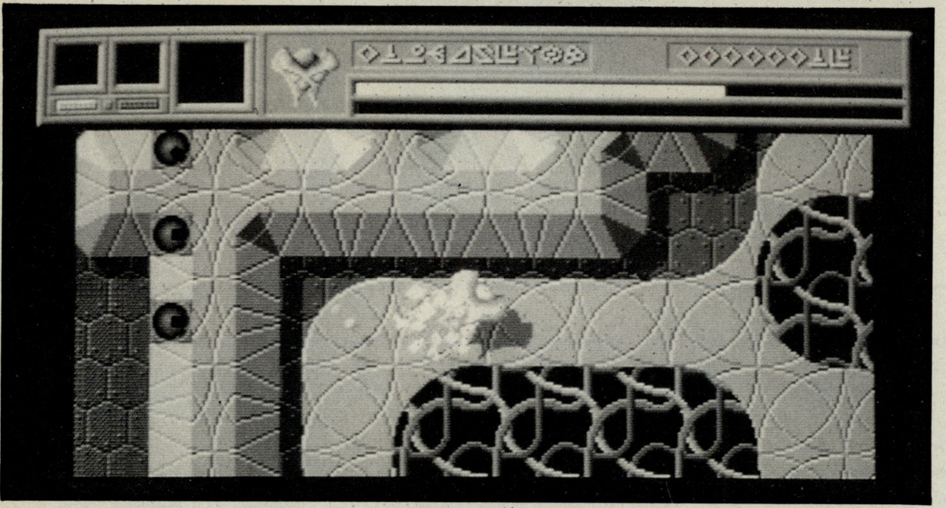


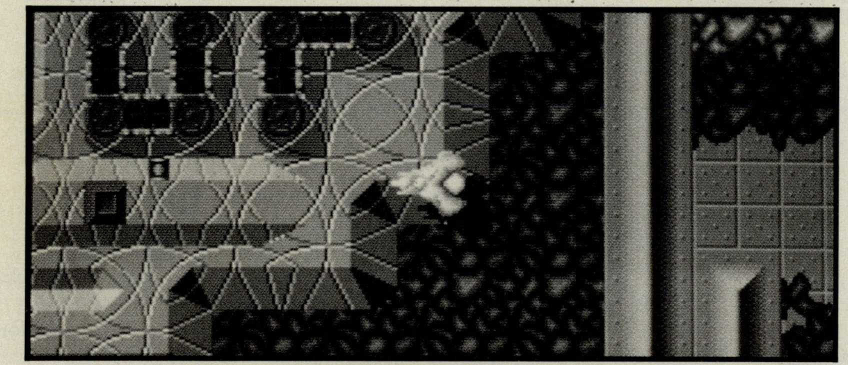
■ **The first thing that struck me about** *Fusion*, the new shoot-em-up from Electronic Arts, was its manual. Not because it was superbly written, or one hundred and pages long, or full of the most erotic artwork yet seen in a computer manual. It struck me, however, Alas, it was none of these reasons, but it is the first I've seen that refuses to take itself seriously.

Not for the authors some angst-written prose about how you are the sole surviving person in the universe capable of single handedly quashing the rebellion where you will find, should you succeed, a voluptuous being wearing a badge with the legend princess written on it. Instead, you, the player, have to 'reach a stage where we agree that you have successfully completed all puzzles and overcome all the obstacles that we have planned.'

Apart from that, the only scenario comes in a section marked 'Not the Story' in which Captain Bloowd single handedly flies his Trang class fighter home from the northeast quadrant of the Spiral when a message is received stating that the Galaxy is under threat. At least there is no mention of the Princess.

All you have to do is collect all the pieces of The Bomb, preferably before your energy gauge runs out and with the added option of a really tough version for smart alecs. You control two ships. A fly-nearly-everywhere mother ship and a much slower Assault Crawler for reaching the parts other craft cannot reach. The only problem here is in finding somewhere suitable for the mother ship to land so that you can launch the crawler. Once the ship is about half an inch away from the crawler on screen, (about 0.3 megaparsecs in instruction-speak) it fades away so you will have to remember where you parked. A sort of inter-galactic multi-storey car park without the fun.





All the really interesting parts of the game are protected by colour-coded gates. To disable them, it is necessary to find the appropriate switch and run over with your crawler. For some reason this form of destruction involving running a tank over an object is known as activation. Only two switches can be activated at any given time.

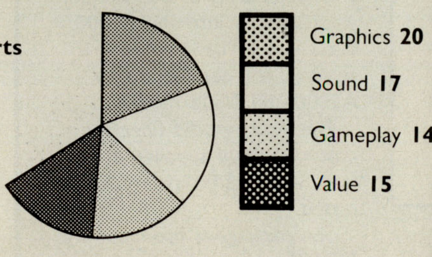
This manipulation of switches is all there really is to the game. Oh sure, there are added touches such as hundreds of alien craft attacking you, ergonomic eruptors, rotating plasmo spheres, nitro-mice and the like. Teleports can whisk you to other levels, extra features such as better weaponry or restored energy can be obtained by shooting the requisite object and your game can be saved if desired for further encounters.

To show how really hip everything is, you don't actually get a score as such but lots of little symbols which can be decoded to show your numerical supremacy over a friend. Personally, I prefer the score in numbers with a decoder giving me the Martian figure should I wish to convert.

The game is little more than an omnidirectional scrolling shoot-em-up with a few puzzles thrown in. It looks good and there are some interesting digitised sound effects but the game itself just doesn't work and lacks that certain something that makes you want to keep going back and playing again. Perhaps it does need the incentive of a helpless princess after all! *YA*

FUSION

Title: **Fusion**
Supplier: **Electronic Arts**
11-49 Station Rd
Langley,
Berks SL3 8YN
Tel: 0753 49442
Price: **£24.95**



Sorcery Plus

Are 8 bit conversions what we're really after in this day and age? Kevin Crosby finds out

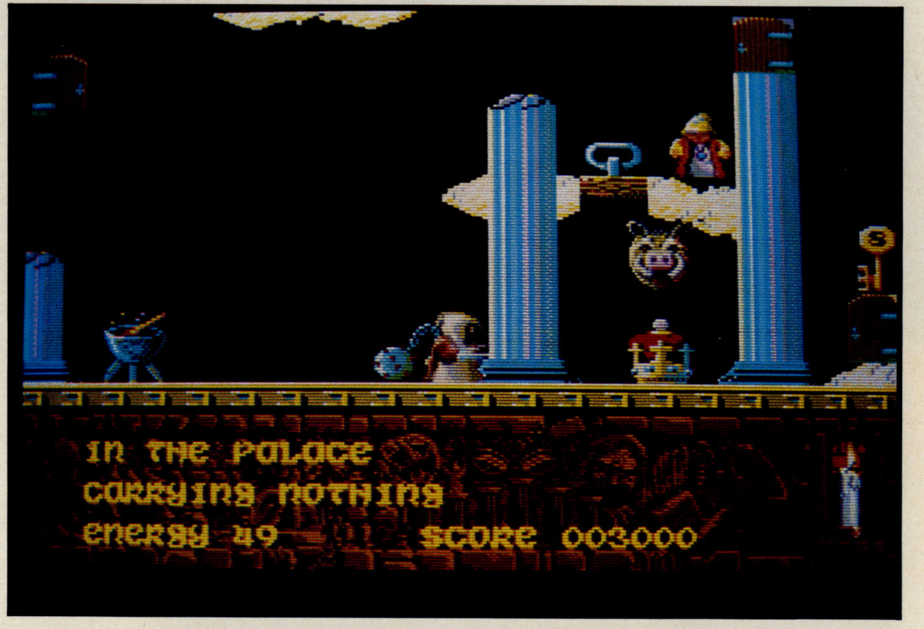
■ **Anyone who bought an Amiga as an** upgrade from a lesser machine may well have come across this game from Virgin already. Well the little sorcerer and his quest to free all his chums have been given a new lease of life on the 16 bit circuit.

For those who haven't come across this game the basic idea is this. Evil Necromancer has gatecrashed a kingdom and locked up all the sorcerers bar one; yourself, who now has to zap round the kingdom freeing your counterparts so that they may return to the sanctuary. Once they are free you can then join them before embarking on the final quest — the destruction of the evil Necromancer.

So what does all this actually mean in practice? Well it's basically a question of flying round 47 locations searching for the various "keys" that unlock the trap doors that hold captive your associates (plus another 28 in the second stage with the Necromancer). To get from one location to another you must use the creek doors dotted about on each screen. I have to admit, this aspect of the game did throw me a little at first as one gets used to just leaving from one screen into another. Makes mapping the kingdom a lot trickier as well. It'll come as no surprise that just about every screen as more than it's fair share of nasties to drain your energy supplies (which starts at 99 units), however you can use shooting stars and spells to waste the bad guys with, along with axes and swords that work on air and land based villains respectively. Strategically placed caldrons or a passing visit to the sanctuary allow you to replenish your energy level.

Quite a wide variety of objects are at your disposal. Some are highly useful such as the coat of arms for instance which gains you access to parts of the palace. However be warned, some objects do nothing but kill you off.

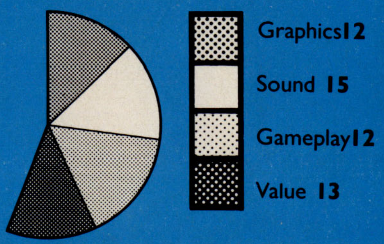
I should point out at this stage that improvements have been made, it isn't just a straight port across from a Amstrad. The program has been re-coded for the 16 bit market by the Gang of Five. Although the gameplay is pretty much the same as in the Martin Wheeler original. The sound effects have been replaced by digitised equivalents including door creaks, thunder cracks and a rather cute "ahhh" as you replenish your strength. The graphics have been "Rezzed" up accordingly. So credit where credit's due they have tried their best with this game although I would have much rather seen the efforts put into a new game which lends itself to these sort of computers more usefully. *YA*



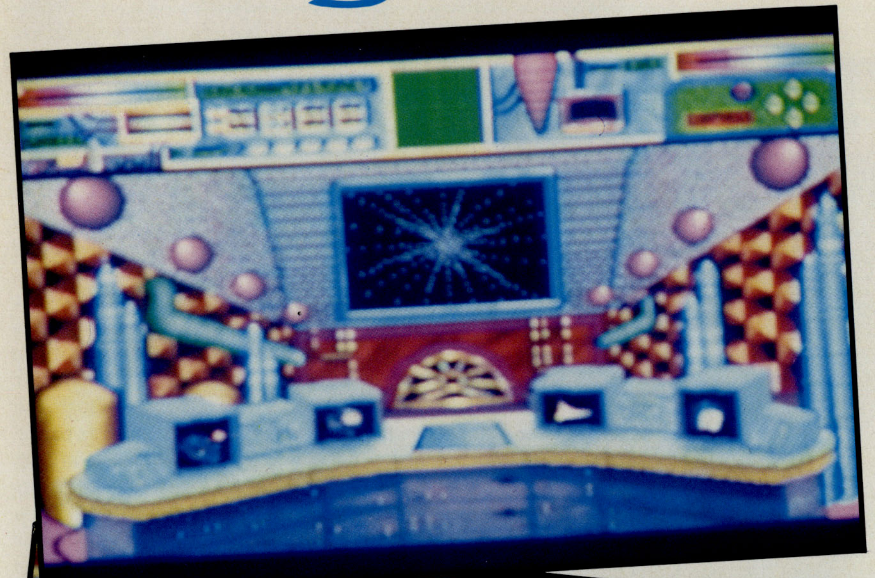
in the palace
carrying nothing
energy 49 score 003000

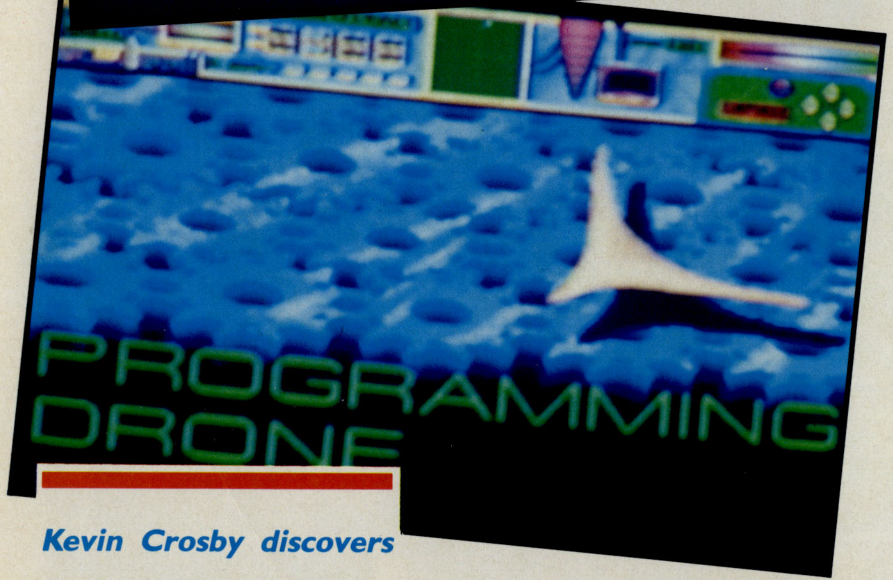
SORCERY PLUS

Title: **Sorcery Plus**
Supplier: **Virgin Games Ltd**
Address: 2/4 Vernon Yard
Portobello Road,
London W11 2DX
Price: **£24.95**



Pioneer Plague

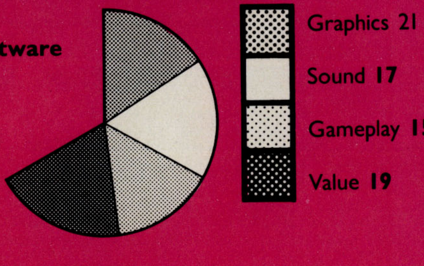




Kevin Crosby discovers that the universe has more than 16 colours with the latest from Mandarin Software

PIONEER PLAGUE

Title: **Pioneer Plague**
Supplier: **Mandarin Software**
Europa House
Adlington Park
Adlington
Macclesfield
SK10 4NP
Tel: **(0625) 878888**
Price: **£24.95**



I get a warm feeling inside when, at long last a software house wakes up to the fact that the Amiga is not restricted to 16 colours, it can, in fact produce several thousand more than that in the now infamous HAM mode which has a reputation for being somewhat problematical for programmers, hence the tendency to simply port across from lesser machines (you know to which machines I refer).

So what is *Pioneer Plague* all about? Well put simply it's a cross between *Star Raiders* and *Time Pilot*; only much prettier than either. The object of the game is to prevent the spread of Mark IV Pioneer Probes which were originally designed to colonise and transform uninhabitable planets. However all that went horribly wrong some centuries ago. A mutation of these probes now lands on any old planet and transforms it, which can be a might inconvenient for anyone living there.

So it's up to you to stop the spread by wiping out the probe silos before any more are launched. Yes folks it's one of those race against time shoot-em-ups.

The game is split up into three areas, the strategy/control area, the planet surface and the sub-euclidian plane.

Your control panel consists of four on-screen monitors which are accessed by clicking the mouse on the one you want. Leftmost display is the "Let's descend to the planet surface and waste a few probes" icon. If any probes are down there you are treated to a very pretty picture of the world you're about to save (hopefully) before you swing into action (more about that later). The next screen is your galactic navigation control which allows you to jump through sub-euclidean worm-holes to other planets in need of assistance. However this is no cosy little jaunt into hyperspace — you have to work quite hard to get from A to B.

Not only must you blast a gravity well to get out of the Wormhole (the longer you leave it the more silos appear on the planet you're heading for) but you also have to fend off space mines which bounce you back to the start of the hole giving those probes more chance to multiply. Sub-euclidean travel is all very well until you sustain some damage. Then you have to plot the course yourself which involves a lot of tedious mucking about with three bits of photocopied paper and four sheets of acetate which don't, to my mind, enhance the game at all although they should mess up the plans of a few software pirates.

Once over the planet you can now leave the safety of the multi-coloured airship and destroy all the silos before they launch probes bound for nearby worlds. However there are plenty of airborne adversaries more than happy to slow you down in your endeavours. Mind you, help is at hand with the programmable drones you have on board. These can be use to deflect enemy fire, take out gun emplacements and collect energy. Drones can be programmed using the third on-screen monitor. Particularly useful patterns can then be saved to disk using the fourth and final monitor.

Not a bad game, I suppose, although I can't honestly say it's one I'd come back to a lot. One problem many of the Christmas purchasers may encounter is that of definition on a television screen as many of you have bought A500s with the TV modulators. You guys will find yourselves with a few problems picking out red (active) silos on your scanner. Still, hats off to Mandarin for acquiring the rights to this game from Terrific Software. Let's hope there's more where this came from.

AmigaBASIC Part 3

Allen Webb explains how to get to grips with Amiga graphics

■ **Graphics are an important feature** of any micro and are of even greater significance in the Amiga. When you bear in mind that the Amiga has probably got the most powerful graphics capability of the home micros available,you will appreciate the necessity of utilising these capabilities to the maximum. AmigaBASIC offers a range of commands for manipulating graphics and is quite adequate for most users. Forthe more advanced users, the graphics libraries can be accessed from Basic to give even more power. In this part I will describe the use of the basic graphics commands.

If you want an excellent collection of Basic programs, I suggest you read *Amiga Tricks and Tips*, published by Abacus (ISBN0-916439-88-7). This book is full of advanced ideas such assetting up HAM and Halfbrite modes from Basic.

I have already given a tutorial on graphics in my "beginners" series and I will not go into such detail here. In essence, the Amiga uses abit-mapped approach such that a block of memory is directly mapped to the screen. This allows you to display an infinite combination of graphics and text albeit at the expense of memory and processing time. In order to minimise the strain on the 68000CPU, the Amiga has an independent blitter (for rapid data movement) and a graphics co-processor called Copper. The Copper allows you to mix graphics modes in different screen zones. The hardware also has some drawing facilities implemented.

Since you have an AmigaBASIC manual, I don't intend to describe the syntax of each command. Instead I will try to cover the details that the manual conveniently overlooks. The listing gives some examples of the use of the commands.

The first step when dealing with graphics is to set up the graphics screen. This specifies the resolution of screen to be used. AmigaBASIC allows the direct creation of four types of screen. Two levels of horizontal resolution are possible:

■ Lower re solution. This mode uses 320 pixels across the screen.

■ High resolution. This offers a maximum of 640 pixels across the screen.

These modes can be used in interlaced (400 pixels high) or non-interlaced (200 pixels high). Interlaced mode gives greater vertical resolution but due to its mode of operation tends to give a flickery display. High resolution gives greater detail but can only use a maximum of 16 colours. Low resolution offers 32 colours. The screen is used as the base upon which windows are created. Not unsurprisingly, you open a screen using the **SCREEN** command. This has the syntax:

SCREEN Number, width, height, depth, mode

I'll leave you to read page 8-132 of the manual to sort out the meaning of the parameters. Some important points should be made. The intuition system requires any screen to beset to the full width of 320 or 640 pixels. Using a lower width may confuse the display. You can use a screen shorter than the full vertical resolution but there is apparently a problem. Intuition requires that the screen sit at the foot of the display. It is reported that Basic puts the screen at the top causing garbage at the bottom of the screen (I have not found this however). You can , however, drag the screen down using the library routine MoveScreen. I'll show you how later.

AmigaBASIC uses the Workbench screen as screen 1. Since Intuition opens this screen automatically, you don't need to worry about it. If you are content to use a high resolution screen with four colours, it is worthwhile using it instead of opening your own — it saves memory! Screens are allocated a depth arranger gadget allowing you to move screens around if required. When a screen is opened, a data structure is set up containing the important information relating to the screen. More on this later.

You cannot actually write to a screen and it is necessary to open windows to do this. This is achieved by using the WINDOW command (see page 8-158 of the manual). Within limitations, you can create windows of almost any size. Subject to the type specification, you may have any combination of gadgets. AmigaBASIC does not, however, allow you to set up a borderless window. You must also bear in mind that the screen border takes up space. If you are using a 640 by 200 screen, the largest window possible, for example, is 631 by 186.

In a manner similar to the screen, each window has a data structure. This structure contains valuable data on the memory locations of the screen and the window bit map. You can locate the address of the data structure by using **WINDOW(7)**. The relevant addresses are held at fixed offsets from this address. Take a look at the program called Bit Planes on the Extras disk for an idea as to how the data structure is set out. While the pointers in the data structure are of limited value to most Basic users, they are invaluable to those of you who want to use library functions.

Once you've set up your window you can start drawing. In keeping with most machines the Amiga uses foreground and background colours (ink and paper). You choose the colours used by the **COLOR** command. The colours used are selected by specifying the appropriate color register and the colour registers are set up in turn by the **PALETTE** command. In the default system **PRINT** changes both the foreground colours onto the screen) and it has an annoying habit of messing up the information on the screen. There are three other possible modes but AmigaBASIC doesn't allow you to select them. Fear not however, I will show you later how to use a library routine to do this for you.

The basic methods of drawing are provided by the **LINE**, **CIRCLE**, **PAINT** and **AREAFILL** commands. The first two are quite straightforward. PAINT flood fills an area with a specified colour. You can specify which colour is to be used as the border to the filled area. There are apparently two possible bugs with this command:

■ First any attempt to use **PAINT** on a window with an identifying number greater than 15 will lead to a crash. The default screen has a number of 31 so you should ensure that you open a window with a value lower than 16 before using **PAINT.**

■ If you specify a set of **PAINT** coordinates outside the boundary of a window you may cause a crash.

AREAFILL fills a polygon of up to twenty coordinate points and allows you to quickly fill complex shapes.

By default the fills and lines are solid comprising of the foreground colour only. By using the**PATTERN** command you can define your own fills. To do this you will need to manipulate binary. I use a CASIO calculator which operates in binary, hexadecimal, octal and decimal modes — it's much quicker than using pencil and paper! **PATTERN** has the syntax:

PATTERN LINEPATTERN%,FILLPATTERN%

LINEPATTERN% is a 16 bit representation of the line's pattern. The value, in binary, uses a 1 to set a pixel to the foreground colour and a 0 to set the pixel to the background. A solid line is therefore represented by1111111111111111 in binary, FFFF in hexadecimal or -1 in decimal. A dotted line with alternating dots and spaces would be represented by 1010101010101010 in decimal or AAAA in hexadecimal. To set the line pattern to this type you would therefore use:

PATTERN&hAAAA

FILLPATTERN% is built up as an array of short integers. This array defines a pattern sixteen bits wide. You can use any pattern height you choose provided that the height is a power of 2 (2,4,8,16,32 etc). The length of the array is then equal to the height of the pattern. Imagine that you want a pattern of squares each comprising of 4 by 2 dots. Since the pattern repeats every 4 lines, the array need only have 4 elements. In binary this has the form:

1111000011110000 = &hf0f0 =FILLPATTERN%(0)
1111000011110000 = &hf0f0 =FILLPATTERN%(1)
0000111100001111 = &h0f0f =FILLPATTERN%(2)
0000111100001111 = &h0f0f =FILLPATTERN%(3)

All you need to do is set up the FILLPATTERN% array and then use the name of the array in the **PATTERN** command. Look at the listing to see how it works.

Using **PATTERN** in multicolour mode is rather more complex. You must first set up the array to represent the pattern on each bitplane. The number of bitplanes is the same as the depth value used in the **WINDOW** command. To size the pattern array you must choose the next power of 2 greater than the product of the height of the pattern and the number of bitplanes. The example in the listing uses a 16 line pattern with 3 bitplanes. The product of these is 48 and the nearest power of 2 above this is 64. Hence the array is 64 elements long. Since this is larger than you really need you must artificially tell the system that the pattern is 16 lines long. (the previous example could find this out from the size of the array) To do this you must place the size of the pattern as the negative exponent of 2 in the rasterport. Since 16 is 2 4 we need to poke-4 into the 29th byte after the rasterport. This is done by the command:

POKE WINDOW(8)+29,-4 AND255

To ensure that the pattern is drawn correctly you must set the foreground to the highest colour register available and the background must be set to 0. This is done by:

COLOR WINDOW(6),0

Due to the complexity of determining the values for the bitplanes, I see little use in multicolour fills for complex designs, especially if you have more than three bitplanes.

The final graphics commands, **GET** and **PUT**, relate to the handling of blocks of graphics. The **GET** command captures a specified rectangular portion of the screen and stows it in an array. You can then use **PUT** to copy the array to the screen. The way in which the array is combined with the screen can be specified to obtain a range of effects.

■ PSET — This mode overlays any information already on the screen. As a consequence the picture looks just like the original.

■ PRESET — In this mode paints each point in its colour complement.

■ AND — In this mode, if the pen of the source pixel is the same as the pen of the destination pixel, the result is unchanged otherwise the result is pen 0. For a 2 bitplane system, the truth table has the following form:

Source Pen	Destination Pen	Result
0	0	0
0	1	0
0	2	0
0	3	0
1	1	1
1	2	0
1	3	0
2	2	2
2	3	0
3	3	0

■ OR — This mode ORs the pen values and puts the result on the display. Again here is a 2 bitplane truth table:

Source Pen	Destination Pen	Result
0	0	0
0	1	1
0	2	2
0	3	3
1	1	1
1	2	3
1	3	3
2	2	2
2	3	3
3	3	3

■ XOR — again the source and destination are combined (exclusive O Red this time) to get the result. The nice aspect of XOR is that if you PUT the same image twice at the same point the images are cancelled out. This is handy if you want to move images over the background whilst leaving the background unchanged. This is the default mode. The truth table is:

Source Pen	Destination Pen	Result
0	0	0
0	1	1
0	2	2
0	3	3
1	1	0
1	2	3
1	3	2
2	2	0
2	3	1
3	3	0

This concludes the introductory tour of the graphics commands available. Next time I will cover the use of sprites and bobs and start the highly fascinating area of library routines. With library routines you will really be able to get some interesting work out of AmigaBASIC.

YA

```
'
' A demonstration of some of
  AmigaBasic's graphics commands
' Allen Webb October 1988
'
title$= "Tester"
Xtopl = 0
Ytopl = 0
Xbotr = 300
Ybotr = 180
type = 22
DIM Design%(63),Des%(7),Des2%(3),
pat%(2200)
'
depth = 2
GOSUB OpenHiresScreen
GOSUB OpenWindowHi
GOSUB HiresDemo
GOSUB pause
WINDOW CLOSE 1
depth = 3
GOSUB OpenLoresScreen
GOSUB OpenWindowLo
GOSUB MultColPatt
GOSUB fadeit
END

WINDOW CLOSE 1

'
' Open a high resolution screen
'
OpenHiresScreen:
    SCREEN 1,640,200,depth,2
RETURN
'
' Open a low resolution screen
'
OpenLoresScreen:
    SCREEN 1,320,200,depth,1
RETURN
'
' OPEN a WINDOW
'
OpenWindowLo:
  WINDOW 1,title$,(0,0)-(300,170),
22,1 RETURN
'
OpenWindowHi:
```

Continued page 20 ▶

Castle Computers

NOW TAKEN

Castle House,
11 Newcastle Street,
Burslem,
Stoke-on-Trent,
ST6 3QB
Tel: 0782 575043

AMIGA SPECIAL OFFERS

Aaargh!	11.99
Action Service	12.99
Afterburner	14.50
Alien Strike	3.99
Alien Syndrome	12.99
Alternate Reality	12.99
A Mind Forever Voyaging	8.99
Army Moves	14.99
Around the World in 80 Days	12.99
Balance of Power	17.99
Ball Raider	3.99
Barbarian II	14.99
Bards Tale I	15.99
Bards Tale II	15.99
Bat	16.95
Batman	16.99
Battle Chess	15.99
Battleships	12.99
Bermuda Project	14.99
Better Dead Than Alien	12.99
Beyond The Ice Palace	15.50
Beyond Zork	16.99
Bionic Commando	14.99
Bismark	16.99
Black Lamp	12.99
Black Tiger	17.99
Blazing Barrels	12.99
Bombjack	14.99
Bone Cruncher	9.99
Bubble Bobble	11.99
Buggy Boy	14.99
Bureaucracy	8.95
Capone	19.99
Captain Blood	15.99
Carrier Command	14.99
C.D. Music	10.50
Championship Cricket	9.99
Chronoquest	19.99
Cogans Run	4.99
Combat School	15.99
Corruption	14.99
Crazy Cars	7.99
Cybernoid	14.99
D. Thompson Olympic Challenge	15.99
Dark Castle	8.95
Deep Thought	16.99
Deja Vu	17.99
Deja Vu II	16.99
Deluxe Paint II	46.99
Deluxe Photolab	46.99
Deluxe Video	46.99
Diablo	3.99
Division I	4.99
Dragon Ninja	16.99
Driller	16.50
Druid II	11.99
Drum Studio	12.99
Dungeon Master	15.99
Earl Weaver Baseball	16.50
Ebon Star	14.99
Elf	14.99
Eliminator	14.95
Elite	14.99
Emerald Mines	12.50
Empire	16.50
Empire Strikes Back	12.50
Excelon	14.99
Faerytale Adventure	18.99
Fantavision	24.95
Federation Of Free Traders	21.99
Fernandez Must Die	14.99
Fire And Forget	14.99
Fire Power	15.99
Fire Zone	16.99
Fish	16.99
Flintstones	12.99
Football Manager II	11.99
Fortress Underground	9.95
Frontier	15.99
Fusion	16.99
Gany Med	14.99
Garfield	12.99
Garrison II	14.99
Gee Bee Air Rally	15.99
Gettysburg	17.99
Ghosts and Goblins	16.99
Giganoid	10.50
Guerilla Wars	16.99
Hacker	4.99
Hacker II	4.99
Hard Ball	6.99
Helter Skelter	10.95
Highway Hawks	13.99
Hollywood Hijinx	8.99
Hollywood Poker	4.99
Hostages	16.50
Hotball	15.99
Hunt For Red October	15.99
Ikari Warriors	14.99
Interceptor	16.50
International Karate +	13.99
International Soccer	12.99
Instant Music	16.99
Iron Lord	17.95
Jet	24.99
Jigsaw Maniac	3.99
King of Chicago	19.99
Kwasimodo	3.99
Lancelot	12.99
Land Of Legends	16.50
Leather Goddess	9.99
Leathernecks	12.50
Live And Let Die	16.99
Lombard Rally Sim.	13.50
Lords Of The Rising Sun	21.99
Lurking Horror	8.99
Macadam Bumper	12.99
Manhatten Dealers	15.99
Mamiax	12.99
Mercenery Compendium	11.99
Mickey Mouse	15.99
Mindshadow	4.99
Moonmist	7.99
Morteville Manor	14.99
Motorbike Madness	9.99
Narcom 6	12.99
Nebulas	14.99
Netherworld	14.99
Night Raider	14.99
1943	17.99
Nord and Bert	7.99
North and South	15.99
Obliterator	15.50
Operation Nempune	15.99
Operation Wolf	15.99
Othello	4.99
Outrun	14.99
Overlander	14.99
Pac Boy	3.99
Pac Land	15.99
Pac Mania	14.99
Pandora	11.99
Paper Boy	16.99
Par 3	17.99
Peter Beardsleys Soccer	12.99
Phantasm	14.99
Platoon	14.99
Plundered Hearts	7.99
Port of Call	27.99
Powerdrome	16.99
Prisoner of War	19.99
Question of Sport	16.99
Rambo III	16.99
Return To Genesis	11.99
Revenge II	6.99
Robbeary	12.99
Robocop	16.99
Rockford	12.99
Rocket Ranger	16.99
Rocky	4.95
Rugby League	9.99
Sarcophaser	14.99
Scorpio	13.50
S.D.I.	18.99
Seconds Out	6.95
Sex Vixens From Outer Space	24.99
Shadow Gate	15.99
Sherlock	9.99
Shoot em Up Con Kit	15.95
Shooting Star	2.99
Sidewinder	6.99
Silent Service	15.99
Sinbad	18.99
Skate or Die	16.99
Keyfighter	3.99
Soccer Supremo	9.99
Space Harrier	14.99
Space Port	2.99
Speedball	15.99
Spider Tronic	12.99
Starfleet I	16.50
Star Glider II	14.99
Star-Goose	12.50
Starwars	11.99
Strange New World	7.99
Street Fighter	16.99
Strike Force Harrier	8.95
Stunt Man	13.85
Sub Battle Sim	17.99
Summer Events	13.50
Suspect	7.99
Sword Of Sodan	P.O.A.
Tanglewood	12.99
Test Drive	16.99
Tetris	6.99
Three Stooges	14.99
Thunder Boy	6.99
Thunder Cats	15.99
T.V. Football	19.99
Ultima 4	14.99
Ultimate Golf	14.99
Uninvited	17.99
Univ Mil Sim	15.99
Vectorball	9.95
Verminator	15.50
Victory Road	16.99
Virus	12.99
Vyper	9.99
Warlocks Quest	12.99
War Zone	3.99
Wec Le Mans	16.99
Weird Dreams	16.99
Western Games	4.99
Whirligig	12.99
Whitness	7.99
Winter Olympiad 88	6.99
Witness	7.99
Wizards Castle	13.99
World Tour Golf	15.50
Xenon	12.99
Zoom	12.50
Zynaps	14.99

JUST ARRIVED
SPECIAL CLEARANCE PRICE

Pro 5000 Black Joystick
RRP 14.95 Our Price 10.50

Pro 5000 Clear Joystick
RRP 15.95 Our Price 11.50

T.R.I.A.D.
Starglider
Defender of the Crown
Barbarian
ONLY 19.99

Have you missed our Special Offers?

Crazy Cars	7.99
S.D.I. (from Cinemaware)	9.95
Powerplay	8.99
Strange New World	4.99
Gettysburg	17.99
Winter Olympiad '88	6.99

Universal Disc Drive Cleaning Fluid	1.99
Antistatic Spray	1.99
Antistatic Foam Cleaner	1.99
Antistatic Screen Cleaner	1.99

or all 4 plus cloths, brushes etc
only £5.95

DON'T GET RIPPED OFF THIS CHRISTMAS COME TO THE PROFESSIONALS!!!!

Most goods sent by return ofpost. New releases send day of release.
Please note. We use 1st class post and offer 1st class service

We offer a fast reliable service by return of Post! Cheque's P.O's to:
CASTLE COMPUTERS
Any games not listed phone our Hotline Now on 0782 575043

CLUB 68000 LARGEST AMIGA CLUB IN EUROPE

DO YOU OWN AN AMIGA COMPUTER?

For everyone who owns one of these computers, **Club 68000** offers members software, hardware and accessories at huge savings **off recommended retail prices!** Each item has been carefully chosen to offer the best value and quality.

HERE'S WHAT YOU GET: When you join you will receive a **free disk** with a **games compendium. A free catalogue** (New every 3 months) on the **top commercial products with huge discounts (top games, top business programs, hardware and accessories).**

HERE'S WHAT YOU DO

Fill out the coupon below and return it to **Club 68000 Ltd.** Your only commitment is to pay £10.00 for one year's membership of **Club 68000,** Suite 1, Wickham House, 2 Upper Teddington Road, Hampton Wick, Kingston on Thames, Surrey KT1 4DP.

HARDWARE & ACCESSORIES

	R.R.P.	MEMBERS
3.5 Disk Drive — DDSD.1 Mbyte	109.95	89.95
5.25 Disk Drive — 40/80 Tracks, IBM	149.95	119.95
3.5/5.25 Disk Drive — 3.5 + 5.25	249.95	219.95
3.5 Disk Drive + Track Display	119.95	99.95
5.25 Track Display Disk Drive	169.95	129.95
3.5 + 5.25 Track Display Disk Drive	269.95	229.95
3.5 Disk Drive — A2000 Internal	99.95	79.95
20 Mbyte Harddisk — A500/1000/2000	499.00	429.00
30 Mbyte Harddisk — A500/1000/2000	599.00	529.00
40 Mbyte Harddisk — A500/1000/2000	699.00	629.00
60 Mbyte Harddisk — A500/1000/2000	799.00	729.00
512K Kam Expansion — A500 + Clock	129.95	119.95
1.8 Mbyte Ram Expansion — A500 Internal Card, without Ram Chips	149.95	119.95
2 Mbyte Ram Expansion — A500/1000	599.00	549.00
Midi Interface — Midi In/Thru/3 Out	39.95	29.95
Boot Selektor — External Drive Autoboot	12.99	8.99
Clock — A1000	69.95	55.00
Digiview Video Digitizer A500/1000/2000	169.95	149.95
Digiview Adaptor	29.95	19.95
Floppy Switch Box — for 3 Drives	49.95	39.95
Genlock — A500/1000/2000	299.95	239.95
Kikstart + Clock — A1000	149.95	119.95
Kikstart 1.3 + Anti Virus — A500	89.95	69.95
Sound Digitizer — Hardware	59.95	45.00
Sound Digitizer — Hardware & Software	69.95	55.00
Sound Digitizer — Prof. Hardware	129.95	109.95

PROFESSIONAL SOFTWARE

	R.R.P.	MEMBERS		R.R.P.	MEMBERS		R.R.P.	MEMBERS
64 Emulator	69.95	49.95	Haicalc	59.95	49.95	Pagesetter Professional	249.95	199.95
Aegis Animator	109.95	76.95	Introcad	59.95	49.95	Photon Paint	69.95	55.00
Aegis Draw Plus	199.95	139.95	K Comm II	29.95	21.95	Photon Paint Utility	24.95	17.95
Aegis Impact	69.95	48.95	K Data	49.95	34.95	Prism Plus	69.95	49.95
Aegis Sonix	59.95	41.95	K Gadget	29.95	21.95	Prowrite	89.95	79.95
Aegis Audiomaster	49.95	34.95	K Roget	49.95	34.95	Quarterback	49.95	34.95
Aegis Videoscape 3D	149.95	104.95	K Seka	49.95	34.95	Superbase Personal	99.95	69.95
Aegis Videotitler-	109.95	76.95	K Spread II	79.95	55.00	Superbase Professional	249.00	174.95
Aegis Diga	59.95	41.95	K Text	19.95	12.95	Sculpt 3D	89.95	65.00
Aquisition 1.3	249.00	186.95	Logistix	99.95	79.95	Studio Magic	69.95	55.00
Cambridge Lisp	149.95	104.95	Lattice C 3.04	169.95	118.95	Toolkit	39.95	24.95
Drum Studio	24.95	15.95	MCC Pascal	89.95	62.95	Turbo Silver 3D	139.95	99.95
Digicalc	39.95	27.95	Microfiche Filer	79.95	69.95	TV Text	79.95	59.95
Devpac	59.95	41.95	Macro Assembler	69.95	48.95	TV Show	79.95	59.95
Digipaint	59.95	41.95	Maxiplan A500	99.95	79.95	VIP Professional	99.95	79.95
Express Paint	69.95	59.95	Maxiplan Plus	149.95	124.95	Ultimate Soundtracker	69.95	48.95
Forms in Flight	69.95	59.95	Omegafile Database	24.95	15.95	Word Perfect 4.2	299.00	209.95
Goldspell	29.95	24.95	Pagesetter	89.95	69.95	X Cad Designer	460.00	399.95
Hercules Copy	24.95	14.95						

TOP 100 GAMES

	R.R.P.	MEMBERS		R.R.P.	MEMBERS		R.R.P.	MEMBERS		R.R.P.	MEMBERS
Aarghhh	19.95	12.95	Football Manager II	19.95	12.95	Overlander	19.95	12.95	Space Racer	19.99	12.99
Alternative Reality	19.95	12.95	4th & Inches	19.95	12.95	Obliterator	24.95	15.95	Strip Poker II	14.95	9.95
Army Moves	24.95	15.95	Fed. of Free Traders	29.95	19.95	Pawn	24.95	15.95	Star Goose	24.95	15.95
Alien Syndrome	19.95	12.95	Games Pack (5 games)	14.99	9.99	Pub Pool	9.95	6.95	Street Fighter	24.95	15.95
Auto Duel	24.95	15.95	Garfield	24.99	15.99	Peter Beardsley's Soccer	24.95	15.95	Sky Chase	24.95	15.95
Bad Cat	24.95	15.95	Garrison II	24.95	15.95	P.O.W.	29.95	19.95	Silent Service	24.95	15.95
Bubble Bobble	19.95	12.95	Gold Runner II	24.95	15.95	Platoon	24.95	15.95	Space Harrier	24.95	15.95
Barbarian	24.95	15.95	Gridstart	14.95	9.95	Phantasm	24.95	15.95	Tanglewood	19.95	12.95
Buggy Boy	24.95	15.95	Gunship	24.95	15.95	Quadralien	24.95	15.95	Terropods	24.95	15.95
Bionic Commando	24.95	15.95	Hunt For Red October	24.95	15.95	Quantox	14.95	9.95	Tetraquest	19.95	12.95
Corruption	24.95	15.95	Ikari Warriors	19.95	12.95	Roadwars	19.95	12.95	Thundercats	24.95	15.95
Capone	29.95	19.95	Impossible Mission II	24.99	15.99	Rocket Ranger	29.95	19.95	Triad	24.95	19.95
Captain Blood	24.95	15.95	International Soccer	24.95	15.95	Rockford	19.95	12.95	Trivial Pursuit II	19.99	12.99
Carrier Command	24.95	15.95	Kikstart II	9.95	6.95	Road Blasters	24.95	15.95	UMS	24.95	15.95
Chronoquest	29.95	19.95	King of Chicago	29.95	19.95	Revenge II	9.95	6.95	Virus	19.95	12.95
Daley Thompson	24.95	15.95	Karting Grand Prix	9.95	6.95	Sargon III Chess	24.99	15.99	Vectorball	9.95	6.95
Defenders of Crown	29.95	19.95	Leatherneck	19.95	12.95	Sentinel	19.95	12.95	Winter Games	24.95	15.95
Eddie Edwards Super Ski	19.99	12.99	Legend of the Sword	24.95	15.95	Skrul	24.95	15.95	Wizard Warz	24.99	15.99
Empire Strikes Back	19.95	12.95	Live & Let Die	24.99	15.99	Slaygon	19.95	12.95	World Games	24.95	15.95
Fright Night	19.95	12.95	Maniax	19.95	12.95	Star Glider II	24.95	15.95	Winter Olympiad	19.95	12.95
Foundation Waste	24.95	15.95	Menace	24.95	15.95	Star Ray	24.95	15.95	Wizball	24.95	15.95
Flight Simulator II	39.95	27.95	Motorbike Madness	14.99	9.99	Star Wars	19.95	12.95	World Darts	14.95	9.95
F.Sim II Scenery Disk	24.95	15.95	Nebulus	19.99	12.99	Street Sports Basketball	19.99	12.99	Whirlygig	19.95	12.95
Fire & Forget	24.95	15.95	Netherworld	19.99	12.99	Summer Olympiad	19.95	12.95	Xenon	19.95	12.95
Football Director II	24.95	15.99	Offshore Warrior	24.95	15.95	Sorcery Plus	19.99	12.99	Zynaps	19.99	12.99

CLUB 68000 INTRODUCTORY OFFER

Mail to CLUB 68000, Suite 1, Wickham House, 2 Upper Teddington Road, Hampton Wick, Kingston upon Thames, Surrey KT1 4DP.

Please send me ____________________ £ ____________________ £

____________________ £ ____________________ £

____________________ £ ____________________ Total £

Plus Membership only £10.00

Type of computer ____________________

Software: Free Post & Packaging within the UK. Europe £1.00. Overseas £3.00. Hardware: Courier service £6.00

Total enclosed £ ______

□ Cheque enclosed □ Visa/Access/Mastercard □ Postal Order □ Credit Card No. □□□□□□□□□□□□□□□□

Name ____________________ Signature ____________________ Expiry Date ____________________

Address ____________________ Telephone No. ____________________

Post Code ____________________ Please allow 28 days for delivery

DEPT YA1

OTG DSM

Exploring the Amiga and the software running on it can be a daunting task. Karen Young unfurls her skull and crossbones and sets sail on a voyage of discovery

■ **For many people, the Amiga is a** closed machine — closed to the operating system via a friendly and powerful WIMP environment, closed to the heart of the 68000 processor by three remarkabkle support chips, Angus, Denise and Paula which are designeed to serve the CPU. The Amiga is closed to the low-level software programmer by an almost frightening lack of decent development tools.

Of course many people feel that C is the ideal language for general software development on the Amiga. With C you have instant access to the very low level functions of the machine, and you also have all the advantages of a reasonably high-level language not to mention the speed of a compiled language.

At the heart of the Amiga is the Motorola 68000 processor, a fine and detailed processor with just enough opcodes to make it friendly enough to program at Mnemonic level, but it also has a small enough command table to make it a fast and reliable CISC processor (complex instruction set computer).

So if you *really* want to get to the heart of the Amiga, you have to resort to machine code, specifically a machine code assembler/disassembler package that will be powerful enough to help the user or programmer decode the millions of hexadecimal digits inside ever Amiga.

Roll On DSM

DSM is a disassembler for the 68000 processor inside the Amiga, OTG software, the creators of this package, claim that their package is the only full-featured disassembler for the Amiga capable of taking raw 68000 code, and convert it into neat 68000 mnemonics with complete accuracy.

After filling in all the forms and making the usual backup of the program disc or moving it onto your hard disc (if fitted), you then start the uphill struggle with learning this package — there are a number of different software programs on the master disc, and not all are needed. The contents of the DSM disc are:

BIN
The directory of executable programs

examples
The directory containing example programs

suite
The directory of assembler test programs

update.doc The file containing bugs and updates in the manual.

The BIN directory contains the files suitable for the DSM package, and a few public domain programs (speaktime, if you haven't already got it).

The Amiga version of DSM reads Amiga binary load files, that is, programs that can be clicked from the workbench (assuming it is assigned an icon) or typed in directly from the CLI, the process of disassembly often results in a source file that is ten to fifteen times the size of the original program, it is recommended that you have the internal 256K memory expansion fitted to your Amiga 1000s if you haven't already had it fitted. Amiga 500s and 2000s are quite happy with DSM's memory based disassembly.

Assembly sourse code output from a disassembler may be useful in a number of ways; for example, a programmer could use the sourse code in order to spot an error in a program, or in the language processor or compiler they are using. Alternatively, the disassembler's output could bej modified, reassembled and recompiled in order to produce a more efficient program.

By modifying the disassembler's output in this way, assembly language optimization of much-used subroutines could be achieved or, and this is the sneaky part, protection systems broken into, thus making it easier to make backup copies (legitimately of course!), for example it is legitimate to make a backup copy so that you may move your software onto a hard disc.

Disassembling certain programs may be against the law, so make sure you are willing to take the rap for any damages resulting from use of the disassembler - we're not saying you should use this program to copy, rather to make a security backup of a program.

Using DSM is easy. You don't need to have a deep understanding of the Amiga, or even an in depth knowledge of the MC68000 processor or even the totally freaked out Amiga load file structure! DSM is pretty easy to use by comparison by simply loading a file into DSM by: DSM [filename] -e You can load and translate a binary program (called (filename)) that would then be passed through an expert system which often leads to a better disassembly of the program code). This will then create a program called *progname.dsm* which would then be saved ready to be committed to a word processor or an editor or even the notepad!

Naturally all of this is done from the CLI — a far better way of getting to grips with the machine than through the mollycoddling of the Workbench — after the program is finished the 1) prompt is returned.

The -e option isn't the only option though, there is the -o option for ordinary translation and -h for HEX dump generation (ideal for looking at the memory map of a computer which is just data and not a command sequence). Possibly one of the most complex command options is the -s option for splitting the file up into more than one file and the -f option which is used to generate offsets, the (progname) file is thus turned into an offset file and not a source file.

Because DSM uses the CLI extensively, the user can have full control over the output of the generated file, be it HEX, offset or expert system assisted disassembly, that means you do what you damn well like with the filename, send it off to any output stream (including the screen or the printer), or just create the output file. The point is that unless you are using memory related disassembly facilities (such as attempting to disassemble zero-page), then you can use DSM as an interactive and memory resident disassembler (by using the COPY *all* TO RAM: command).

There is a utility supplied with the package called ATEM, this is a variation of the ATOM utility (AmigaDOS Technical Reference Manual).

ATOM was designed to assist programmers and, more importantly, programs so that object code files can be modified in order to control different memory types (CHIP, FAST or PUBLIC). ATEM performs this same function, but is operates on executable files (eg, programs) instead of object files.

I have used DSM for some months now, but have had no reason to use ATEM yet, although I dare say that someone will need it — I believe it is also available on the public domain.

Because the Amiga uses a form of optimized C object code as part of the operating system (BCPL in other parts using aj run time linker), the error messages are a good deal more arcane than one is normally used to seeing on other 68000 based operating systems (eg, the Mac), nevertheless, if you know where to find hunk tables and segment boundaries, then you will be okay.

The Amiga is a machine with more pitfalls than not, certainly low level programming on ones own is not a task I would relish, but at least DSM attempts to simplify the process of machine exploration (okay okay, I really mean piracy), but after about a month or so of playing around with the low level routines of AmigaDOS, I certainly came out with a better understanding of the machine, and even started a pet project based around the clanky old disc drives — certainly a new routine to speed the old things up had been knocking around inside my head for a few months.

DSM made the inner workings of the Amiga and the programs that run in it a whole lot easier to understand — the door has been opened, and after a few months I feel that I will be confident enough to move over to a more powerful and more interactive software development package. The Amiga is still a complex machine, sometimes annoyingly so, but at last I won't have to be tied down to the rigid environment one often comes across when using a compiler or even (dare I say it) an interpreter.

YA

Title: **OTG DSM**
Supplier: **OTG Software**
200 West 7th St
Suite 618
Fort Worth TX 76102

DR. OXIDE SLICES PRICES!

Comp-U-Save's cost-cutting clinician is at it again! This month Dr. Oxide, a bargain-basement surgeon if ever there was one, offers the industry's lowest prices on hardware and software, plus special deals on products available only through Comp-U-Save!

AMIGA PUBLIC DOMAIN

OVER 600 DISKS!

Largest Amiga PD Library in the World

also C-64 & C-128 — Write for Free Catalogue

Amiga PD — $4.00 each

BUSEXPANDER FROM BILL'S BOARDS

The only board for the Amiga 500 or 1000 that expands either machine to 12 slots! Fits in any baby AT case and provides 6 slots for the 2000, 6 for the PC (4 of those for the AT). Now you can use most of the expansion cards designed for the 2000 – hard disk controllers, 2/4/8 meg RAM cards, A2088 Bridgeboard, etc. Use low cost IBM-compatible expansion cards already supporting a wide range of business and scientific applications. Designed to work with auto configurable cards. Meets ZorroBus and Amiga 2000 Bus electrical specifications. Available exclusively through Comp-U-Save!

12 Slots for your 500/1000! Price: $495

Amiga Hard Drives 500 - 1000 - 2000

20 Meg - $585.00 32 Meg - $699.99 48 Meg - $799.99

Amiga Dual Drive 500 - 1000 - 2000

With Own Power Supply – $399.00

Amiga External Drive $137.99

Only Uses Half the Power of 1010 – with Pass Thru

Disk Drive & Monitor Extension

Cables 30"– $19.99

Panasonic WV1410 Video Cameras

For Digitizers – $204.99 16MM Lens – $29.99

Special 2400 Baud Modem – $154.00

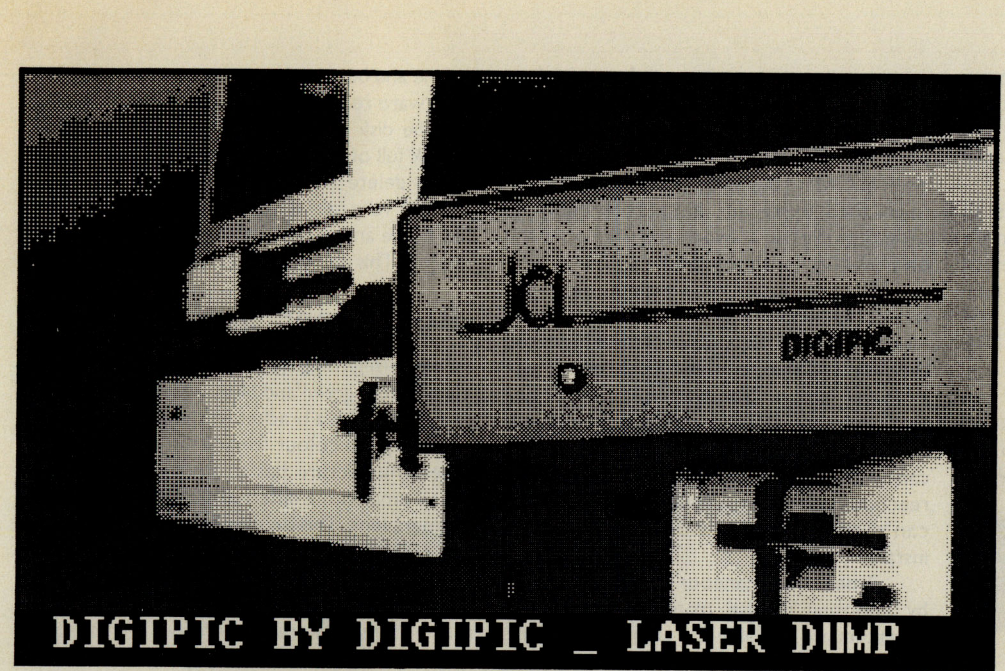
★ Trackball	$45.00
★ Plastic Diskbank (Holds 120 3.5 in. Disks)	$16.99
★ Copy Arm (Heavy Duty)	$29.99
★ Mouse Mat (Fabric)	$ 5.00
★ Mouse Mat (Teflon)	$11.00
★ Gender Changers – All Types	Call
★ Static Mat (23.5 x 25.5 in.)	$24.00
★ Rapid Fire Joysticks	$12.00
★ Printer Buffer (32K–512K)	Call
★ RF Modulator	$14.99
★ A/B Switch (Ser.)	$13.99
★ A/B Switch (Par.)	$14.99
★ A/B/D/E Switch	$29.99
★ Crossover Box	$39.99
★ Cables 500 – 1000 – 2000	Call
★ Teak Diskbank (holds 150 3.5 in. disks)	$39.99
★ Teak Diskbank (holds 200 5.25 in. disks)	$39.99
★ The Library (holds 80 3.5 in. disks)	$19.99
★ Floppy Wallets (Many Sizes)	Call
★ 3.5 in DS/DD Disks (Bulk)	$1.10 ea.
★ 5.25 in. DS/DD Disks (Bulk)	$ .39 ea.
★ Books – All Titles 15% Off	Call
★ Memory 512K – 4 Megs	Call

Call for Details

Please send International
Money Order for correct
amount in dollars.

Comp-U-Save

410 Maple Avenue, Westbury, NY 11590
In NY State (516) 997-6707 (Tech Support) Outside NY State (800) 356-9997 (Orders Only) Fax (516)334-3091



The Amiga screen resolution and art packages like Deluxe Paint have instantly raised expectations of what we should be seeing on our screens. Pictures captured from video camera or recorder meet those expectations. Ann Owen focusses on the facts about the DigiPic video digitiser.

Picture This

■ **Pictures from a video source are** digitised (turned into an image in digital form) by a special unit. They can then be saved in a format suitable for use in a paint or animation package, the standard Amiga graphics format known as IFF.

DigiPic is a plain box, but a box of tricks which achieves the above. It plugs into the parallel (printer) port of any model Amiga so you have to swap cables in order to use a printer and DigiPic. Remember to turn off when you do!

You'll need a cable to link your video source into the DigiPic unit but from then on it's a matter of focussing the camera.

Image processing

The DigiPic software can do a little bit more than just manage the process of capturing an image. There is a menu of image processing facilities which can help you prepare your picture for use. By using the REPORT function you can analyse the range of intensities in the pictures. CLIP can "trim" excess white or black.

BRIGHTNESS and CONTRAST work in the same way as they do on your television set. THRESHOLD switches levels of grey at one end of the scale to white, and at the other to black. QUANTISE reduces the grey scale to a smaller number of steps, 4, 8, 16, 24, 32 and so on.

HISTOGRAM EQUALISE attempts to correct an imbalance in the distribution of the grey scale intensities. EDGE provides what can be the very useful facility of edge detection. There is a choice of algorithms and you can't do any harm just experimenting with this one. It can be the first stage towards creating a line drawing out of the original.

Sharpening or softening an image is achieved with SHARPNESS. A slider gadget provides control over the rate of change in intensity. Finally there's PICASSO!, the results of which I'm sure you can guess.

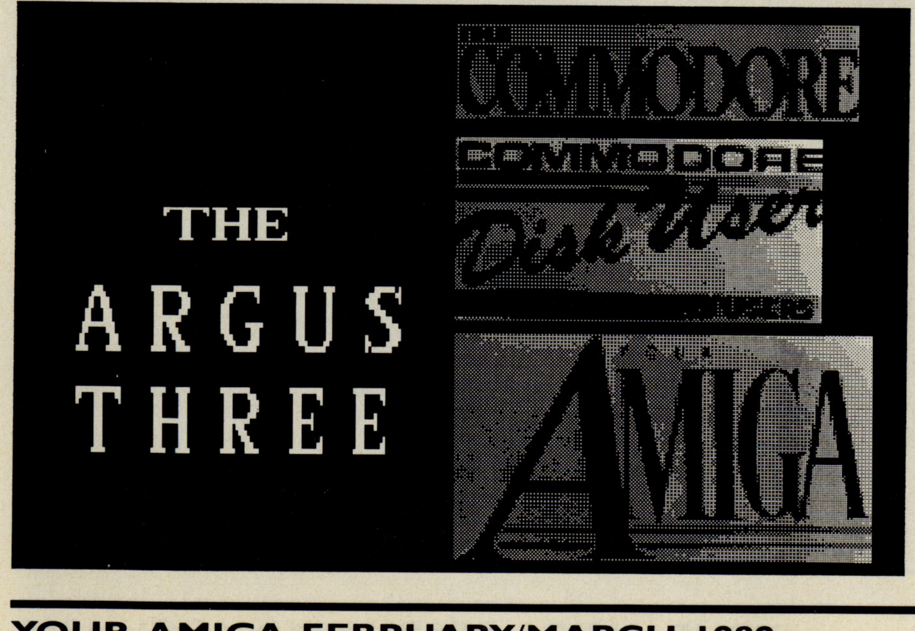
Grey shades

DigiPic provides an interesting attempt via hardware to provide 32 grey shades for captured images rather than the standard 16. The EGSA (Extended Grey Scale Adaptor) module connects between the computer and monitor, doing away with colour and turning the monitor to monochrome. The EGSA works with other software as well, such as Dpaint, but pictures created with the EGSA require its presence at all times to be properly displayed. If you are intending to use DigiPic for display purposes then this dependence upon hardware may not worry you.

To digitise or not to digitise?

The IFF screen pictures which resulted from using DigiPic were excellent and we've had great fun reworking them in paint and image processing programs. Digitised pictures can also provide interesting backgrounds and objects for animation in programs such as *The Stand* and *Animator's Apprentice*. The image processing software provided with DigiPic isn't, of course, as sophisticated as dedicated programs.

DigiPic could be used to catalogue a collection of objects or a group of people



because databases such as *Superbase Professional* and *Acquisition* support digitised IFF files connected to records. IFF files can also be displayed via BASIC or one of a dozen public domain utilities for creating slide shows.

For desktop publishing, digitising is an alternative to scanning flat artwork but it might be worth waiting for a professional scanner to emerge for the Amiga. After all you can take black and white photographs and scan the prints if you want "live" action in your DTP pages.

DigiPic is good value — especially if you already have the camera equipment — but no toy. I recommend it to you if you have an end product in mind or you can afford to play!

Supplier: **Precision Software**, 6 Park Terrace, Worcester Park, Surrey KT4 7JZ
Tel: **01-330 7166**

Action!

The camera supplied by *Precision* was a no frills unit which did the job well. The difficulties you will face in order to achieve a good image on screen include getting a consistently good light source and supporting and adjusting the camera. If you have a camera tripod it can, in all likelihood, be used with the video camera. The better the tripod the easier it becomes. The camera supplied could focus effectively up to about two yards. You can also digitise from other sources, such as a video recorder, but if it's a colour source you will have to adjust a link on the DigiPic hardware.

Capturing images of people and everyday objects is easy enough. They are large and, for black and white digitising, you don't even have to get them to stay still! It is possible but more difficult to digitise artwork and large print and only practical with a tripod.

The DigiPic unit has a separate video out socket to supply a second monitor with a "live" picture. If you don't have the hardware then the software comes to the rescue with a "loop" feature. The picture is digitised and displayed continuously, with a slight delay of course but quickly enough to be practical. You can hold the display by pressing Escape but the reaction time of the software means that you can't expect to stop the process instantaneously.

A near instantaneous capture is possible by choosing digitise from the menu (key shortcuts are available for all main functions) and pressing a mouse button or (on the latest release software) joystick button to activate the frame capture.

On a 512K machine there are three slots into which the software places its captured images. Beyond three you must save them slot by slot before overwriting. You can save in a four plane IFF format for compatibility with art and image processing software or in a six plane format for further use in DigiPic. Either type can be saved in a compressed format.

Filing activities take place via a typical Amiga requestor box for pathname and filename and buttons for drives zero and one. The software reads file information on every visit to the disk so things can get a bit slow with a disk full of pictures and their.info files. There is no delete file facility.

A very useful feature is that of being able to frame any part of the on-screen picture for saving. This can save both time and disk space.

Red, green and blue

Colour digitising requires the use of three filters, red, green and blue, which are provided together with a bracket to support them. You can only use colour for still life shots because the same picture has to be captured thrice, each time with a different filter in front of the camera lens. This isn't really a satisfactory way of creating colour images.

Firstly the set of colour filters fell apart without any prompting and secondly the finished results were colourful but not the same colourful as in real life. Despite considerable experimentation the colour results remained inaccurate, mainly I think because of the limited 32 colour palette. Here some guidelines from the experts might have been helpful. The DigiPic manual is excellent in its description of setting up the hardware and using the software but there is little general advice for the beginner to digitising.

The manual is full of technical information which is quite informative for the newcomer and will allow the experienced user to reconfigure the hardware should they so wish. *YA*

AmigaBASIC

(continued from page 14)

```
  WINDOW 1,title$,(0,0)-(600,170),
22,1 RETURN
'
DATA &haaaa
'
'Polygon data
'
DATA 1,1,  161,66,  310,90,
155,145,  1,1
'
'Hires monocolour pattern data
'
DATA &hF0F0,&hF0F0,&h0F0F,&h0F0F

'
' Multicolour pattern data
' Bit Plane 0
'
DATA &h0000,&h0000,&h0000,&h0000
DATA &hFFFF,&hFFFF,&hFFFF,&hFFFF
DATA &h0000,&h0000,&h0000,&h0000
DATA &hFFFF,&hFFFF,&hFFFF,&hFFFF
'
' Bit Plane 1
'
DATA &h0000,&h0000,&h0000,&h0000
DATA &h0000,&h0000,&h0000,&h0000
DATA &hFFFF,&hFFFF,&hFFFF,&hFFFF
DATA &hFFFF,&hFFFF,&hFFFF,&hFFFF
'
' Bit Plane 2
'
DATA &hFF00,&hFF00,&hFF00,&hFF00
DATA &hFF00,&hFF00,&hFF00,&hFF00
DATA &hFF00,&hFF00,&hFF00,&hFF00
DATA &hFF00,&hFF00,&hFF00,&hFF00
'
'
MultColPatt:
CLS
FOR x = 0 TO 200
    LINE (x,0)-(x,RND*150),RND*7
NEXT x
GOSUB pause
CLS
y=20
FOR x = 20 TO 200 STEP 10
    LINE (x,y)-(x+20,y+20),RND*7,bf
    y=y+3
NEXT x
GOSUB pause
CLS
    FOR i = 0 TO 47
    READ Design%(i)
    NEXT i
PATTERN -1,Design%
RasterPort = WINDOW(8)
POKE RasterPort+29,-4 AND 255
COLOR WINDOW(6),0
LINE (0,0)-(100,100),,bf
LOCATE 15,1
PRINT "A user defined multicoloured
pattern"
GOSUB pause
'
' Use of PUT & GET to create movement
' EOR is used to preserve the
  background
' Sorry about the flicker - direct
  access to the blitter does
' better job!
'
GET (0,0)-(100,100),pat%
LOCATE 5,20
PRINT "And move it!"
    FOR x = 0 TO 150
        PUT (x,0),pat%
           FOR i = 0 TO 100
           NEXT i
        PUT (x,0),pat%
    NEXT x
GOSUB pause
RETURN
'
'
HiresDemo:
y = 160
FOR x = 0 TO 400 STEP 5
    LINE (x,0)-(0,y)
    y = y-2
NEXT x
LOCATE 7,7
PRINT "Using LINE to draw a fancy
design"
GOSUB pause
CLS
READ Linepatt
PATTERN Linepatt
    FOR x = 0 TO 200
    LINE (0,0)-(x,100)
    NEXT x
    FOR y = 100 TO 0 STEP -1
    LINE (0,0)-(200,y)
    NEXT y
LOCATE 1,28
PRINT "Using pattern fills with LINE
and CIRCLE"
LOCATE 1,15
PRINT "Using patterns with AREAFILL"
    GOSUB pause

CLS
'
    FOR x = 0 TO 7
    Des%(x) = INT(RND*&H8000)
    NEXT x
    PATTERN -1,Des%
    CIRCLE (400,100),100
    PAINT (400,100)
GOSUB pause
CLS
    FOR i = 0 TO 4
    READ x,y
    AREA (x,y)
    NEXT i
    AREAFILL
    FOR x = 0 TO 3
    READ Des2%(x)
    NEXT x
    PATTERN -1,Des2%
    CIRCLE (400,100),100
    PAINT (400,100)
LOCATE 5,5
PRINT "A user defined pattern as
described in the text"
RETURN
'
'Simple colour fade routine showing
how to change colours using
'PALETTE.
'
fadeit:
CLS

LINE (60,60)-(230,160),3,bf
more:
    red= RND: green=RND: blue=RND

FOR i = 1 TO 200
tweak2=i/200: tweak1=1-tweak2
PALETTE 0,red*tweak2,green*tweak2,blue*tweak2
PALETTE 1,red*tweak2,green2*tweak2,blue*tweak2
PALETTE 2,red*tweak2,green*tweak2,blue*tweak2

PALETTE 3,red*tweak1,green*tweak1,blue*tweak1
NEXT i
GOTO more
RETURN
pause:
    FOR i = 1 TO 10000
    NEXT i
RETURN
```

VIP

VIP stands for Virus Infection Protection, but Karen Young discovers that it also stands for Very Important Program!

■ **Viruses are big news at the moment** — two years ago, the happy pastime for spotty little reprobates with a predilection for pre-stale cheese bars was hacking, that innocent and (mostly) honourable sport whereby one logs onto a computer, finds one's way into the system and then leaves a little message to tell the system operator that you are a clever little reprobate with a predilection for stale cheese bars.

But in this sexually-aware, AIDS-conscious society (well, we can all hope can't we?) Computer users are living under the shadow of their own silicon equivalent of the HIV virus — The computer virus.

The computer virus first started off as a joke, probably some computer programmers though it would be a net idea to write a program that lived in the machine somewhere and copied itself automatically, so that on a predetermined time, members of a computer club all got the same message on their disks — the virus was safe because it had a self-destruct device built into it so that after the message, it killed itself off.

This is a safe virus, and virologists create these all of the time for example, bacteria that are spread over an oil slick break down the oil and then die; they are not given a chance to reproduce after a certain number of generations and (moral issues aside) this is generally considered to be the safest way of dealing with oil slicks. (Bacteria behave in a quite different way to viruses, but I think the analogy is till relevant).

But the important factor is this security device, the self destruct, if a computer virus doesn't go unchecked, then it will spread and spread until it reaches your machine because successive generations are not wiped out to feed the new generation (as is the case with many traditional viruses). This has been a real problem for the Macintosh and the Amiga, and now that the problem has reached epidemic proportions, somebody is doing something about it.

Writing for a magazine has the additional problem of obtaining disks from dubious sources that you cannot be certain are infected or not, for example, one virus, the SCA virus was effectively wiped out by Datel's SCA virus killer program. The disk it came from? Who knows? But it certainly came to me care of some spotty little reprobate.

There are cures for these viruses of course, and for a while, the SCA virus killer did the trick, but there are well over 35 known viruses and, unfortunately only a very few of the virus killers can deal with them all.

Enter *VIP*, a totally new concept behind virus protection, so simple and straightforward it will make you kick yourself when you find out how easy it is to not only protect your most important disks, but your entire disk collection as well!

VIP is simply a librarian program for any Amiga, be it a 500, a 1000 or a 2000, it simply creates a special disk and creates a hierarchical database of standard and customised bootblock information (the place where viruses are supposed to exist), but not only that, *VIP* enables the user to look at the disk, examine the virus strain, and then take preventative measures such as protect a disk from infection, and also inform the user when a virus is active in the machine.

The Amiga 1000 is probably the most prone to infection as it has an area of memory that is both hidden from the user, and read/writable (whereas it is only readable on the 500 and 2000), this is, sadly why viruses are so popular for the Amiga, as many users out there have 1000s with this nice and warm area for viruses to live in.

Don't feel left out though, all you 500 and 2000 owners, the viruses are just as happy living on your disks as well, in fact, they spread and do just as much damage regardless of machine, it is just a little harder to rout them out on a 1000.

Viruses know nothing of international barriers or quarantine, but no matter, *VIP* works in a number of different languages, and it even stores bootblock information from protected games disks — yes folks, even the highly protected games for the Amiga tend to leave the bootblock alone, thus enabling users to happily archive the bootblock into the database for later retrieval — assuming, that is, you contract a virus in the future!

VIP comes in some neat packaging with the disks, a manual and some stickers; these sitckers are to go onto the disk once they have been *VIP*'ed and then they are numbered according to their position in the database (although you can access the database by name).

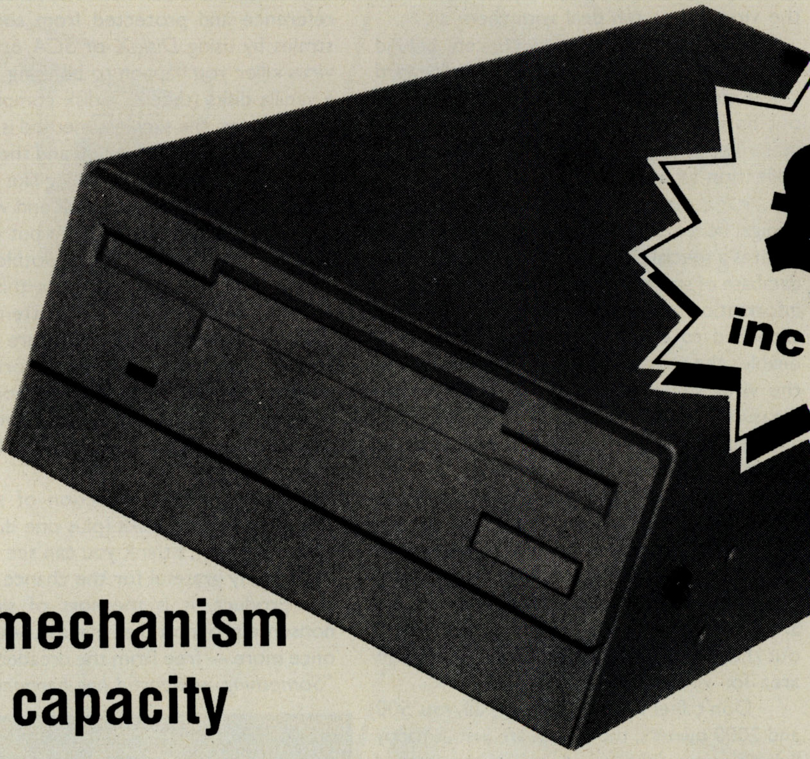
The process is very simple, the disk goes in, it can then be examined for the presence of a virus, and it can then be stored away for reference and protected from some future strains by using *Disk-X* or *SCA* or whatever virus killer you happen to be using. To create a vanilla disks (that is, a disk completely free of all viruses, the simplest method is to format a blank disk from power-up and then save the disk as "Vanilla" or something like that, then every time you format a disk and want to be sure it is free of Viruses, you put it through *VIP* to create a nice clean bootblock.

If you are careful and use this program with *Virus-X* and *Disk-X* (written by Steve Tibbit who upgraded this software by making some improvements to the original *VIP* program, a much simpler beast), you will have a combination that will not only knock out all of your viruses, but it will also help kill off this stupid joke.

Having a disk collection of some 200 Amiga formatted disks (plus one dual format ST/Amiga disk), I think you can see why I was particularly grateful for the chance to review this product, it is the sort of simple, no-nonsense program that makes computing fun once more — free from the dreaded message, *"Something wonderful has happened..."*. ***YA***

Title: **Virus Infection Protection**
Supplier: Discovery Software
International
163 Conduit Street
Annapolis
MD 21401 USA
Price: **TBA**

2ND AMIGA DISK DRIVE



£85

inc VAT & P&P

NEC mechanism
880K capacity

By selling direct we can provide the best price and maintain the highest quality. Rest assured that if, for any reason, you do not wish to keep the item, then return it to us within 14 days of purchase and we will refund your money in full. Should any item purchased from us fail during the first 12 months then we will repair it free of charge.

2ND PRINTER INTERFACE

£29.00

inc VAT & P&P

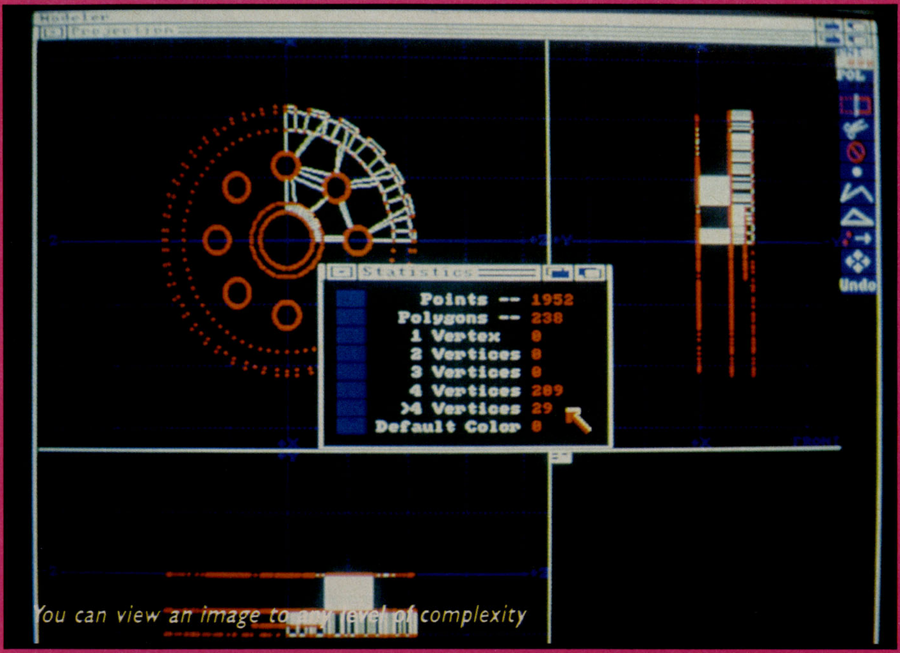
Use your Amiga's serial port to drive a second Centronics compatible printer. A 3 metre cable is included - no extras required.

MIRACLE SYSTEMS

NBC, Dean Road, Yate, Bristol BS17 5NH
Telephone orders welcome on (0454) 317772

Aegis 3D

A 3D package with the ability to animate images on screen is a tantalising thought. Karen Young finds out more



Points -- 1952
Polygons -- 238
1 Vertex 0
2 Vertices 0
3 Vertices 0
4 Vertices 209
>4 Vertices 29
Default Color 0

You can view an image to any level of complexity

Over the past few months there have been a plethora of different 3D and object-orientated drawing/drafting packages for the Amiga; as one may suspect, these packages have all had their various advantages and disadvantages, some have been suitable for rendering technical illustrations whereas some are more suited to producing graphically pleasing images or demos — something the Amiga has proved itself to be best at.

As far back as July 1987, Aegis development Inc released *VideoScape 3D* (version 1), *VideoScape* represented something that the Amiga has taken to its heart, Interactive family software — A suite of interconnecting software packages that deal with graphics and sound and then link together to form a greater whole.

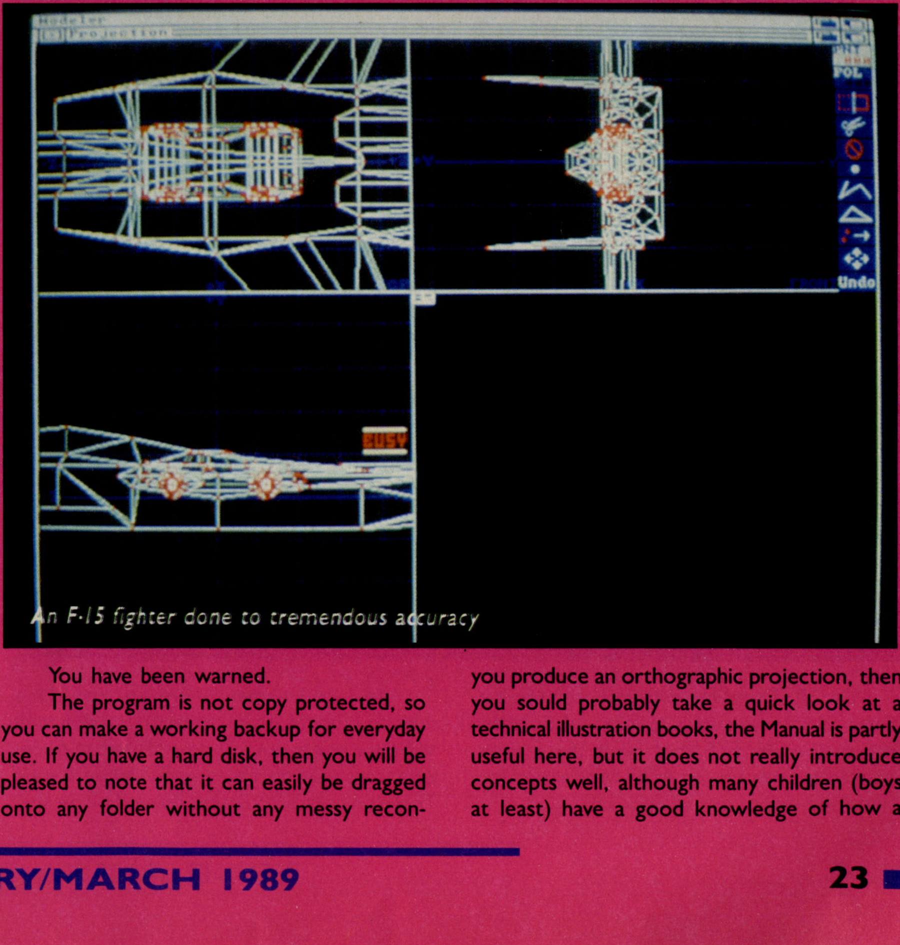
Now, interactive graphics are not new for $5000 systems or dedicated VAX machines running decent graphics based software packages (try out the GPX terminal — a snip at $40,000!). Anyway, the point is that unless you are going to use object or model orientated drawing packages on these afore-mentioned minicomputer beauties, then the ability to create graphic images made from smaller building block images (such as a polygon) or indeed an irregular image would be very difficult on any microcomputer — in fact, it would probably be impossible.

Modeler 3D(sic) is Aegis' attempt at solving all this — on an Amiga 500 (basic model, no frills) you are able to create images of such complexity, that five years ago few people would have believed possible.

Minimum Requirements

As I mentioned earlier, *Modeler 3D* will happily run on any of the Amiga family of computers, you don't need the full megabyte of memory, you don't need more than one disc drive, and if you have an Amiga 1000, the only prerequisiste is to have a Kickstart 1.2 on power-up.

That said, I really would recommend you have a full megabyte of memory as a 512K machine will require the slightly arduous task of closing off each window before starting work on another one! This is catered for most of the time, but occasionally *Modeler 3D* did crash when attempting to open up more than one window.



An F-15 fighter done to tremendous accuracy

You have been warned.

The program is not copy protected, so you can make a working backup for everyday use. If you have a hard disk, then you will be pleased to note that it can easily be dragged onto any folder without any messy reconfiguring that unusally has to be undergone.

Many users will want to enter *Modeler 3D* from the CLI, this is easy to do. By simply typing M3D from the "1)" prompt, you can directly enter the program — naturally the software exists so that CLI input is unnecessary when in *Modeler 3D*.

There is the usual bitch of a time waiting for the Amiga's clanky old disc drives to get their act together and load the program into memory and after a while you will be presented with the main editing screens. These screens are what you will use to enter information such as position, shape, co-ordinate and polygon commands, but more of this later.

The *Modeler 3D* Universe

It may sound a little pompous calling what you are about to design a "universe", but in many ways, it is the most apt way of getting to grips with a new world on your screen that you have total control over, if you want a point to appear, then it will, if you want to create a shape or a basic building block, then it will.

If you are unfamiliar with the way a 3D image works, or if you are unsure about how a rendered image looks on the screen when you produce an orthographic projection, then you sould probably take a quick look at a technical illustration books, the Manual is partly useful here, but it does not really introduce concepts well, although many children (boys at least) have a good knowledge of how a technical drawing is put together, both in orthographic and isometric projection.

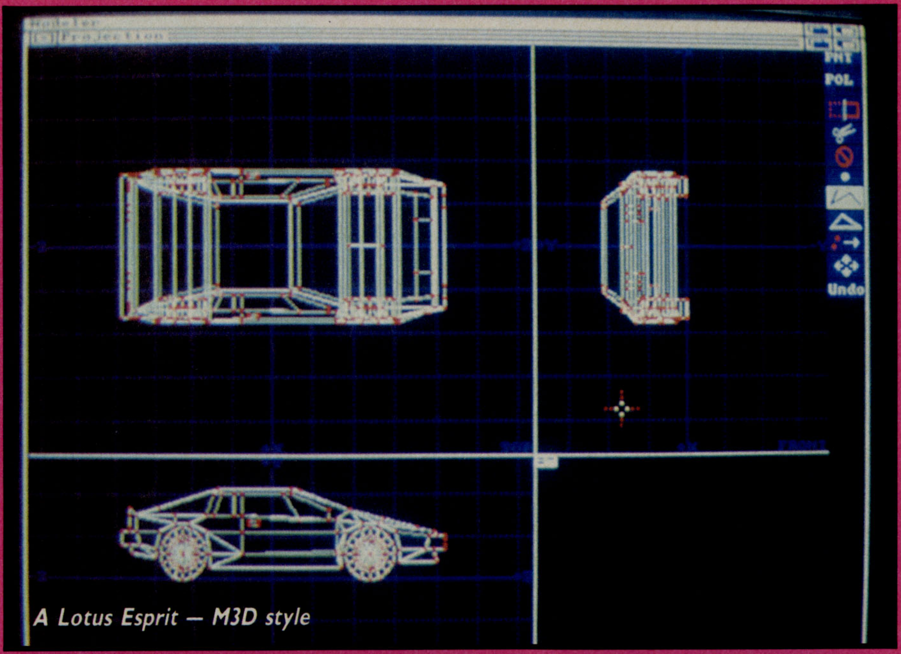
When you start up in *Modeler 3D*, you will first see the menu bar and the projection window, with its tool bar and all the usual display gadgets. These tools can be built up from objects in the window currently selected (there are four in all) and the screen area is, naturally enough, divided up into four, although each window can be enlarged and moved about if needs be.

In real life, we genereally tend to look at an object with two eyes, working with two simultaneous views of an object; with *Modeler 3D*, you can look at any of the three sides that make up a correct reperesentation of a 3D image. You can flip from one view to the other with ease. For safety's sake there is a command that makes it impossible to move out of one particular plane, as if you were working with a 2D drawing, and there is the excellent option of putting the pieces in different drawing layers, so that when you are finished working with the layers, you can put them back in one single layer.

The LAYER gadget lets you choose which layers to display (you can display more than one at any time), you can also choose which layers are active, thus selecting which images are capable of being edited.

There is a GRID gadget which is used to indicate the current scale and is displayed in X,Y,Z co-ordinates (tri-ordinates). and they can be enlarged or shrunk to any scale you wish.

The tool bar is a little different. There are a number of different tools that all interract differently. For example, the MODE selection tool lets you choose a level at which to work; In point mode your command entries affect points, so in Polygon Mode your commands will affect polygons — likewise Volume Mode deals in a similar fashion with either points or polygons that are enclosed in a defined volume or area.



A Lotus Esprit — M3D style

Impossible Designs — care of M3D

The CUT tool removes a chosen section of the image and pastes it to another layer, likewise the DELETE tool takes these away completely whereas the ADD tool is used for generating points out of thin air.

The CURVE tool is a simple two-point polygon generator, suprisingly enough, it cannot generate bezier curves, so its effectiveness is limited to simple circle orientated shapes.

Of course there is an UNDO option which works only on the last action, although other actions can be simulated by clever ADD and CUT options.

To explain how the point and curve options work it would be best to use the analogy of a board with pins that are nailed into the board so that rubber bands can be stretched from one pin to the other. The difference is that with *Modeler 3D*, you can place pins in any of three dimensions (the equivalent of placing pins in solid air) and that these pins can be any thickness and any length you care to imagine.

As I said earlier, there are the three selection modes (Point, Polygon and Volume), and since points are just co-ordinates in space, they can only do three very limited things; They can only be created, deleted or moved. Ways of moving these points around include translation, rotation, scaling, remapping etc. These types of systematic motions are called "transformations".

Polygons can be created and deleted, but they cannot under any circumstances be moved. The points which represent its vertices *can*, however, be moved. So connected is a point in the 3D universe that the simple movement of a single vertice will change the fundamental shape of an object in all three dimensions.

Polygons can, for the benefit of the user, be coloured in — not very useful in itself, but it is ideal for setting up a sense of depth. These polygons can also be flipped or have their vertices added (or even removed) — some vertices can be split or merged although these can only be perfomed on selected polygons using the arduous "Direct Edit" commands whilst in Polygon Mode.

Direct editing involves taking each point and editing each view so that each point in each of the three editing windows is subsequently moved, rotated, or whatever.

Volume mode selects all of the points and/or the shapes at the same time. The points selected are those that lie within the selected area (the volume). Selected shapes (polygons) lie within the volume area, so that their vertices are all selected *points*. The polygons that span the border of the volume are either selected or not, depending on the volume mode currently set up. In *Exclusive Mode*, those spanning shapes are not selected, whereas in *Inlusive Mode* they are.

The CUT, DELETE and COPY operations are all called *Universal Commands* (as are the MODIFY menu options). These act on all of the active points and polygons (even the points that make up a polygon).

The Universal Commands are all very convenient as thay enable the user to affect everything at once, thus making it unnecssary to go to each point in each window to make sure everything is just right. These rules apply in all the other modes (Polygon and Volume), although the way they affect co-ordinates in POINT mode is more dramatic.

Several operations make copies of any or all of the active items. For example, the COPY option (quite obviously) takes each object in the currently selected window and duplicates it in all other windows. So, if COPY is selected when nothing is *explicitly* selected, everything in that particular layer will be highlighted. The upshot of all this is that you will be able to quickly duplicate your work, reposition it, and then lay it down in any other layer.

The deleting of polygons can be a little difficult. Deleting a Polygon (to use the manual's idiom), "just unhooks the rubber band, and throws it away!". Deleting a point is like deleting the hook on which the rubber band is connected. This will alter the way the other rubber bands relate to each other and thus the polygon undergoes a topological change. Geometrically it is different (and in volume it has changed), but the relationships between the points all remain the same, even though one rubber band has just twanged off into the formless void — the balance has been kept.

UNDO, is quite possibly one of the most powerful items, in that it enables you to try out ideas, and then discard them and then reinstate them *ad infinitum*. I found that UNDO was great for just whapping down a few "rubber bands" and trying out, if they interfered with the drawing in some unexpected way, or if they just did not look right, then I could quite easily UNDO the operation and that's that! Of course you can undo an UNDO operation.

Documentation

The manual takes the user through all of these steps in easy, but verbose, lessons. I was able to get to grips with the main functions of *Modeler 3D* in under 45 minutes, the rest — they say — is up to art — The more you experiment, the better you get to know what the program can, and cannot do.

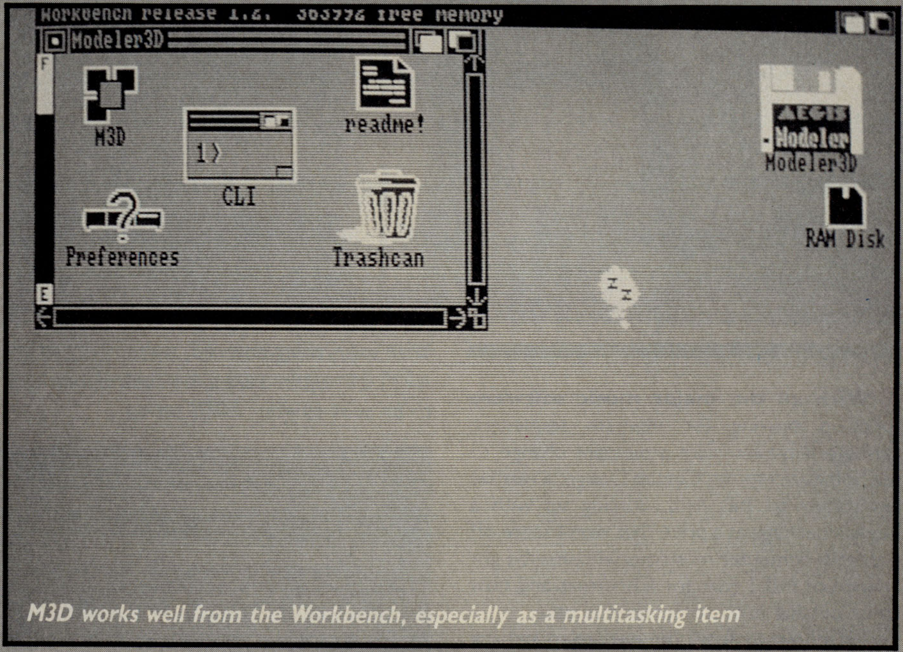
As *Modeler 3D* is designed primarily to tie-in with VideoScape 3D, you can actually make some pretty nifty animated technical illustrations with the shapes generated with the package, you can create something called a *Camera Motion File* which gives the *Modeler 3D* drawing the ability to appear as if it is moving on the screen.

By setting up a path along which the imaginary camera (which outputs to the screen) will travel, you can easily set up *fly by* illustrations, pan in and outs, and all soarts of camara trickery with the ease of clicking a mouse button. You can state how many frames thare are between each point, and you can even select colours at this point. (although you will have to have access to a colour/number chart) because the image is still a wire-frame image and is, thus, not very colourful.

Along the path you want your camera to slide along, you will have to cut it up into frames, experimentation is the key, so you will have to get the right trade-off between speed and jerkiness. The old trick of having various images pan out of sight will not help much (as you can do with more conventional VideoScape drawn Motion Files) as the points are held in memory all the time, and any movement recalculates the points of all the images in the *Modeler 3D* file. By ending up where you begin, you can get wonderful looping images that loop over and over again.

Aside from all this, after reading *Dirk Gently's Holistic Detective Agency* I decided to re-enact a scene in the book on a friend's Amiga (mine is undergoing repairs at the moment) and, after a few hours getting the wire-frame of a particularly nice sofa, I managed to get a really great image of a sofa spinning around in space! Now I'll bet you will go and read the book now won't you?

Still, this example is a good one as I shall explain. You see, most of the 3D programs attempt to incorporate all of the motion facilities and animation facilities into one package — Aegis, like Electronic Arts' up and coming program have gone the opposite way, they supply a separate package that will take the file created by one package, and animate it using those facilities, thereby concentrating on getting a program that specialises, and therefore does the job better (in my humble opinion that is). I saw *Forms In flight*, and was not impressed by this facility, although it does perform various functions a little easier that most. -



M3D works well from the Workbench, especially as a multitasking item

It is possible to take the image created using *Modeler 3D* and turn it into a proper ANIM file by simply processing it through VideoScape 3D, I personally liked this facility as it enabled me to take a drawing or illustration and incorporate it in some useful ways, for instance, I was able to look at the way a garden shed would affect various surrounding patches of land at different times of the day, and was able to position the shed at the right angle so as to avoid plants and flowers that require the Sun.

A slightly more useful end would be to have *Modeler 3D* and VideoScape 3D spin an image in free air, thus making it easy to see how various objects would appear in real life — this is called *Product Visualization* and is a big thing in the states. It also makes groovy video effects and is, of course easily able to create Isometric renditions of technical drawings from Orthographic projections, thus teaching people about the complex relationships between movements of objects or points of objects in the X, Y or Z planes. If used properly, it could be a great boon for schoolteachers wanting to teach the fundamentals of plane geometry and locational matrices as well as offering a practical way of depicting movements along axes in real time.

Conclusions

I found *Modeler 3D* to be an excellent tool, initially for pleasure, I admit, but it soon found itself as a viable educational tool, and an excellent teaching tool.

Modeler 3D seems to me to be the most straightforward of all of the 3D modelling packages originally popularised by Byte By Byte's *Sculpt 3D*, although *Forms in Flight* looks to be a worthy competitor, and is likely to give *Modeler 3D* a run for the money.

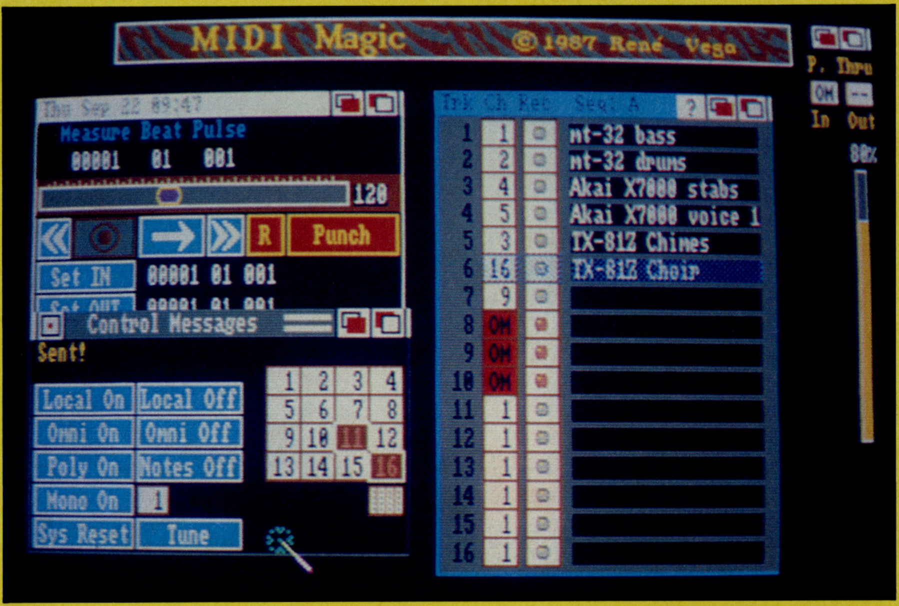
I particularly liked the straightforward way of being able to plonk an image on the screen and directly manipulate that image — if you are looking for just a rendered 3D image then fine, you need not worry too much about the number of points and links you make (to use the old analogy of pins and rubber bands again), but if you are using *Modeler 3D* to create 3D images for animation and rotation, then some care must be taken so that the computer is not having to recalculate too many points at any one time.

All in all a good package — the only minus points I can put to it are that the colour facility could have been implemented better, and that it is a little finnicky when it comes to working with some accelerator boards. If you want a good 3D package for the unexpanded Amiga 500, then this one is ideal as it is simple, reasonably low cost, interfaces with other animation packages, and is, of course, a mite cheaper than a VAX or anything available in the boring world of minicomputer 3D packages.

Title: **Aegis 3D**
Supplier: **HB Marketing Ltd**
Brooklyn House
22 The Green
West Drayton
Middx UB7 7PQ
Tel: **(0895) 444433**
Price: **TBA**

If your idea of MIDI Magic was making a synth disappear read on as Darrin Williamson puts the record straight

MIDI Magic



With everything entered, we're ready to go. All that's required are a few control messages to be sent to your MIDI set-up

With all the major music software suppliers striving for ultimate sequencer package it makes quite a pleasant change to see Software House; Circum Design, a relative newcomer to this field, coming up with a flexible package at a price that won't scare off potential computer musicians who, maybe got a piece of MIDI gear for Christmas and now fancy putting it to some serious use.

It's obvious from the packaging and the price that this product is designed to both educate newly converted MIDI musicians and act as a useful production tool for a musician of any level.

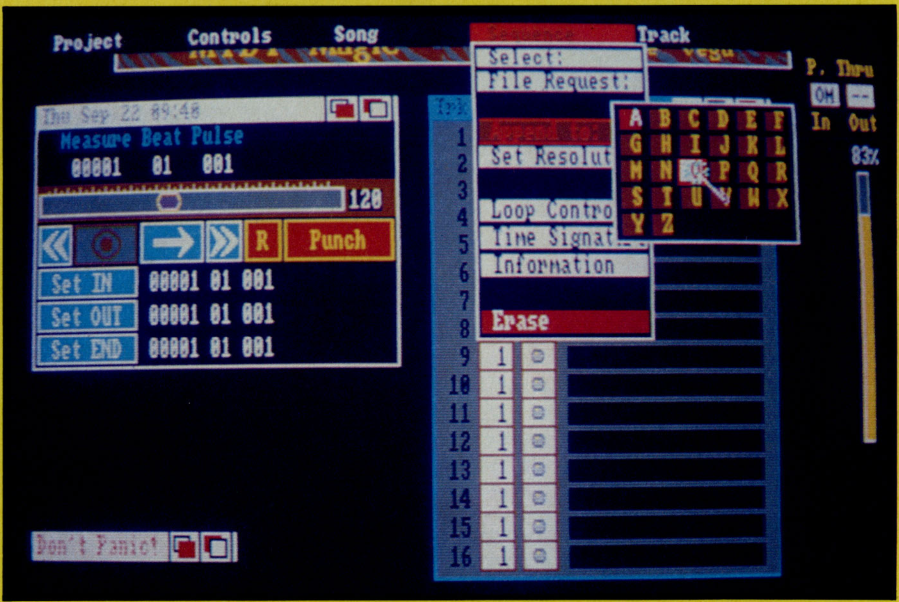
The software runs on a 512K machine but does take up over 70 per cent of the memory so I would recommend some sort of memory upgrade if you really want to use this package seriously. It's obvious right from the start that the *MIDI Magic* was written with the Amiga in mind as the whole program is WIMP driven and runs quite happily as a foreground task with additional programs held in memory as well. In fact Rick Hamouris, author of the instruction manual hints at a few additional "Modules" to compliment their sequencer. I can only assume that these will include things like a generic patch librarian or patch editors for leading makes of synth. I'll keep you posted as these become available.

Upon loading *MIDI Magic* you are greeted with a screen full of familiar looking Amiga gadgets. On the left of the screen are the "tape" transport controls and a horizontal tempo slider. Also within this section are your punch in/out setter which allow you to begin recording a pre-set point from within the song. Just below the transport controls is the Don't Panic button which come in very handy if you get any stray notes that refuse to shut up when you stop a sequence mid-way as used to be a common problem with early DX-7s. All this device does is send an All Notes Off MIDI message across all 16 channels.

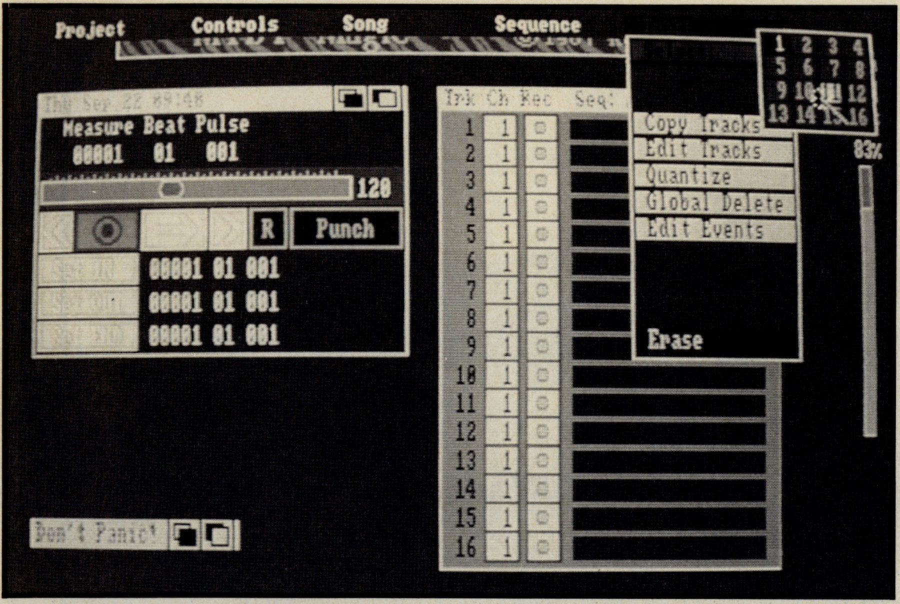
To the right of the screen we have the track list. As you can see you have 16 tracks to record on. This may seem a bit of a limitation but bear in mind that each of these will receive MIDI data on any, or indeed all MIDI channels. From here individual tracks can be labelled for convenience, muted or soloed (cut out or left as the only thing left playing) just like a multi-track tape recorder. On the extreme right of the screen is a vertical memory bar which shows you just how much workspace you have left. This is generally blue but turns yellow when 80 per cent full and flashes red when 98 per cent full. If at that stage you find yourself in dire need of that last ounce of memory you can switch out the workbench which may be just enough to get by.

Just above the memory meter is the play through gadget which allows you to alter the channels of reception and re-transmission so if, for instance, your main keyboard only transmits on channel one you can set the play through IN to one and set the out to whichever channel you need for that track. So, provided your existing keyboard has all the performance features you require you don't have to rush out and buy a master keyboard. You may have noticed at this point that little or no space is wasted on-screen. Even the title bar has a function as it acts as a visual metronome when recording. The philosophy has been to get everything on one screen rather than getting lost flipping between lots of different screens

The menu bar has five sections to it; Project, Controls, Song, Sequence and Track. Project contains Help options (a strong feature of this product), workbench on/off facility and a choice of medium or high resolution screen



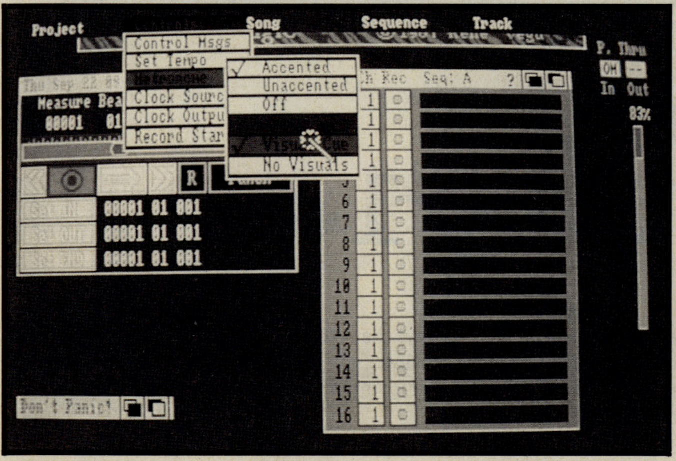
Lo and behold, your sequence options appear. Looping, time signature and resolution are all tweakable from here



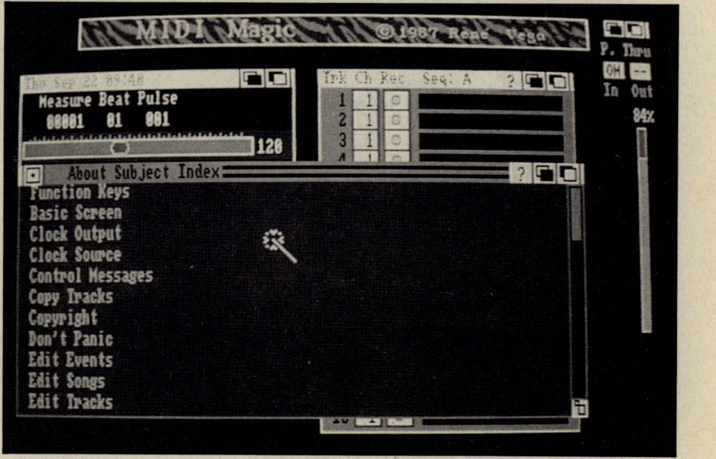
A highly flexible set of track editing facilities can be accessed from the Track Bar

modes. The package defaults to medium res but switching to high res allows more windows on the screen at one time (at the expense of some memory

The Controls bar offers a number of MIDI utilities which streamline the recording process. Specific MIDI modes can be altered for all MIDI channels. Additionally tempo can be altered from this menu by tapping a key on



Every effort has been made to make the recording process as painless as possible



One of MIDI Magic's best features is the extensive information windowing. As you can see, lots of help is available

your synth or the space bar which should please all those die hard rhythmists out there. Syncing protocols, both in and out can be tweaked. So you can select the internal clock or you can get the sequence to wait for MIDI timing codes from a drum machine or another sequencer (the latter is useful if you are transcribing songs from a hardware sequencer to *MIDI Magic*. You can even put it into step mode which allows you to tell the clock just where in the piece to go and how fast. This feature opens up a window with two rows of notes which can be inserted at any point. Similar rules apply on the clock output stage. You can send out a MIDI clock to a Drum Machine or a SMPTE unit. Alternatively, a non MIDI Drum sync can be out put to a Drum Machines without the provision of a MIDI clock. Finally we have the Record Start options. Here you can have the option to begin recording with one measure count in (you know, one, two, three, four) or you can loop round the first measure until you play something or you can begin recording from when *MIDI Magic* receives its first Note on message (very useful if your Amiga isn't at arms reach from your synth.

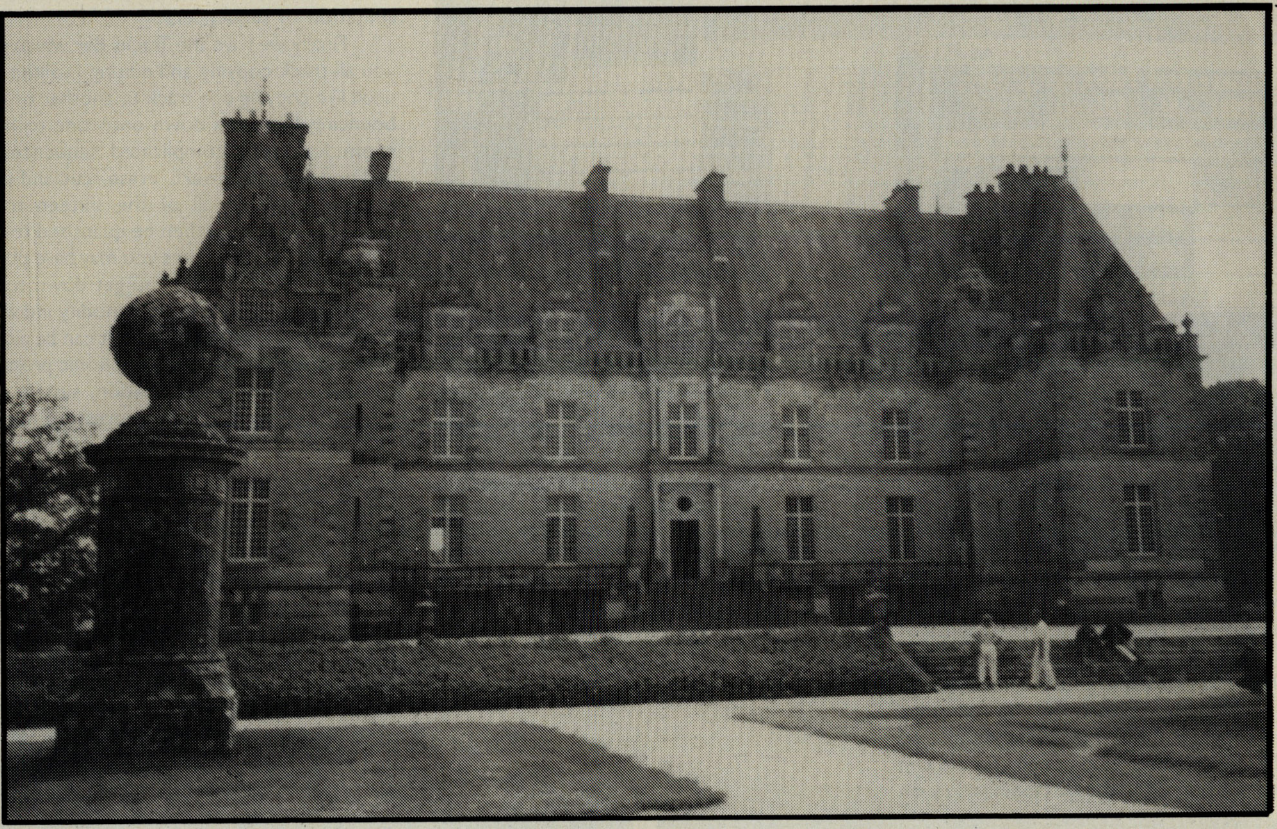
Next on the list is the Song Editor which allows you to arrange sequences into songs in a very straightforward manner. From this menu you can pick which sequence you want, what order they go in, how many times they repeat (if at all), what tempo each sequence should be and whether sequences should be transposed up or down in pitch.

The Sequence bar does much the same sort of thing only in this case for individual sequences rather than songs. Here you can set note resolution up to 480 PPQN (Pulses Per Quarter Note) which means that you can put a high degree of "Human Feel" into a piece which, in the past, has always been a serious limitation of sequencers.

Finally we have the Tracks Bar which gives you all track copying and editing facilities you need like copying one track to another or even bouncing 15 tracks down onto one to make room for more complicated sequences. In addition you can invert, erase, cut and pitch transpose any or all of the sixteen tracks available. Tracks can then be quantised to iron out any out of time notes you may have played. Next option really lets you get down to the nitty gritty of your song and modify individual events. Every kind of MIDI event can be altered here; note on, note off, polyphonic pressure, controller changes (modulation, sustain etc), program changes, channel pressure and of course pitch bend. Any or all of the above parameters plus the entire sequence (if you're feeling particularly destructive) can be globally deleted if you feel the need.

MIDI Magic offers everything you need to get started in MIDI sequencing yet will remain useful after a newcomer has got to grips with the whole sequencing process. And for those of you who have been weened off Hardware based sequencers in favour of the floppy variety, rest assured, *MIDI Magic* is just as flexible as all hardware products of this ilk. up to the '1000 mark. Hardened Amiga users will also find it particularly quick to use due to its Workbench-based environment which is used to such good effect. Also a word of praise to Rick Hamouris who has come up with an excellent 80 page manual which, not only takes you through the system in a logical fashion with some nice, humorous touches. In short, not the most comprehensive package to date but excellent value for money. I look forward to seeing more from these guys soon. ***YA***

Title:**MIDI Magic**
Supplier:**Circum Design Inc.**
Price:**TBA**



Software, French Style

A French software company is nurturing programming talent in the depths of the countryside of Brittany and even now is launching the results on to an unsuspecting British public. A Your Amiga team spent *le weekend* ***finding out more.***

■ **Ubi Soft are a newly formed French** software house connected to the largest distributor in France. Their highly sophisticated programs are available in the UK through Electronic Arts. They kindly invited a Your Amiga team onto their home territory to check out some, at the time, unfinished games and to experience the French approach to software development.

Britanny Bound

A quick look at the ferry timetables on Prestel (via Micronet, of course) and a call to our friendly travel agent, got us on our way. Portsmouth to Cherbourg and on to Rennes in Brittany. We could not have imagined the picturesque setting in which the launch was to take place, a chateau surrounded by 700 acres of woodland, circled by a 10-kilometre stone wall and with a stunning view across the Brittany countryside.

This is the Ubi Soft chateau. It is rented to provide the Ubi Soft creative team with the ideal surroundings for the development of new software. Less generously, it could be considered a "prison" for 35 reluctant programmers. The young talent is sought out by Ubisoft at the Paris computer shows. Coders, artists and musicians are all involved. In some ways the setup is reminiscent of Virgin Records' countryside studios to which it sends its star performers. They all seem happy enough!

Inspirational Games

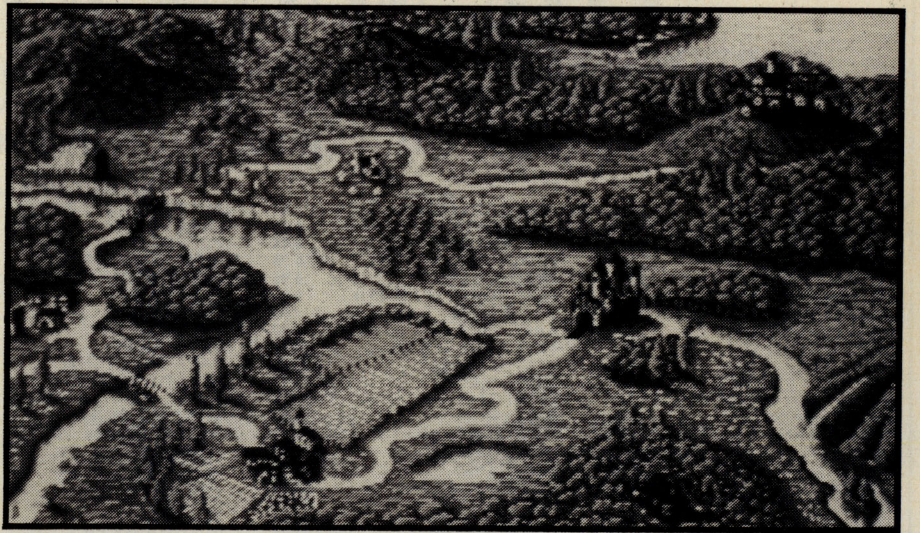
Ubi Soft have most of the chateau while the impoverished baronial family live these days in the outbuildings close-by. An imaginative bunch, these programmers once slept 12 to a room having convinced themselves that the chateau was haunted! It is not difficult to draw comparisons between Ubi Soft and Electonic Arts, both offering a refreshingly creative approach to software production.

The inspiration for a game called *Iron Lord* is plain to see in the surrounding fields. The programmers held court in a room overlooking the chateau grounds, lake, woods, winding roads and French hamlets. Tapestries hung on the walls and an Amiga 500 lay on the four poster bed.

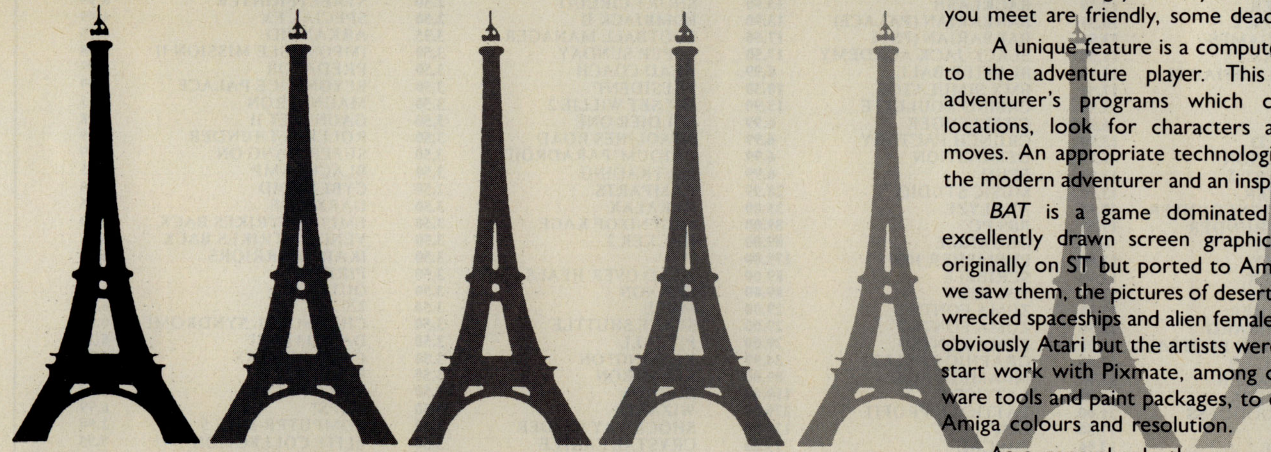
Iron Lord lives up to its beginnings. It's immediately reminiscent of *Defender of the Crown* but with more imaginative characters and more depth of gameplay. The graphics, animation and sound are of the highest quality, especially obvious in a 3D fight sequence not seen before. This sword fight incorporates the sampled sounds of a fight staged by the programmers themselves for the occasion. The action includes archery and arm wrestling. *Iron Lord* is an impressive game on all counts and eagerly looked forward to. It is a two disk game on the Amiga.

Skateball is five aside Rollerball on ice! There are keyboard or joystick control, human versus computer, human versus human and two humans versus computer options. Fast, well animated sprites and a range of "pitches" on which to play are the plus points. Violence is not restricted to the crowd either and you are expected to play dirty. Check out the Olympic ice hockey for the right moves.

Puffy — a fiery dragon-like creature — takes part in his own "saga", an archetypal 2D arcade game made good by noisy samples and humourous action. The Amiga version is full screen and smooth scrolling and superior to the Atari's block scroll and small window on the action. Claude, the programmer, has come up with an original, a female version of the character called Puffin!



Final Command is another port from the Atari ST. It is a graphical space adventure with great promise. The comic strip style presentation and appropriately phrased dialogue in speech bubbles will remind you of your favourite sci-fi comic.

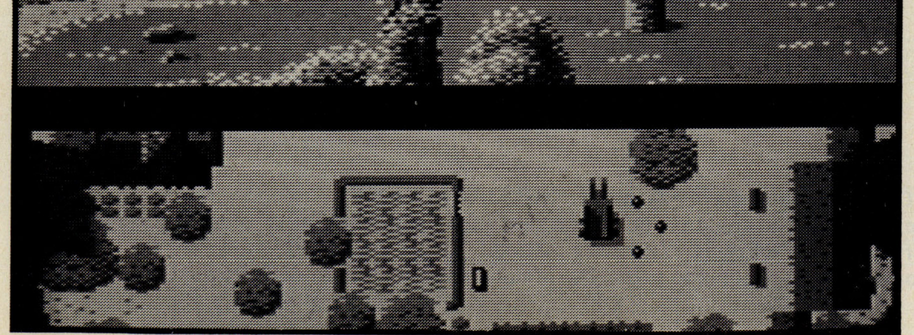


The scenario has you chasing data banks in a space sation threatening to explode at any moment. Once you have mastered your space shuttle, you find out that you are not the only one on the track and that the station and other shuttles are not as lifeless as they once looked. A traitor and enemy of the Confederation will set traps all along your way. Some creatures you meet are friendly, some deadly!

A unique feature is a computer available to the adventure player. This runs the adventurer's programs which can search locations, look for characters and repeat moves. An appropriate technological aid for the modern adventurer and an inspired touch.

BAT is a game dominated by some excellently drawn screen graphics, created originally on ST but ported to Amiga. When we saw them, the pictures of deserted planets, wrecked spaceships and alien females were still obviously Atari but the artists were about to start work with Pixmate, among other software tools and paint packages, to convert to Amiga colours and resolution.

As a general rule the programmers insisted that every advantage will be taken of the Amiga's better technical qualities and, although there was evidence that this was so, we can't judge until the final release versions. The games viewed were developed on Atari ST, some in STOS, and time pressure may take its toll on the conversions. The programmers confided that the Amiga and Atari specialists constantly competed to make their machine look the best!



In *Dracula* the main character swaps between vampire, bat and werewolf to help him in his quest for a talisman now in the possession of Doctor Van Helsing. Objects need to be collected and an army of vampire hunters outwitted and destroyed. The twitter of the dawn chorus is the least welcome sound effect in this game.

Will Ubi Soft emerge to lead the European Amiga software challenge to Electronic Arts and Microprose? *Iron Lord* pictures on this page — is now available with other regular releases in 1989. The new adventures feature arcade interludes and look set to change our expectations of this style of game. Allez Ubi Soft! *YA*

“THE MAIN EVENT”

DEPT (YC) 61 STAFFORD STREET, HANLEY, STOKE-ON-TRENT,
STAFFORDSHIRE ST1 1LW

24 HOUR ANSWER SERVICE. TEL: 0782 281544

AMIGA	
FIRE AND FORGET	17.50
HOT SHOT	£13.90
PLATOON	£17.50
GRYSOR	£17.50
COMBAT SCHOOL	£17.50
GREMLINS HOTS HOTS	13.90
CHUBBY GRISTLE	13.90
GREAT GIANA SISTERS	17.99
BETTER DEAD THAN ALIVE	13.90
BEYOND ICE PALACE	17.50
BIONIC COMMANDO	17.50
BEYOND ZORK	17.50
BUREACRACY	24.99
EBO??STAR	17.50
FIREPOWER	17.50
G L FOOTBALL	17.50
FLINSTONES	13.90
TERRAMEX	13.90
WORLD GAMES	17.50
BAD CAT	17.50
MOUSE TRAP	10.50
THREE STOOGES	21.00
JOE BLADE	6.99
JET	28.50
GARRISON 2	17.50
GARRISON	17.50
GOLDRUNNER	17.50
GRAND SLAM TENNIS	21.00
HUNT RED OCTOBER	17.50
INDOOR SPORTS	17.50
MEAN 18 GOLF	21.00
SIDEWINDER	6.99
STARGLIDER	17.50
STRIP POKER 2	10.50
WESTERN GAMES	13.99
WINTER GAMES	17.50
WINTER OLYMPIAD 88	13.90
WIZBALL	17.50
XENON	13.90
XR 35	6.99
SPIN WORLD	13.90
SPACE STATION	6.99
SPACE QUEST 2	13.90
SKY BLASTER	13.90
ROMANTIC ENCOUNTER	17.50
ROLLING THUNDER	17.50
PROTECTOR	6.99
POWERPLAY	13.90
POWER STYX	13.90
POLICE QUEST	13.90
PINK PANTHER	13.90
PINBALL WIZARD	13.90
P. BEARDSLEY FOOTBALL	13.90
MIND FIGHTER	17.50
LITTLE DRAGON	13.90
LEATHERNECK	13.90
LEATHER GODDESS	21.00
JINKS	17.50
GUNSHOOT	13.90
FLIGHT SIM 2	28.99
FLINSTONES	13.90
FOOTBALL MANAGER 2	12.99
ROAD BLASTERS	17.50
ALIEN SYNDROME	13.90
CORRUPTION	17.50
FIRE STATION	6.99
FINAL MISSION	13.90
EXPLORA	28.00
EMERALD MINE	13.90
EBONSTAR	17.50
DROID 2	13.90
CRYSTAL HAMMER	13.90
BUBBLE BOBBLE	13.90
BUBBLE GHOST	13.90
BAD CAT	17.50
PHANTASM	13.99
CARRIER COMMAND	17.50
ZOOM	13.90
WORLD DARTS	6.99
FIRE 'N' FORGET	13.90
IKARI WARRIORS	17.50
ALIEN	13.90
SREET FIGHTER	17.50
CITY DEFENCE	10.50
SUPER HUEY	13.90
THUNDER BOY	10.50
AAARGH	13.90
BATTLESHIPS	13.90

AMIGA	
SPACE STATION	6.95
WAR ZONE	6.95
FIRE BLASTER	6.95
PROTECTOR	6.95
ENFORCER	6.95
PLAYHOUSE STRIP POKER	9.99
ONE ON ONE	14.90
ARCHON ??	14.90
7 CITIES OF GOLD	14.90
STATION FALL	13.99
MOON MIST	13.99
SUSPECT	13.99
BALLYHOO	13.99
MIND FOREVER VOYAGING	13.99
DEADLINES	13.99
LITTLE COMP PEOPLE	13.99
PORTALL	13.99
MINDHSADOW	13.99
CUT THROATS	13.99
ZORK 2	13.99
ZORK 3	13.99
ENCHANTER	13.99
SORCERER	13.99
INFIDEL	13.99
STAR CROSS	13.99
HACKER	13.99
PUB POOL	6.99
AMIGES	10.50
ANNALS OF ROME	13.90
ARENA	21.00
ART OF CHESS	17.45
BALANCE OF POWER	21.00
BACKLASH	13.90
BARBARIAN (PALACE)	13.90
BARBARIAN (PSY)	17.50
BLACK JACK ACADEMY	17.50
BLASTER BALL	6.99
BMX SIMULATOR	10.50
CASINO ROULETTE	13.90
CHALLENGER	6.99
CRUNCH FACTORY	6.99
DEMOLITION	6.99
FEUD	6.99
MUSIC STUDIO	28.99
ANALYZE	35.00
BBS PC	89.00
ORGANIZE	89.00
PUBLISHER 1000	179.00
SCRIBBLE	89.00
T.V. TEXT	89.00
ZUMA FONTS 1	29.00
ZUMA FONTS 2	29.00
ZUMA FONTS 3	29.00
GRAPHIC TABLET	34.99
MI-AMIGA FILE	89.00
LATTICE C	119.00
LATTICE C PROFIT	279.00
LISP	139.00
PASCAL	79.00
PRO SOUND DESIGNER	69.99
MUSIC STUDIO	29.99
FOOTBALL DIRECTOR 2	11.00
FOOTBALL MANAGER 2	10.50
FERNAND?? MUST DIE	17.45
ROCKET RANGER	20.00
POOL OF RADIANCE	17.45
MENICE	13.90
NETHERWORLD	13.90
CARRIER COMMAND	17.45
FEDERATION OF FREE TRADERS	19.49
PAC MANIA	13.90
WHERE TIME STOOD STILL	11.95
BIONIC COMMANDOS	17.45
STAR GLIDER	17.45

COMMODORE 64	
JACK THE NIPPER	2.50
LAST MISSION	2.50
DECEPTOR	2.50
FIELDS OF FIRE	2.50
KNIGHT GAMES II	2.50
OLTAPOLIS	2.50
QUARTET	2.50
SUPER SPRINT	2.50
FIRE TRAP	2.50
BACK TO FUTURE	2.50
SPORTING CLASSICS	2.50
ACE	2.50
MYSTERY OF NILE	2.50
ON THE TILES	2.50
SIDEWIZE	2.50
SCARY MONSTERS	2.50
SOLOMON'S KEY	2.50
RYGAR	2.50
TRANTOR	2.50
BLADE RUNNER	2.50
CAPTAIN AMERICA	2.50
FIVE STAR GAMES 2	2.50
STAR PAWS	2.50
MARBLE MADNESS	2.50
ARFIC	2.50
LEADER BOARD	2.50
SCOOBY DOO	2.50
PING PONG	2.50
TAG TEAM WRESTLING	2.50
GAUNTLET	2.50
GAUNTLET DEEPER DUNG	2.50
DONKEY KONG	2.50
BREAKTHRU	2.50
SHORT CIRCUIT	2.50
BOMBJACK II	2.50
FOOTBALL MANAGER	3.95
SUPER SUNDAY	3.50
HEAD COACH	3.50
PRESIDENT	3.50
JET SET WILLIE 2	3.50
SOLDIER ONE	3.50
SHAOLINES ROAD	3.50
URIDUM/PARADROID	3.50
PSI TRADING	3.50
RAMPARTS	2.50
PARALAX	3.50
LEGEND OF KAGE	3.50
HACKER 2	3.50
V	3.50
HEAD OVER HEALS	3.50
ZAXXON	3.50
MULE	3.50
SPACE SHUTTLE	3.50
PITFALL	3.50
KENSINGTON	3.50
ROBOTRON	3.50
ZORRO	3.50
WIZARD	3.50
SHOCKWAY RAIDER	3.50
CRYSTAL CASTLE	3.50
JEWELS DARKNESS	3.50
BOULDERDASH CON KIT	3.50
ROAD RUNNER	3.50
RENEGADE	3.50
STAR GAMES II	3.50
FLIGHTPATH 737	3.50
THUNDERCATS	3.50
BUBBLE BOBBLE	3.50
BUGGY BOY	4.75
SPACE MARRIOR	4.75
ADICTIBALL	4.95
FRANKENSTAIN	2.50
I ALIEN	2.50
VENGEANCE	2.50
DISCOVERY	2.50
JET BOYS	2.50
TIME FIGHTER	2.50
THUNDER CROSS	2.50
MANDROID	2.50
OINK	2.50
WOLF MAN	2.50
STRATTON	2.50
TRAXXION	2.50
BOOK OF DEAD	2.50
LIFE FORCE	2.50
BIG TROUBLE IN CHINA	2.50
SUPER STAR PING PONG	2.50
PARADROID/URIDIUM	2.50
THE YOUNG ONES	2.50

COMMODORE 64	
PINK PANTHER	6.99
SKATE CRAZY	6.99
HERCULES GREMLIN	6.99
BLOOD BROTHERS	6.99
GREMLINS HOTS HOTS	6.99
G.I. HERO	6.99
LAST NINJA II	8.99
BIONIC COMMANDO	6.99
EVERY SECOND COUNTS	5.59
GREAT GIANA SISTERS	6.99
ROAD BLASTERS	6.99
DREAM WARRIORS	6.99
MARRAUDER	6.99
MATCHDAY II	6.25
WIZARD WARS	6.99
KIDSPLAY COMPILATION	6.99
SALAMANDER	6.25
MICKEY MOUSE	6.99
OBSOLATOR	6.99
ALIEN SYNDROME	6.99
HOPPING MAD	6.99
THE VINDICATOR	6.25
EDDIE EDWARDS SUPERSKI	6.99
PLATOON	6.99
RASTAN	6.25
THUNDERCATS	3.95
COMBAT SCHOOL	6.25
GAYSTOR	6.29
D.T. OLYMPIC CHALLENGE	6.95
TARGET RENEGADE	6.25
KONAMI ARCADE COLLECTION	6.99
STREETFIGHTER	6.99
SPECIAL FX	6.99
ARKANOID	6.25
IMPOSSIBLE MISSION II	6.98
PREDATOR	6.99
BEYOND ICE PALACE	6.99
MAGNETRON	6.25
GAUNTLET II	6.98
ROLLING THUNDER	6.99
SUPER HANG ON	6.99
BLACKLAMP	6.25
CYBERNOID	6.99
DARKSIDE	6.95
EMPIRE STRIKES BACK	6.99
VENOM STRIKES BACK	6.99
IKARI WARRIORS	6.95
FIREFLY	6.25
OUTRUN	6.99
720	6.99
CHERNOBYL SYNDROME	6.90
DAN DARE II	6.25
CARD SHARKS	6.95
MINI PUTT	6.95
NORTH STAR	6.99
OOPS!	6.99
COMPUTER HITS 5	2.50
ELITE COLLECTION	9.95
GOLDEN OLDIES	2.50
KNIGHT RIDER	6.25
DRAGONS LAIR	6.25
10 GAMES BY GREMLIN	6.95
FERNANDEZ MUST DIE	6.90
NATO ASSAULT COD	9.00
TRAZ	6.95
CRAZY CARS	6.95
YETI	6.95
CROSSWIZ	6.25
CHARLIE CHAPLIN	6.99
NOW GAMES 3	2.50
VIXEN	6.95
TAITO COIN OPS	8.90
THE FLINSTONES	6.99
THE ULTIMATE PACK	5.90
ACTION PACK II	2.99
GUTZ	6.2?
CHUBBY GAISTLE	6.98
INTENSITY	6.90
ECHELON	8.99
BUTCHER HILL	6.50
SHACKLED	6.99
BUGGY BOY	3.9?
MASK	2.5?
TYPHOON	6.99
BARDS TALE	6.95
TRACKSUIT MANAGER	6.99
HERD OF THE LANCE	6.95
MERCENARY	6.95
TECHNO COP	6.95

Please cross PO/Cheques and make payable to: THE MAIN EVENT.
To help maintain our low prices we would be obliged if you could include a first class stamp with your order, although this is NOT compulsory. Free Post & Packaging. Overseas please add 75p per tape.
We apologise for any alteration or omissions since going to press. The Main Event is a division of Software City.

AN AMIGA NEW YEAR WITH G.T.S.

£4.99 SPECIALS

THE JUMPDISK AUTUMN SPECIAL — Main features are: Amigantt — A project Management Scheeduler/Atlantic City Blackjack — includes Multiple Players, Strategy, Casino Specific Rules/Addicon — Assign Icons to Files/ Catalogue — all the past issues of Jumpdisk.

New! **Shareware Extravaganza** — put together specially for Jumpdisk by Chet Solasce this disk contains 26 programs of the best new software Solace found on the American bulletin boards during an intensive October foray. The disk contains about 1.5 meg of material stored in a highly compressed self-booting format. It is menu driven and ready to run with 8 colour Icons, Ansi text, animation and full documentation. Recommended!

THE JUMPDISK 'DOUBLE'

THE JANUARY ISSUE OF **JUMPDISK** WITH EITHER ONE OF THE **£4.99 SPECIALS** FOR ONLY **£12.50 inc.**

£19.99 MEGA PACKS

NEW! MEGA GAMES PACK — VOL. 1

Yes, at last it's available! *Seven* disks packed full of the best Public Domain, Shareware and our own games programs. All disks autoboot and all programs run on 512K.
Battlemech, Bullrun, Super-Breakout, Grravattack/Cos (Wheel of Fortune), 3D Othello, Stoneage (Rockford Clone), Q-Bert, Crystal-Vision/Hack: (The Latest 3D Version of the Famous Dungeons Type Game) /Monopoly (updated version), Deluxburger, Asteroids, Mercenary Simulator, W/B Puzzle/Escape From Jovi, Amoeba Invaders, Missile Command, IFF2PCS/Battleship, Descender, Pacman '87, Tiles/Revenger of Doom (not PD or Shareware) new and original Space Arcade Type Game.

NEW! MEGA DEMO PACK — VOL. II

7 disks packed with some of the greatest graphic and sound demos. We've even included an 'Intro Maker'. Now it's possible to make up your own intro screens using IFF sounds and pics. Messages may be displayed vertically and horizontally.

STILL AVAILABLE! MEGA DEMO PACK — VOL. I

The original *7 disks* pack of spectacular visual and sound demos. These should sell an Amiga to any Atari owner! Order in January and receive an extra demo FREE!

STILL AVAILABLE! THE PD STARTER PACK

This best selling pack now includes *9 disks*! The Goldfish (3 disks), Best of Amicus (4 disks) and our own Double Library Disk Pack. All for little more than the price of 9 Blank Sony Disks! There's no better way to find out about Amiga PD.

NEW! THE 'ALTERNATIVE' DEMO PACK

8 disks featuring a collection of the very best demos on PD. Includes old favourites such as 'Robot City' and many entries to the Badge Killer Demo Contest. We've searched over 600 PD disks in order to put this collection together! Some demos need 1 mag.

PUBLIC DOMAIN LATEST NEWS

JUST ARRIVED FROM THE U.S.A — F.A.U.G. DISKS 76 THROUGH 85
THESE ARE £3.00 PER DISK OR ANY MAY BE PURCHASED AS A PACK OF 8 FOR ONLY £19.99

FAUG 76 DOSK wik (set up a Ram disk about 3 times faster than DOS), LDebug (Working demo from Lattice), MenuLink (Run 'Menu using' programs while using CLI), Garrote (Edit 'Garrison' Dungeon Maps), Moria (Dungeon Adventure Game), Postcard (Apples hypercard?).

FAUG 77 DMouse (Mouse — Intuition enhancer), Orbit3D (Space Game), MyBatch (Run batch files without the execute command), VScreen (Make screens larger than the display area of monitor), C-Light (Working demo of Ray Tracer/Object Editor).

FAUG 78 DiskCheck (Checks disks for errors), MG (a PD Emacs Style Editor V2a), MachII (Mouse accelerator+Hot Keys, etc.), WBLace (Increases resolution of W/B screen), DiskX (Sector-based Disk Editor V2.0), Storage (Displays disks storage capacities), QLens (Magnifies portion of any screen V2.02).

FAUG 79 VLabel (Label printing program V1.2), KickFont (Change font on Kickstart Disk), RunBack 2.0 (Runbackground type program), Fix-I8 (Installs Interceptor on Hard Disk), D50-Editor (Working demo for Rolands D-50 Keyboard).

FAUG 80 Crunch (File compression program), VirusX (V1.5 of Icon-driven virus checker), Sid (system Information Display V1.00), ASCTable (Pop-up ASC table), Undelete (Recovers deleted files), Tiles (Shanghai type game).

FAUG 81 *Access 1.4 (Telecommunication program), ViewBoot (Examine Disk Boot Block/Memory for Virus checking), VirusX (V1.7), Wheel (Wheel of Fortune style game with speech).*

FAUG 82 Umaster (CLI replacement program), Crunch2 (File cruncher with documentation), FastDisk 'optimizer' V1.5), Profile 1.1 (Profiler for MANXC), DMouse9 (V1.09 of Popular Input Handler).

FAUG 83 DME (Programmers Editor), SupLib (General G Support Library), AmyCal (Console Calendar), IconLab, ExtractFont (Creates C Source File for an Amiga Font), VGad (Gadget Making Tool).

FAUG 84 *WC (Counts words, etc), TWin (Text Server Window), NoBorder (Provides a true 80×24 display area), VirusX (V2.0), Warp 1.2 (Disk Tracker/Compressor), TRAvail (Updates Memory usage), Scheme (Dialect of LISP), QView (File viewer), ClipIt (Take 'Clips' of graphics from any screen), GWar (V2.0 of Game 'Gravity Wars').*

FAUG 85 HandyIcons (Extension to W/B Menu Strip), Lookfor (File Searcher), QuadPrint (Screen Dump for small posters), Shakespeare (Fully functional demo of V1.1 D.T.P.+20 minus self-running demo).

FOR MORE INFORMATION PLEASE REFER TO V1.4 OF OUR LIBRARY DISKS. PRICE £4.99 (INC. £5.00 VOUCHER) AND DON'T FORGET YOUR FREE UPDATE! NO MEMBERSHIP REQUIRED.
REMEMBER YOU READ ABOUT THE LATEST P.D. RELEASES HERE FIRST!

WE ARE GEORGE THOMPSON SERVICES · OUR ADDRESS IS: DIPPEN, BRODICK, ARRAN, SCOTLAND, KA27 8RN.
OUR TELEPHONE NUMBER IS (077082) 234.

Please send a cheque (with your bank card No./expiry date on the reverse) or P.O. for the appropriate amount and we will despatch your order immediately

AmigaVision

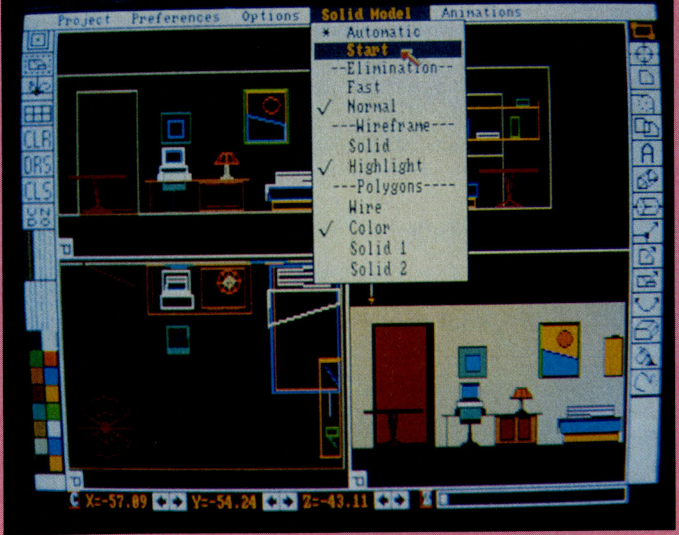
A bespectacled Ann Owen looks at Forms in Flight and Design 3D

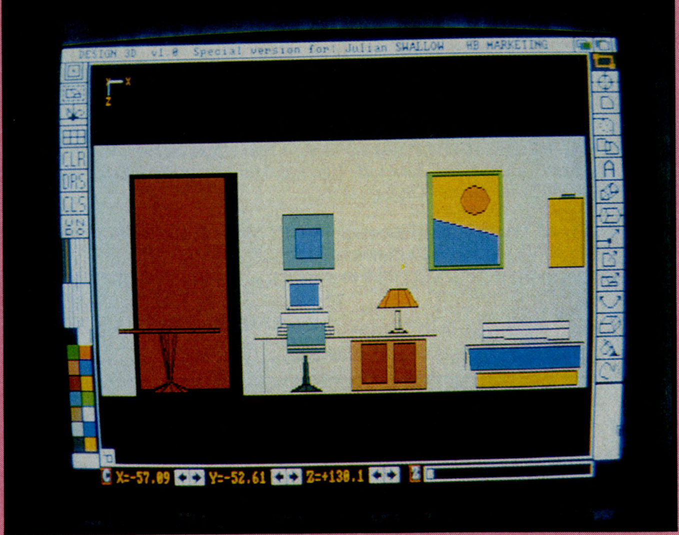
■ ***Design 3D* is the latest release from** Canada's Gold Disk, a software house with an impeccable pedigree on the Amiga, creators of *Comic Setter* and *Professional Page*. *Design 3D* is launched into an area so far dominated by Aegis with *Sculpt 3D* and *Modeller* programs. *Forms in Flight* is by Micro Magic who hail from Palo Alto in California. It has been available in the UK for about a year. Both programs require one megabyte of memory to run.

Design 3D automatically configures itself for NTSC (America) or PAL (Europe) operation. It can work in low or high resolution and in two, four or 16 colours.·

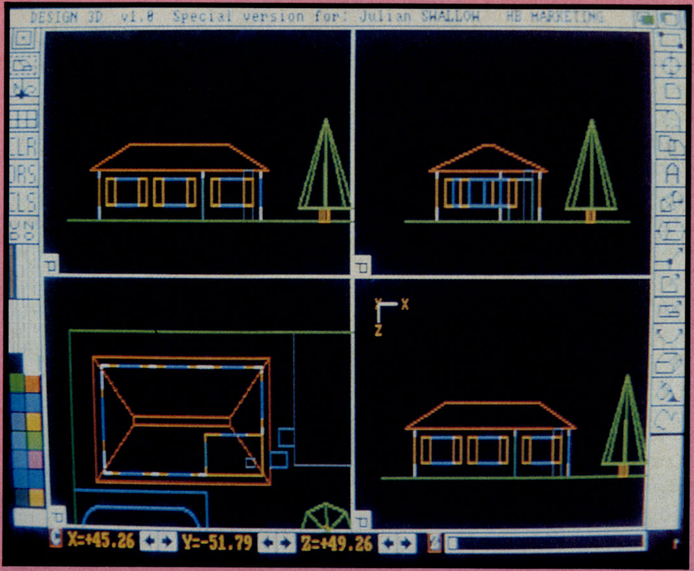
For image creation, *Forms in Flight* supports 640x400 pixels (interlaced) and 640x200 pixels with 2,4,8 or 16 colours. For sequence playback it supports these two screen modes plus 320x400 (interlaced) and 320x200, the latter two supporting 32 colours.

As always in the interlaced modes there





DESIGN 3D v1.0 Special version for: Julian SWALLOW HB MARKETING
X
Z
CLR
DRS
CLS
UNDO
A
X:-57.09
Y:-52.61
Z:+130.1

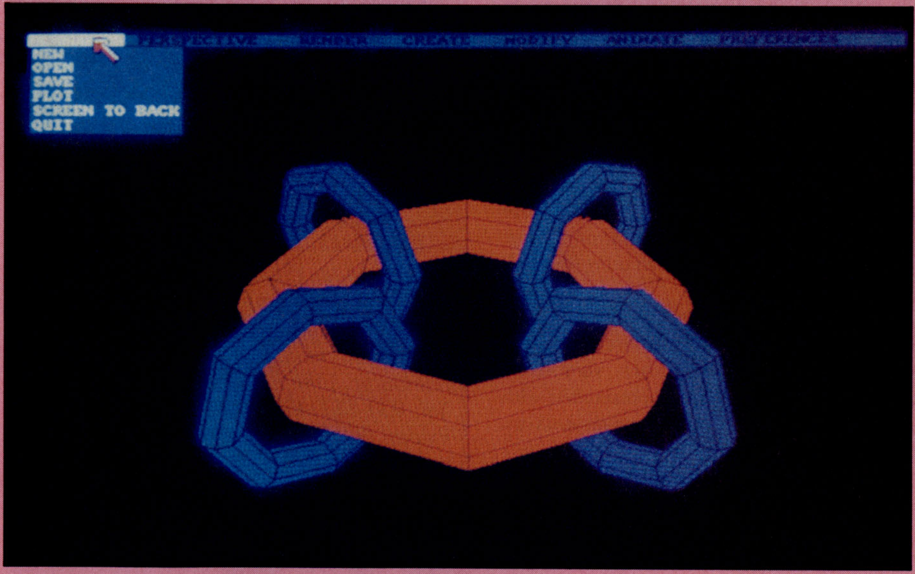

DESIGN 3D v1.0 Special version for: Julian SHALLOW HB MARKETING
X=+45.26
Y=-51.79
Z=+49.26

is considerable flicker in the display. Both *Design 3D* and FIF work across three disks for program, data and animation so disk swapping is required on a single drive system. *Design 3D* saves drawings in its own internal format and as IFF for transfer into other paint or animation programs. Typically you could use an object generated by *Design 3D* to visualise a drawing in three dimensions alongside a plan view generated in a CAD package. If you are interested in the animation element of the program for desktop video then you will appreciate its support of the ANIM format — the equivalent of IFF for animations.

Printing from *Design 3D* is via Preferences and HPGL (Hewlett Packard Graphics Language) plotters are supported. *Forms in Flight* also plots via HPGL.

Third Dimension

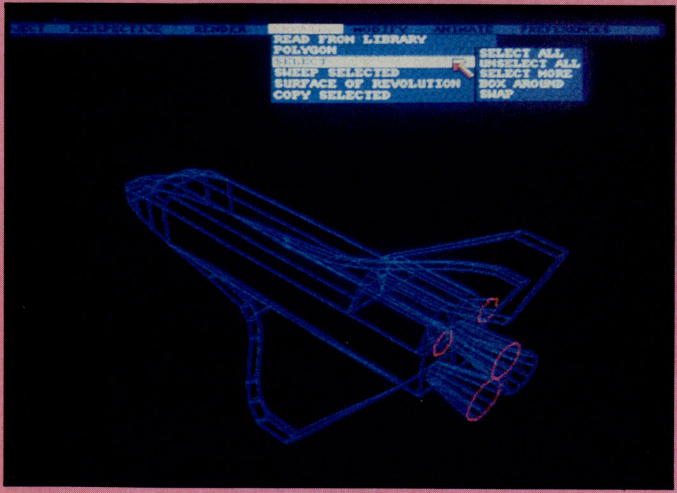
■ **To borrow the analogy from Micro** Magic, the difference between 2D drawing and 3D drawing is the difference between a drawing on a sheet of paper and a sculpture. The sculpture can be viewed from any direction and the image changes for each new position the viewer takes up, while the drawing on paper, which can be made to appear three dimensional by an artist, only changes size or shifts up/down or left/right with a change in the viewer's position.

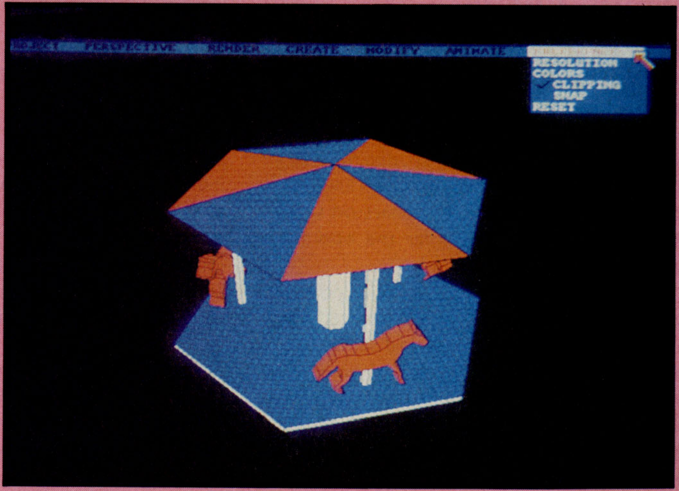


Design 3D Detail

■ **Four windows provide views onto the** object being designed, front, side, top and perspective. 2D and polygon creation tools are used to create an object which can then be displayed from different points of view, with up to four light sources and as a solid model. You can set up a grid, line styles and colour palette. The current drawing can be erased completely, erased but stored and retrieved to the screen. There's also an undo feature for cancelling the last polygon in the list. A "nodes" tool connects two close points on an object, useful for saving memory and speeding up rendering.

Design 3D has its own preferences menu to set up the units of measurement, the palette, the scale, the distance between points of the X,Y,Z grid, the number of points used to create the ellipse polygon (eight would provide an octagon, 90 a "circle"), the number





of sectors used by the SPIN tool, the angle of view of the object representation in the perspective window in degrees (this can also be changed in realtime) and the X,Y,Z positions of the four possible light sources, called lamps as in ray tracing.

The drawing tools are rectangle, ellipse, user-defined polygon (from triangle to "circle", arc, clone (polygon duplication) and text. Grouped under "volume creation tools" are CONNECT, which automatically connects polygons containing the same number of points and SPIN, which creates an object around a vertical axis. Manipulation tools are POINT SELECT, POLYGON SELECT and OBJECT SELECT.

There are various facilities to help achieve accurate drawings. CROSS extends the cross-hairs to the full width and height of the work window, DIMENSIONS displays co-ordinate values relative to the last entered point. SECTION turns a newly entered polygon into a section of an object in solid view mode. 3D COPY copies the perspective window to another.

When modelling solids you can choose how to fill the polygons, in a chosen colour, without shading, with shading using palette colours — made easier by a gradation gadget in the palette — or with shading as dot patterns. You can choose to have the wire-frames visible or invisible. A 3D font file is supplied with the program and a font editor allows you to select a letter from the alphabet and change it to your specification. Fonts can be saved and reloaded at any time.

Because Your Amiga was sent a pre-production version of *Design 3D* I got an interesting insight into the work that the publisher has put into the manual. The program author is Arnaud Ribadeau Dumas but the famous literary name means nothing. The original documentation was full of jargon terms and assumptions. Any credit for the user-friendly set of instructions and uncomplicated tutorials goes to the Gold Disk manuals department.

Taking Flight

■ ***Forms in Flight* works in one screen** for object creation with a variety of requestor boxes providing access to parameters and managing the fairly complex animation facility. When an operation, such as loading a large object from disk or removing hidden lines takes some time then a counter appears on screen giving you an idea of how far the operation has progressed.

You can create various types of polygon, turn a 2D object into a 3D object, copy, delete, move and resize selected surfaces and move nodes. A polygon shape can also be read in from a library of such objects, an important time saver if you are serious about creating animations. There are some objects already on disk.

Entering coordinates involves typing them into a calculator type screen, clicking on an existing point on the screen or drawing lines with the mouse in 2 or 3D mode. When in 3D mode, the fixed coordinate can be either the X,Y or Z axis.

A preferences menu provides the means of switching screen resolution, dealing with colours and resetting the system, as well as switching on and off clipping and snap modes.

You can change the "view" by panning, moving the point at which you are looking, changing the perspective, and magnification. Hidden lines can be removed, shading and light sources added, selected surfaces ghosted and surfaces left hidden from view.

A red/blue stereo image can be created (for use with red/blue stereo glasses which are not supplied!) but not animated. We weren't sure about the purpose of this feature but had fun with it anyway.

An animation is defined by a tree hierarchy of objects along with their own frame blocks which define their individual movement. Various types of motion are allowed, straight line in any direction, straight line with changes in direction at any time, rotation around any point and any combination of these. Finally the camera (your view) can be moved around too.

The frames are calculated and automatically saved to disk in a compressed format. At 15 frames per second, a 30 second animation requires the processing of 450 images. A new frame can take over a minute to calculate so unattended operation is vital! This can cause a problem when using floppies, not hard disk, as storage.

Frames are decompressed and played back in realtime if required. Both recording and preparation for playback are very time consuming but the results are outstanding with a mixture of quite complex movements on screen.

The *Forms in Flight* manual is a ring binder with over 100 sturdy single-sided A5 sheets. There's a quick introduction and example plus lots of detail on each facility. No index unfortunately.

Comparisons

■ **Both programs are easy to use** perhaps *Design 3D* slightly more so with its icon tools, although I had to keep referring to the manual to recall what functions they represented. *Design 3D* gets you good results quickly, the realtime animation is fun but what sets it apart from other three dimensional modellers are the helpful tools.

As the title subtly suggests, *Forms in Flight* is biased towards the creation of animated objects. It has three means of output, a screen, a HPGL plotter or a video recorder. You can record animations in real time with any recorder while a flying-erase head camera will allow you to capture the sequence a frame at a time.

If you want printer output, intending to get into video later, you can confidently choose *Design 3D*. If you are into video animation now then both programs are worth a long look.

YA

Factfile

■ ***Design 3D*, published by Gold Disk, is available in the UK from HB Marketing.**

■ ***Forms in Flight*, by Magic Micro, is available from:**
Amiga Centre Scotland.
4 Hart Street Lane
Edinburgh EH1 3RN
Tel: **031-557 4242**
Prices: **V1.0 — £69.00**
V2.0 — £79.00

Amiga.lib rules OK!

Margaret Stanger continues her survey of the Amiga's library routines

■ **The Intuition library is featured this** month. It enables you to open a custom screen with a few windows. Intuition keeps track of the properties of a screen by creating a structure for it, and stashing this away in memory. The variables in a structure are always in a standard order, the offset of the Left Edge (say) from the base screen address is always the same.

To save us hassle, there is a disk file available with details of all the Intuition structures, together with the numerical value of any flags used. This file, *include/intuition/intuition.h* can be included in the C source code by a statement at the beginning of the code. There is a small file called *include/exec/types.h* that is included first to interpret the structures in the other include files. It tells us that a byte increases the offset by one byte, and a word by two bytes and so on. Some of the include files call in other include files, and there are many cross references.

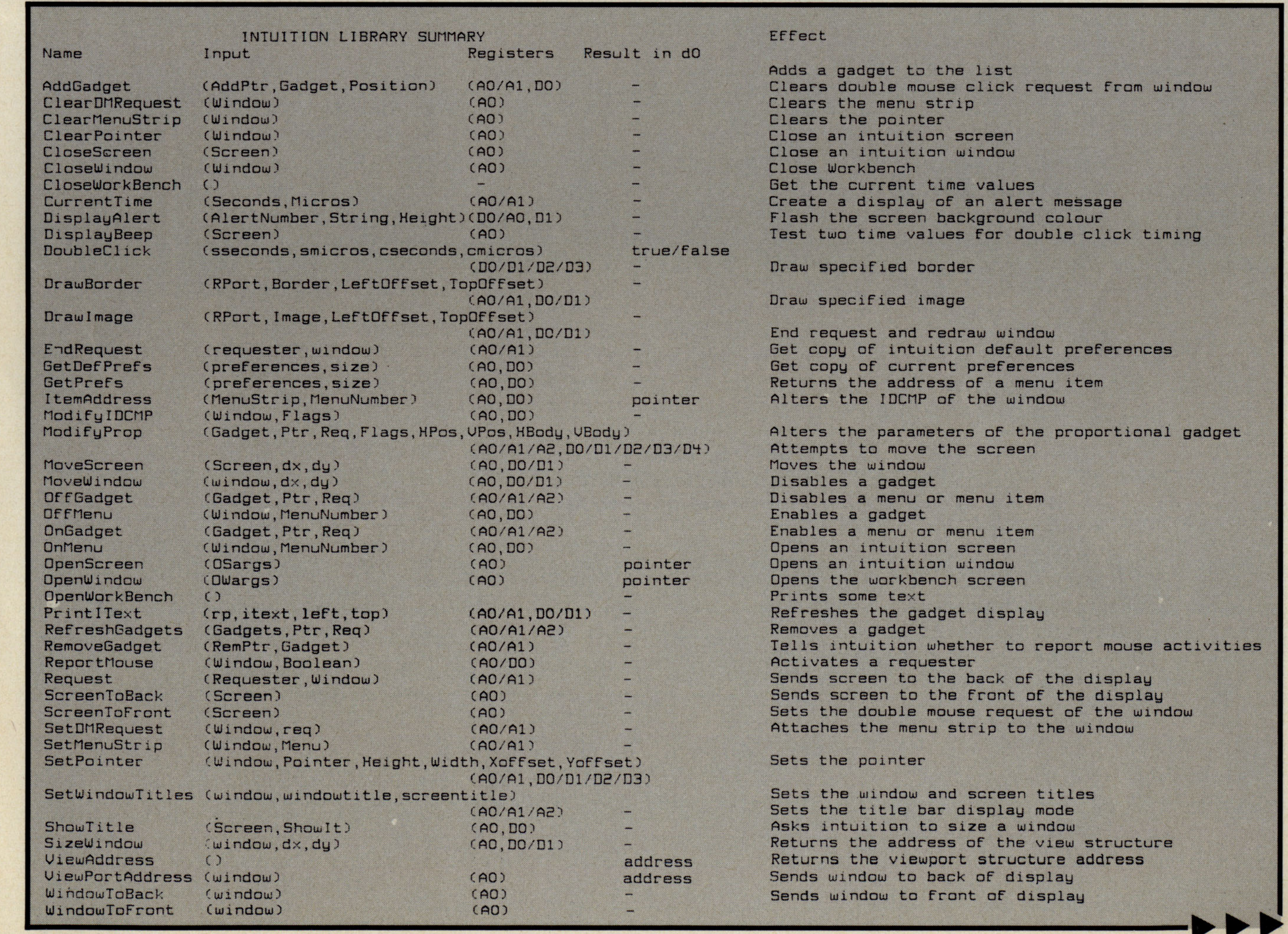
There is a similar arrangement for assembler with the *intuition/intuition.ifile*. Unfortunately my assembler used on.an A500 did not have enough workspace for the necessary include/intuition file, so I picked out the structures and flags I needed and added them to the source code.(I edited the file, marked blocks, wrote them to disk and joined them to the source code rather than typing them in the hard way.)

Intuition will give us the base address of a custom screen if we send it the base address of a set of values in a new screen structure. I always feel as if I have to fill in an application form before I am allowed to have a screen. There is a similar arrangement of filling in a new window structure for each window in the screen, I used the same application form for each of my three windows and changed the details slightly afterwards.

The program opens a custom screen, and the first two windows. The windowflags are set for most of the system gadgets, and Intuition takes care of their activities. The IDCMP CLOSEWINDOW flag is deactivated on the second window, the title is changed and it is slowly moved down. The third window is opened, modified and moved in a similar way. These windows can all be dragged, sized, moved back and front and made active. The program only exits when the first window is closed using the close gadget.

The program uses a few of the routines from the Intuition library, but there are many more available. I have included a list of all the 1.1 library Intuition routines that were documented. For each one I have listed the input, input registers, output (where applicable) and a short description of the effect of the routine. The source code is provided in C and Assembler; the C source code file was compiled and linked using Lattice C version 3.03 and the assembly source code file was assembled and linked using the Metacomco assembler version 10.178.

The next part of this series will feature setting up a display using routines from the graphics library, and will include a library summary. *YA*



```
                    INTUITION LIBRARY SUMMARY                              Effect
Name            Input                       Registers  Result in d0
                                                                           Adds a gadget to the list
AddGadget       (AddPtr,Gadget,Position)    (A0/A1,D0)      -              Clears double mouse click request from window
ClearDMRequest  (Window)                    (A0)            -              Clears the menu strip
ClearMenuStrip  (Window)                    (A0)            -              Clears the pointer
ClearPointer    (Window)                    (A0)            -              Close an intuition screen
CloseScreen     (Screen)                    (A0)            -              Close an intuition window
CloseWindow     (Window)                    (A0)            -              Close Workbench
CloseWorkBench  ()                           -              -              Get the current time values
CurrentTime     (Seconds,Micros)            (A0/A1)         -              Create a display of an alert message
DisplayAlert    (AlertNumber,String,Height)(D0/A0,D1)       -              Flash the screen background colour
DisplayBeep     (Screen)                    (A0)            -              Test two time values for double click timing
DoubleClick     (sseconds,smicros,cseconds,cmicros)         true/false
                                            (D0/D1/D2/D3)   -              Draw specified border
DrawBorder      (RPort,Border,LeftOffset,TopOffset)         -
                                            (A0/A1,D0/D1)                  Draw specified image
DrawImage       (RPort,Image,LeftOffset,TopOffset)          -
                                            (A0/A1,DC/D1)                  End request and redraw window
EndRequest      (requester,window)          (A0/A1)         -              Get copy of intuition default preferences
GetDefPrefs     (preferences,size)          (A0,D0)         -              Get copy of current preferences
GetPrefs        (preferences,size)          (A0,D0)         -              Returns the address of a menu item
ItemAddress     (MenuStrip,MenuNumber)      (A0,D0)         pointer        Alters the IDCMP of the window
ModifyIDCMP     (Window,Flags)              (A0,D0)         -
ModifyProp      (Gadget,Ptr,Req,Flags,HPos,VPos,HBody,VBody)               Alters the parameters of the proportional gadget
                                            (A0/A1/A2,D0/D1/D2/D3/D4)      Attempts to move the screen
MoveScreen      (Screen,dx,dy)              (A0,D0/D1)      -              Moves the window
MoveWindow      (window,dx,dy)              (A0,D0/D1)      -              Disables a gadget
OffGadget       (Gadget,Ptr,Req)            (A0/A1/A2)      -              Disables a menu or menu item
OffMenu         (Window,MenuNumber)         (A0,D0)         -              Enables a gadget
OnGadget        (Gadget,Ptr,Req)            (A0/A1/A2)      -              Enables a menu or menu item
OnMenu          (Window,MenuNumber)         (A0,D0)         -              Opens an intuition screen
OpenScreen      (OSargs)                    (A0)            pointer        Opens an intuition window
OpenWindow      (OWargs)                    (A0)            pointer        Opens the workbench screen
OpenWorkBench   ()                                          -              Prints some text
PrintIText      (rp,itext,left,top)         (A0/A1,D0/D1)   -              Refreshes the gadget display
RefreshGadgets  (Gadgets,Ptr,Req)           (A0/A1/A2)      -              Removes a gadget
RemoveGadget    (RemPtr,Gadget)             (A0/A1)         -              Tells intuition whether to report mouse activities
ReportMouse     (Window,Boolean)            (A0/D0)         -              Activates a requester
Request         (Requester,Window)          (A0/A1)         -              Sends screen to the back of the display
ScreenToBack    (Screen)                    (A0)            -              Sends screen to the front of the display
ScreenToFront   (Screen)                    (A0)            -              Sets the double mouse request of the window
SetDMRequest    (Window,req)                (A0/A1)         -              Attaches the menu strip to the window
SetMenuStrip    (Window,Menu)               (A0/A1)         -
SetPointer      (Window,Pointer,Height,Width,Xoffset,Yoffset)              Sets the pointer
                                            (A0/A1,D0/D1/D2/D3)
SetWindowTitles (window,windowtitle,screentitle)                           Sets the window and screen titles
                                            (A0/A1/A2)      -              Sets the title bar display mode
ShowTitle       (Screen,ShowIt)             (A0,D0)         -              Asks intuition to size a window
SizeWindow      (window,dx,dy)              (A0,D0/D1)      -              Returns the address of the view structure
ViewAddress     ()                                          address        Returns the viewport structure address
ViewPortAddress (window)                    (A0)            address        Sends window to back of display
WindowToBack    (window)                    (A0)            -              Sends window to front of display
WindowToFront   (window)                    (A0)            -
```

▶▶▶

WindowLimits	(window,minwidth,minheight,maxwidth,maxheight)	(A0,D0/D1/D2/D3)		Sets the maximum and minimum size of the window
SetPrefs	(preferences,size,flag)	(A0,D0/D1)	-	Sets the preferences
IntuiTextLength	(itext)	(A0)	length	Returns the length of the text in pixels
WBenchToBack	()		-	Sends the workbench screen to the back
WBenchToFront	()		-	Sends the workbench screen to the front
AutoRequest	(Window,Body,PText,NText,PFlag,NFlag,W,H)	(A0,A1,A2,A3,D0,D1,D2,D3)	true/false	Builds and gets a response from a requester
BeginRefresh(Window)		(A0)	-	Sets the window for optimised refreshing
BuildSysRequest	(Window,Body,PosText,NegText,Flags,W,H)	(A0,A1,A2,A3,D0,D1,D2)	-	Builds and displays a system requester
EndRefresh	(Window,Complete)	(A0,D0)	-	Resets the layer and window
FreeSysRequest	(Window)	(A0)	-	Free the memory used to call a system request
MakeScreen	(Screen)	(A0)	-	Updates the screens viewport
RemakeDisplay	()		-	Remakes the display using any new values
RethinkDisplay	()		-	Updates the viewport and remakes the display
AllocRemember	(RememberKey,Size,Flags)	(A0,D0,D1)	-	Uses intuition to allocate memory
FreeRemember	(RememberKey,ReallyForget)	(A0,D0)	-	Frees this memory

```
;assembly source code

 INCLUDE  "exec/types.i"     ;these files are needed to interpret
 INCLUDE  "exec/exec.i"      ;the intuition structures


 XREF  _AbsExecBase         ;routines from exec library
 XREF  _LVOOpenLibrary
 XREF  _LVOCloseLibrary
 XREF  _LVOWait

 XREF  _LVOOpenWindow       ;routines from intuition library
 XREF  _LVOCloseWindow
 XREF  _LVOOpenScreen
 XREF  _LVOCloseScreen
 XREF  _LVOSetWindowTitles
 XREF  _LVOMoveWindow
 XREF  _LVOModifyIDCMP

main:
 lea       IntuitionName,a1      ;open the intuition library
 clr.l     d0
 move.l    _AbsExecBase,a6
 jsr        _LVOOpenLibrary(a6)
 movem.l   d0,intuitionbase      ;and store the pointer
 beq       abort                 ;exit if the library does not open

 lea       TheNewScreen,a0       ;pointer to new screen information
 move.l    intuitionbase,a6
 jsr       _LVOOpenScreen(a6)    ;open a custom screen
 movem.l   d0,customscreen       ;and store the pointer
 beq       abort                 ;exit if the screen does not open

 lea       TheNewWindow,a0       ;pointer to new window information
 move.l    d0,nw_Screen(a0)      ;attach the window to the custom screen
 jsr        _LVOOpenWindow(a6)   ;open the window
 move.l    d0,thiswindow         ;and store the pointer
 beq       abort                 ;exit if the window does not open

 lea       TheNewWindow,a0       ;use the same information
 move.b    #3,nw_BlockPen(a0)    ;change it slightly
 jsr        _LVOOpenWindow(a6)   ;and open a second window
 move.l    d0,thatwindow
 beq       abort
 move.l    thatwindow,a0         ;change the windowtitle
 lea       screentitle,a2
 lea       thatwindowtitle,a1
 jsr        _LVOSetWindowTitles(a6)
 move.l    thatwindow,a0         ;change the IDCMP flags so that the window
 clr.l     d0                    ;does not close when the close gadget is clicked
 jsr        _LVOModifyIDCMP(a6)

 move.l    #50,d6
window2loop:                     ;slowly move thatwindow down
 move.l    thatwindow,a0
 move.l    #0,d0
 move.l    #1,d1
 jsr        _LVOMoveWindow(a6)
 dbra      d6,window2loop

 lea       TheNewWindow,a0      ;use the new window information
 move.b    #1,nw_DetailPen(a0) ;change it slightly
 jsr       _LVOOpenWindow(a6)  ;and open a third window
 move.l    d0,otherwindow
 beq       abort

 move.l    otherwindow,a0       ;set its title
 lea       screentitle,a2
 lea       otherwindowtitle,a1
 jsr       _LVOSetWindowTitles(a6)
 move.l    otherwindow,a0
 clr.l     d0
 jsr       _LVOModifyIDCMP(a6) ;and deactivate the close gadget

 move.l    #50,d6
window3loop:                    ;slowly move otherwindow down and across
 move.l    otherwindow,a0
 move.l    #2,d1
 move.l    #1,d0
 jsr       _LVOMoveWindow(a6)
 dbra      d6,window3loop

 move.l    thiswindow,a0
 move.l    wd_UserPort(a0),a0  ;find the first windows user port address
 move.b    MP_SIGBIT(a0),d1    ;find the bit number of its signal
 move.l    #1,d0               ;calculate the signal value
 lsl.l     d1,d0
 move.l    _AbsExecBase,a6
 jsr       _LVOWait(a6)        ;and wait until the signal occurs

 move.l    intuitionbase,a6
 move.l    thiswindow,a0       ;close the windows
 jsr       _LVOCloseWindow(a6)
 move.l    thatwindow,a0
 jsr       _LVOCloseWindow(a6)
 move.l    otherwindow,a0
 jsr       _LVOCloseWindow(a6)
 move.l    customscreen,a0     ;close the screen
 jsr       _LVOCloseScreen(a6)
 move.l    _AbsExecBase,a6
 move.l    intuitionbase,a1
 jsr       _LVOCloseLibrary(a6)
abort:
 clr.l     d0
 rts
; --- IDCMP flag used            ;flags  and structures used by program
CLOSEWINDOW equ   $00000200      ;see include/intuition/intuition.i file
;-----Window flags used           ;for full notes
WINDOWSIZING   equ $0001
WINDOWDRAG     equ $0002
WINDOWDEPTH    equ $0004
WINDOWCLOSE    equ $0008
SIZEBRIGHT     equ $0010
;-----Screen flag used
CUSTOMSCREEN   equ   $000F
STRUCTURE Window,0
   APTR wd_NextWindow
   WORD wd_LeftEdge
   WORD wd_TopEdge
   WORD wd_Width
   WORD wd_Height
   WORD wd_MouseY
   WORD wd_MouseX
   WORD wd_MinWidth
   WORD wd_MinHeight
   WORD wd_MaxWidth
   WORD wd_MaxHeight
   LONG wd_Flags
   APTR wd_MenuStrip
   APTR wd_Title
   APTR wd_FirstRequest
```

```
    APTR wd_DMRequest
    WORD wd_ReqCount
    APTR wd_WScreen
    APTR wd_RPort
    BYTE wd_BorderLeft
    BYTE wd_BorderTop
    BYTE wd_BorderRight
    BYTE wd_BorderBottom
    APTR wd_BorderRPort
    APTR wd_FirstGadget
    APTR wd_Parent
    APTR wd_Descendant
    APTR wd_Pointer
    BYTE wd_PtrHeight
    BYTE wd_PtrWidth
    BYTE wd_XOffset
    BYTE wd_YOffset
    ULONG wd_IDCMPFlags
    APTR wd_UserPort
    APTR wd_WindowPort
    APTR wd_MessageKey
    BYTE wd_DetailPen
    BYTE wd_BlockPen
    APTR wd_CheckMark
    APTR wd_ScreenTitle
    SHORT wd_GZZMouseX
    SHORT wd_GZZMouseY
    SHORT wd_GZZWidth
    SHORT wd_GZZHeight
    APTR wd_ExtData
    APTR wd_UserData
    APTR wd_WLayer
    LABEL wd_Size
 STRUCTURE NewWindow,0
    WORD nw_LeftEdge
    WORD nw_TopEdge
    WORD nw_Width
    WORD nw_Height
    BYTE nw_DetailPen
    BYTE nw_BlockPen
    ULONG nw_IDCMPFlags
    LONG nw_Flags
    APTR nw_FirstGadget
    APTR nw_CheckMark
    APTR nw_Title
    APTR nw_Screen
    APTR nw_BitMap
    WORD nw_MinWidth
    WORD nw_MinHeight
    WORD nw_MaxWidth
    WORD nw_MaxHeight
    WORD nw_Type
    LABEL nw_SIZE
 SECTION data,DATA
IntuitionName:                     ;library name
 dc.b 'intuition.library',0
 dc.w  0
thiswindowtitle:                   ;window title text
 dc.b 'this window',0
 dc.w  0
thatwindowtitle:
 dc.b 'that window',0
 dc.w  0
otherwindowtitle:
 dc.b 'other window',0
 dc.w  0
screentitle:                       ;screen title text
 dc.b   'custom screen',0
 dc.w   0
TheNewScreen:                      ;NewScreen structure- see intuition.i
 dc.w 0                            ;LeftEdge
 dc.w 0                            ;TopEdge
```

►►►

```
 dc.w 320                       ;Width
 dc.w 200                       ;Height
 dc.w 3                         ;Depth
 dc.b 1                         ;DetailPen
 dc.b 3                         ;BlockPen
 dc.w 0                         ;special display modes
 dc.w CUSTOMSCREEN              ;screen type
 dc.l 0                         ;pointer to custom font structure
 dc.l screentitle               ;pointer to screen title
 dc.l 0                         ;pointer to screen gadgets
 dc.l 0                         ;pointer to custom bitmap

TheNewWindow:                   ;NewWindow structure for this window
 dc.w 20
 dc.w 20
 dc.w 150
 dc.w 50
 dc.b 0
 dc.b 1
 dc.l CLOSEWINDOW
 dc.l WINDOWSIZING+WINDOWDRAG+WINDOWDEPTH+WINDOWCLOSE+SIZEBRIGHT
 dc.l 0
 dc.l 0
 dc.l thiswindowtitle
 dc.l 0
 dc.l 0
 dc.w 64
 dc.w 20
 dc.w 300
 dc.w 200
 dc.w CUSTOMSCREEN

 SECTION mem,BSS
intuitionbase:            ;intuition library pointer
 ds.l 1
thiswindow:               ;pointers to Window structures
 ds.l 1
thatwindow:
 ds.l 1
otherwindow:
 ds.l 1
customscreen:             ;pointer to Screen structure
 ds.l 1
 END
```

```
/*c source code*/
#include <exec/types.h>
#include <intuition/intuition.h>
struct IntuitionBase *IntuitionBase;    /*library pointer*/
struct Screen *customscreen;            /*pointer to screen structure*/
struct Window *thiswindow;              /*pointers to window structures*/
struct Window *thatwindow;
struct Window *otherwindow;
char *thatwindowtitle="that window";    /*window title text*/
char *otherwindowtitle="other window";
char *screentitle="custom screen";      /*screen title text*/
long i;
struct NewScreen TheNewScreen =         /*the new screen structure*/
{
 0,0,320,200,3,
 1,3,
 NULL,
 CUSTOMSCREEN,
 NULL,
 "custom screen",
 NULL,
 NULL
};
struct NewWindow TheNewWindow =       /*a new window structure*/
{
 20,20,150,50,0,1,CLOSEWINDOW,
 WINDOWSIZING¦WINDOWDRAG¦WINDOWDEPTH¦WINDOWCLOSE¦SIZEBRIGHT,
 0,0,"this window",
```

Continued page 59 ▶▶▶

Amiga Aided Design

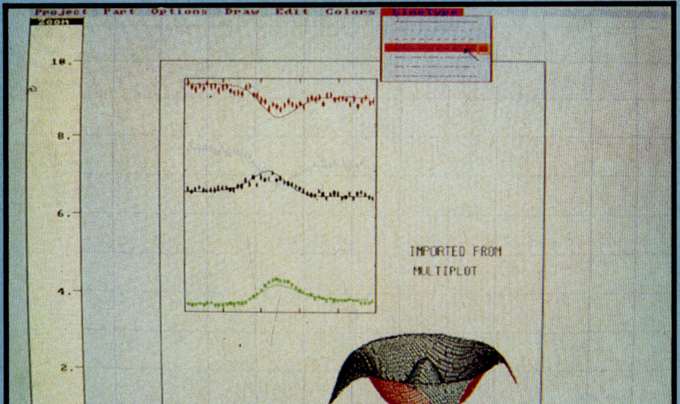
Buy an Amiga for CAD and you'll find a good selection of software. Anne Owen checks out a few options

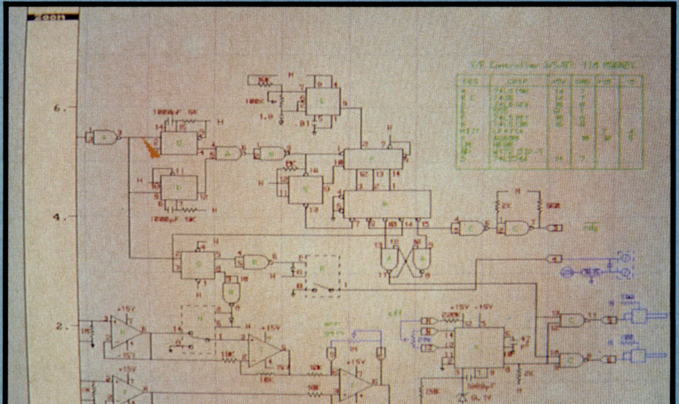
■ **Computer Aided Design is often seen** as a costly industrial process employed by the likes of the Ford Motor Company or British Telecom. Expensive workstations and plotters are envisaged and any thoughts of using a personal computer for CAD start to fade.

Not so with the Amiga. Although plotter hardware and hi-res screens are available, you can start with an Amiga 500 with 1Mbyte of memory and a dot matrix printer.

Entry level

■ ***IntroCAD* from Progressive Peripherals** even makes CAD possible on a 512K machine and proves that CAD can be achieved on the Amiga without large financial outlay. The name gives it all away: this is an entry level program, only £59.95 from Precision. However the packaging, excellently written and presented manual, and the software itself reveals no corner cutting.





IntroCAD is very nice to use with full Intuition interface. It works in high resolution mode so you have to contend with the famous flicker. The cheap way around this is a plastic screen (widely advertised), the expensive way (if you own an A2000) is MicroWay's Flicker Fixer board.

Back to the drawing board

■ ***IntroCAD*, like other CAD software,** employs an "object oriented" approach. This means that, in contrast to a paint program, everything you draw on screen is stored as a series of coordinate points. These points can be manipulated mathematically to perform actions such as rotation, scaling and zoom.

IntroCAD uses a second pointer design to differentiate a tool in use. The basic drawing functions available are: box, circle, line, arc and text. A tool remains in use until another is selected. Line type and colour can be selected beforehand or applied to an object after drawing.

To help the draughtsperson there are rulers at the top and lefthand side of the screen. These are marked out in inches (the Americans will never go metric!). A grid mesh can be turned on and the grid made as fine or coarse as necessary.

IntroCAD also allows the draughtsman to group objects together so that actions, such as rotation, are performed on the group as a whole.

Printer and plotter support consists of drivers for dot matrix printers and standard plotters. The program does not go through Preferences. Printer output can be quick, draft or final.

Prize Draw

■ ***Draw2000* from Aegis comes on three** discs, one with the main program, one a main program with 68020/68881 coprocessor board support, the third with example data files. I've seen no indication that the "2000" tag on this product restricts its use to the A2000 alone. Certainly the minimum hardware configuration for *Aegis Draw 2000* is one megabyte of memory and twin disk drives but such an A500 setup would cost in the region of £900 as opposed to £1200 for the 2000. Further memory expansion for the A500 is limited so this should be taken into account if multiple drawings in memory are a requirement.

The main new feature of *Draw 2000* is its ability to display drawings in Workbench windows. This means that, given the memory, you can have multiple drawings on screen, moving or copying objects between them. Within each drawing, you can have multiple layers (up to 250), allowing similar features to be displayed together, uncluttered. A single mouse click can move around layers, which can be saved or plotted separately.

Draw can bring together objects (circles, lines etc) into parts (or groups). The part can then be treated as a single object to be moved, cloned or saved to a directly editable parts list for future use. Here the power of the microcomputer approach to design really scores with libraries of useful parts quickly built up and ready to use. A part can always be exploded to edit it or use elements of it.

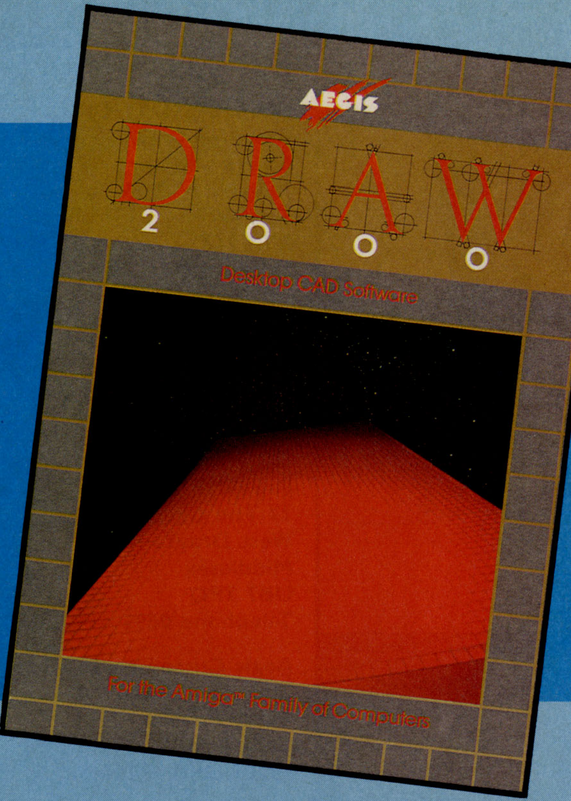
Shapes can be filled from a selection of eight line patterns designed for plotting. Dimensions can be added automatically and edited to fine tune presentation. Editing operations can be carried out upon marked areas of a drawing, operations such as the powerful mirror, distort and array, which allow flipping, proportional sizing and accurate movement and/or rotation.

Aegis Draw 2000 features a marvellous manual with tutorial, menu features and technical information such as file format and plotter driver details.

The coprocessor version of the software runs between 20 per cent and 50 per cent faster according to Aegis'own tests which we could not confirm without the add-on hardware. Because displaying large drawings does take a few seconds on a standard Amiga this version and the appropriate hardware could be used to improve productivity.

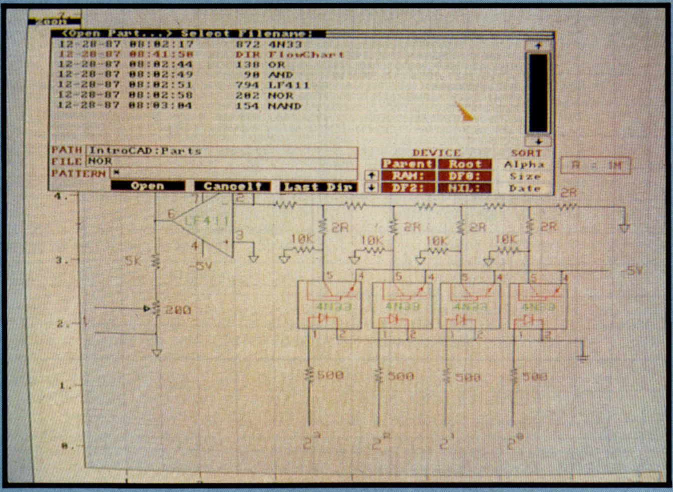
X marks the spot

■ ***X-CAD* from Taurus Impex is a** heavyweight CAD package for the professional user. Its recommended hardware requirements are 2.5-8.5MB RAM and 20MB hard disc. We

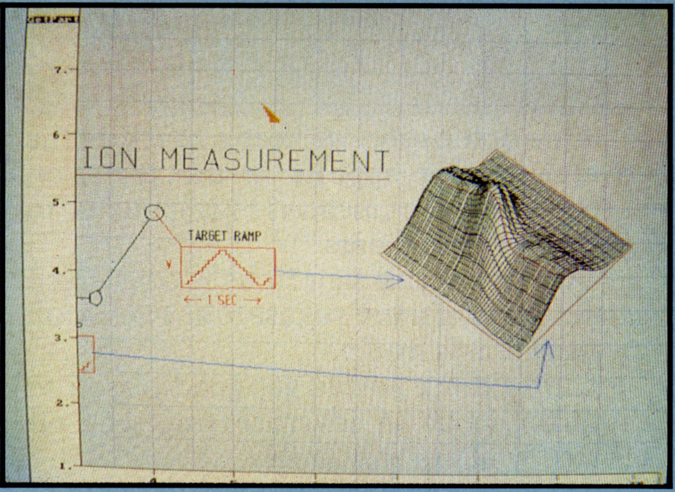


will therefore not dwell too long on this package in this review but endeavour to bring you a user profile in a future issue.

Input to *X-CAD* is achieved via automenus which display available options at every stage. An "Englishlanguage" interface is the command method provided. This is an inspired combination of the window environment which is considered the natural Amiga user interface and an input method well-known to CAD users. Commands such as OPEN DRAWING NAME, CHANGE TEXT PARAMETERS SLANT 10 WIDTH 4 and SELECT GRID XPITCH.1 YPITCH .2 are easy to understand in their context.



```
<Open Part...> Select Filename:
12-28-87 08:02:17      872 4N33
12-28-87 08:41:58      DIR FlowChart
12-28-87 08:02:44      138 OR
12-28-87 08:02:49       98 AND
12-28-87 08:02:51      794 LF411
12-28-87 08:02:58      202 NOR
12-28-87 08:03:04      154 NAND

PATH    IntroCAD:Parts
FILE    NOR
PATTERN *
```



A command can be followed by parameters taken from the mouse or digitising tablet eg DRAW CIRCLE DIAMETER: Loc x1 x2, where x1and x2 are defined by the user on the screen or tablet. Keyboard entry of commands and parameters is also possible at any time. An icon based menu can be built by the user to represent often used strings of commands.

A digitising tablet can be attached for tracing drawings and generally interacting with the program. By building a "tablet menu", the user can instigate functions from the tablet as an alternative to mouse or keyboard.

Some degree of automation can be achieved by building text files of commands, modifiers and parameters ready for execution. This can be done simply by opening a file on any device (eg PRT: or RAM:) to accept a sequence of keyboard commands.

X-CAD certainly contains a depth of features. Output support is excellent. There are commands to plot on Roland RD/GL, Hewlett Packard HPGL, Houston Instruments DM/PL and to a Kyocera laser printers' native graphics language. Rasterplotting is used to support dot matrix, thermal and other laserprinting devices. A raster image file is created at a specified dots per inch resolution. The drawing is split into a number of tiles which are collated and then compressed to conserve disk space. Printers supported in this way are Epson dot matrix, Mitsubishi colour thermal, Xerox 4020 colour inkjet and HPLaserJet+. Further support comes in the form of PLOT HARDCOPY which goes through Preferences and PLOT IFF which generates an IFF file. Both work on the current screen image, not the complete drawing.

What to buy?

■ **The home user will get good value** from *IntroCAD* because it does not require the purchase of expensive peripherals to get results. The school computer centre will benefit for the same reasons and if a plotter (such as a Linear Graphics Plotmate with HPGL ROM) is available then *IntroCAD* can take full advantage of it. Progressive Peripherals has also announced *UltraCAD* for $250 in the USA so it might be worth checking out its features before deciding on a purchase.

The professional CAD user, looking for powerful but easy to use CAD software, can look to *Draw 2000*. Existing Draw users should be happy that their desire for multiple drawings has been met. *Aegis Draw2000* retails for $279.95 in the USA. If you are a registered owner of Aegis Draw Plus then you should be notified of a $30 upgrade. Aegis Draw owners can upgrade for $150. At the time of writing no UK price had been set by its distributors.

X-CAD is more difficult to get to grips with but the power of the software is undoubted. The ability to customise the user interface is very advanced and ideal for the commercial environment. *YA*

Hardware

■ **If you do wish to embark on professional level CAD then the Amiga supports all the necessary hardware.**

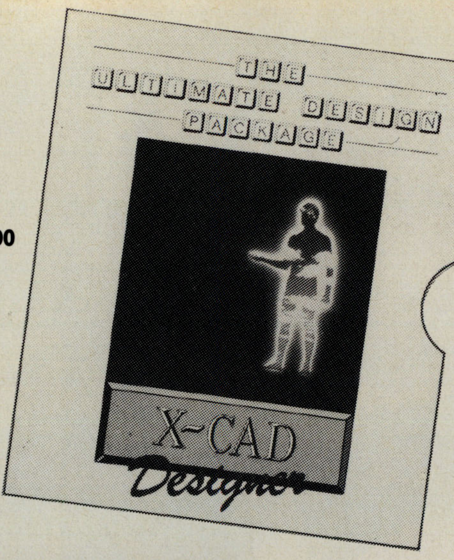
You'll need a megabyte — £100 upgrade for theA500 — for anything more than a single layered drawing and a plotter capable of doing justice to your drawings — £500 plus. But for professional users the right hardware can be considered an investment not a cost!

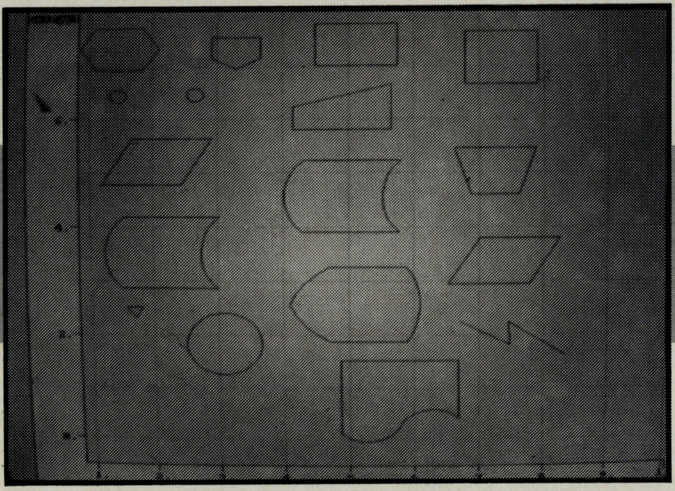
▶▶▶▶

INPUT
Cherry A3 digitising tablet £550

OUTPUT
Roland plotter £1500
Linear Graphics plotter £500
Laser printer + HPGL interpreter £2000
PostScript laser printer £3000

DISPLAY
NEC multisynch monitor £450
Taxan multisynch £350
MicroWay's Flicker Fixer £250
STORE
Upgrade to 1Mbyte (A500) £105
Upgrade to 3Mbyte (A2000) £395
Second floppy (A500) £217
Hard disc (A2000) £595





Factfile

■ ***IntroCAD(ITX-)***
Precision Sofware
6 Park Terrace
Worcester Park
Surrey
KT4 7JZ
Telephone 01 3307166

Progressive Peripherals & Software
464 Kalamath Street Denver
CO 80204
Telephone (303)825-4141

Aegis Draw 2000
HB Marketing
Brooklyn House
22 The Green
West Drayton
Middlesex
UB7 7PQ
Telephone 0895 444433

Aegis Customer Support
2115 Pico BBlvd
Santa Monica
California
90405 (213) 3926445

X-CAD
Taurus Impex
3 Bridge Street
Guildford
Surrey
GU14RY

It's easy to complain about an advertisement. Once you know how.

One of the ways we keep a check on the advertising that appears in the press, on posters and in the cinema is by responding to consumers' complaints.

Any complaint sent to us is considered carefully and, if there's a case to answer, a full investigation is made.

If you think you've got good reason to complain about an advertisement, send off for a copy of our free leaflet.

It will tell you all you need to know to help us process your complaint as quickly as possible.

The Advertising Standards Authority. ✓
If an advertisement is wrong, we're here to put it right.
ASA Ltd, Dept 1 Brook House,
Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.

SOFTViLLE PD SERVICE
55 HIGHFIELD AVENUE, WATERLOOVILLE
HAMPSHIRE PO7 7PY, ENGLAND. Tel 0705-266509

a SMALL selection from our collection of over 400 disks FULL of the best PD available for your
AMIGA

SOF7 - "PROBE", the best demo of its type.
SOF2 - TUTORIAL, crammed with information
SOF9 - PERFECT SOUND player/recorder etc.
SOF15 - CLIP ART in IFF formaat
SOF20 - NASA digiview slideshow, 14 pics
SOF26,27 - Two disks full of Comms programs
SOF10 - FULL of MIDI progrms
SOF11 - DISK TWO of MIDI utilities
SOF17 - RAY TRACING construction set
SOF21 - Pagesetter CLIP-ART
SOF25 - JET FLIGHT demo from Sublogic
SOF33,34,35 & 36 - Four disks full of the very best demos from the Badge Killer contest
SOF40 - The fabulous "Walking Legs" demo
SOF43 - AmigaSpell, spelling checker
SOF45,46,47 & 48 - Four disks full of tunes for use with the SONIX program.
SOF49 - FONTS & FONT UTILITIES
SOF50 - Printer Driver Generator
SOF54 & 55 - The brilliant NEWTEK demo. You need both disks, two drives and 1Meg
SOF56 - The incredible "DOC demo + 4 others
SOF57 - WORDWRIGHT Word processor

SOF59 - JUKEBOX MUSIC demo, great samples
SOF60 - Digitised music, over 11.5 hrs worth!
SOF66 - FIVE sampled tunes - half megs only
SOF67 - Bootbench V2.0 - make your own demo programs you've always marveled over.
SOF68 - CLASSIC BOARD GAMES
SOF69 - CLASSIC ARCADE/BOARD GAMES
SOF70 - CLASSIC AMIGA DEMOS
SOF72 - Four great graphic mini-movies
SOF73,74,75 & 76 - Miami Vice, digitised. You need all four disks (but only one drive).

All these plus the full range of FRED FISH disks (currently up to FF154), the best of the AMICUS collection, all the FAUG disks, the complete set of PaNorAmA disks and the SLIPPED DISK collection. Whatever you want, we've got it.

NO MEMBERSHIP - NOTHING TO JOIN

We simply provide the very BEST service to enable you to get great software cheaply. Plus GUARANTEED same day postage of all orders.

Disk prices; £3:00p each - £2:75p for 6 & over - £2:50p for 11 & over
Deduct £1:00p per disk price if you send in your own blank disks.
Cheques/po's payable to 'SOFTVILLE' at above address.

We also stock a great range of essential goods for every Amiga owner
PRO-5000 J/Stick (clear or black) - £13:50 Quickshot2 (Turbo) - £11:00
Disk Boxes (lockable) to hold 50 - £8:00 to hold 100 - £10:00
Library case to hold 10 - £1:50 MOUSE PADS, good quality - £3:50
CUMANA 1meg DRIVES - £104:99
A full range of Amiga specific books at discount prices - ask for list
Commercial Software at 25% off r.r.p
e.g. Great Giana Sisters - Fire & Forget - £18:25 each
plus Labels, Paper, Blank Disks, branded and bulk etc. etc.

PUBLIC DOMAIN SOFTWARE
Send a stamp for our FREE 24 page catalogue, state clearly "AMIGA" and this magazine.
BRITAINS No1 PD SUPPLIER

Data Maker

Convert those object files into convenient DATA statements

■ **Datamaker is a utility to convert data** files created by the Object Editor into DATA statements that can be merged into a Basic program. The DATA statements created by DATAMAKER describe the object's shape, size, colors and other attributes.

When the program is run a message will appear asking for the filename of the file created by the Object Editor, type in the filename exactly as it is on the disk and press RETURN.

The program will then ask for the output filename, that is the filename you want the data to be saved under. If the filename is the same as the filename of the Object Editor filename the program will ask you if you want to replace it or not. The program will then read the file and save the data file onto disk. The program will then ask you if you want to convert another file.

After the file has been created you can merge the data file into your own Basic programs with the following statement. MERGE "filename" (where filename is the name of the data file)
The following short program will read the data created and create the object.

```
REM *** DATAMAKER ***
REM *** REVISED VERSION 1.1 ***
REM.*** BY PIERLUIGI CERUTTI ***

WINDOW 1,"Datamaker",(0,100)-(400,180),22
WIDTH 38

opendata:
  CLS:PRINT
  PRINT "Enter filename of BOB or SPRITE:"
  LINE INPUT "1> ",filename$
  OPEN filename$ FOR INPUT AS 1
  object$=INPUT$(LOF(1),1)
  CLOSE 1

makedata:
  CLS:PRINT
  PRINT "Enter output filename for shape data:"
  LINE INPUT "1> ",filename2$
  IF filename2$=filename$ THEN
    PRINT "Output filename=input name..."
    PRINT "Replace existing object file (y/n)?"
    LINE INPUT "1>",answer$
    IF answer$="n" GOTO makedata
  END IF
  OPEN filename2$ FOR OUTPUT AS 1
  PRINT:PRINT "Creating ";filename2$;"..."
  datalines=INT(LEN(object$)/8)
  remainder=LEN(object$)-datalines*8
  PRINT#1,"' ";filename2$;" --";LEN(object$);"bytes."
  FOR n=1 TO datalines*8 STEP 8
    PRINT#1,"DATA ";
    FOR nn=0 TO 7
      PRINT#1,USING"###";ASC(MID$(object$,nn+n,1));
      IF nn<7 THEN PRINT#1,","; ELSE PRINT#1,""
    NEXT nn
  NEXT n
  PRINT#1,"DATA ";
  FOR n=1 TO remainder
    PRINT#1,USING"###";ASC(MID$(object$,(LEN(object$)-remainder)+1,1));
    IF n<remainder THEN PRINT#1,"," ELSE PRINT#1,""
  NEXT n
  CLOSE 1
  temp&=TIMER
  WHILE TIMER<temp&+6:WEND
  PRINT:PRINT "FINISHED."
  PRINT "Use program again (y/n)"
  LINE INPUT "1>",answer$
  IF answer$="y" GOTO opendata
  WINDOW 1,"Basic",,31:END
```

```
DIM shape$(elements)                (where shape$ is the string array
RESTORE linelable                   (that will contain the object data.
FOR n=1 TO elemants                 (elemants is the number of data elemants
   READ a                           (noted by the REM statement at the
   shape$=shape$+CHR$(a)            (beginning of the data statements.
NEXT n                              (linelabel is the label that you
OBJECT.SHAPE object-ID,shape$       (define at the start of the data
                                    (statements.
                                    (object-ID is the identification
                                    (number of the object that you
                                    (assign.
```

For an object called ALIEN which has 200 elements and with object-ID 2 the routine would look like this:

```
DIM alien$(200)
RESTORE aliendata
FOR n=1 TO 200
   READ a
   alien$=alien$+CHR$(a)
NEXT n
OBJECT.SHAPE 2,alien$
```

YA

FIRST WORDS IN C

C flexes its muscles to tap the memory supply of the Amiga in the form of arrays and dynamically allocated memory. Mark Burgess covers the basics

■ **Whatever the nature of your** programming, there comes a time when ambition finally exceeds the capabilities of purely high level computing. Sophistication begins to place demands upon programming languages which often cannot be met, and many programmers resort to machine code or assembly language in order to carry through their tasks unhindered. Not so the C programmer.

In purely high level programming there are only two choices for storing data: variables and arrays. Variables are either integers, floating point (real) numbers or characters and access to the computer's large supply of unused memory is held under asphyxiating control. The way out of this strangehold on computer resources is for programmers to actually take on the task of managing free memory personally: to stake a claim for a share of what memory is available as it becomes necessary. The disadvantage to this no nonsense approach (if you see it as a disadvantage) is that it becomes the programmer's responsibility to keep tight control of the memory which is claimed and to return it to the pool of free storage when it is finished with, so that other programs can use it. C enables this kind of memory management to be handled in a very high-level, worry-free way, allowing programs to grow dynamically in response to their needs.

Address Pointers

The memory of a computer can be thought of as a long line of storage compartments, lying side by side. Each compartment is called a byte and each one has a number, starting at zero and rising up to the maximum number. The number of each byte is called its address.

Addresses are crucially important, because they enable programs to look directly at particular places in memory, and to store things knowing how they can be retrieved. In high level languages, the idea is to completely eliminate the need to know where data are stored by giving variables names instead of using their addresses, but sometimes (as we shall soon see) this is too restrictive and it becomes necessary to return to the use of addresses.

C programmers have a slightly easier life than assembly language programmers do, in that C provides a class of variables called 'pointers' which save programmers the inconvenience of working with the actual numbers of addresses. In other words, it is not necessary to say 'I want to look at what has been stored at byte number 2145': the number itself is irrelevant and cumbersome to keep track of. Instead, C uses pointers which, used properly, keep track of the numbers for you.

What is a pointer? It is a variable whose purpose is to store the address at which another variable is stored. In other words, it is a way of jotting down the location of some information. A pointer is declared as a variable in its own right and it has a name of its own. This sounds a little peculiar, but before elaborating upon this, see how pointers are declared and how they are used.

Two special symbols are associated with pointers and addresses. They are * and &.
* is normally read as 'the contents of' and is used next to a pointer.
& is normally read as 'the address of' and is used next to a
variable (which is not a pointer).

To allow for all the peculiarities of computers, pointers also have types! A pointer which stores the address of a floating point variable may not be the same type as a pointer which holds the address of a character variable, for instance. Readers can object: surely an address is just an integer number, whatever happens to be stored there. This is true and in all probability the different pointers are the same, but not necessarily for some systems.

To declare a pointer to an integer then, you must declare:

```
int *ptr;
```

The only purpose of the * here is to define *ptr* to be a pointer. It should not be read 'the contents of'. *ptr* can be given a value. For example, define:

```
int i = 10;
```

then:

```
ptr = &i;
```

sets *ptr* to contain the address of the variable *i*. If you were to print out *ptr*, as though it were an integer, it would have some value of no great interest, namely the address of *i*. But we do not really care what that value is, because having a variable is better: it is a name, it is easier to remember. If we write:

```
printf ("%d",*ptr);
```

the result is 10. The reason for this is that the * symbol means 'the contents of' so **ptr* prints out the contents of the variable pointed to by *ptr*, which is *i* and has the value 10. The fact that the pointer has a type, ensures that the compiler understands that the number stored at the address held in *ptr* is an integer.

Consider a float type pointer.

```
float *fptr;
```

and two variables:

```
float x = 2.9, y = 3.2;
```

Writing:

```
fptr = &x;
```

makes *fptr* point to the address at which *x* is stored. The result of:

```
printf ("%f",*fptr);
```

is 2.9. Similarly,

```
fptr = &y;
```

makes *fptr* point to y and the result of

```
printf ("%f",*fptr);
```

is now 3.2.

Notice that pointers do not have to point to one object exclusively: they are variables and can be altered to point to any place at all in the memory.

Where does a pointer point?

Integer variables and floating point variables both take up at least four bytes of memory each, so which byte does a pointer point to? The answer is the address of the first byte: the start of a block of memory which holds the value. This is a convention which will be extended to more complex situations in which blocks of any size and shape can be manipulated by pointers.

To summarise:

A pointer holds the exact place in memory at which some variable is stored. It can be altered to point to another variable at any time and may be used to examine the value stored at that location.

The Point

This is all very well, but have we gained anything? In fact, yes, but it is not immediately obvious at this stage. Variable names are no longer needed. We have something more generally useful: names for manipulating any kind of chunk of memory f any size.

Recall that, in the last issue, file pointers were declared:

```
FILE *fp;
```

These are pointers of type *FILE*, which is actually a *struct* type object, consisting of several variables of different types and occupying many bytes. (More on this in a later issue.)

PEEK and POKE

The operators * and &, as used above, are the backbone of pointers. They give C programmers facilities more powerful than BASIC's PEEK and POKE commands, and even more elaborate than BBC BASIC's ?,! and $ operators. PEEK and POKE may be translated as follows:

PEEK(2748)->*(2748)

and

POKE address,value->*address = value

Arrays

Before going on to discuss memory management in the true sense, first consider simple arrays of variables. Arrays are a way of grouping several variables *of the same type* under a single name. In C, an array is defined using square brackets. A five element integer array would be declared like this:

```
int my__array[5];
```

This tells the compiler to create five integer variables, all referred to by the name *my__array*. The individual variables, grouped collectively as *my__array*, are accessed by using an index (plural: indices) like this:

```
my__array[0] = 10;
my__array[1] = 20;
my__array[2] = 30;
my__array[3] = 40;
my__array[4] = 1000;
```

The number in the square brackets refers to which slot in the array is required. Notice that the values run from zero to one less than the number of elements in the array.

The whole array might be printed out using *a* for loop:

```
/* int i; — declared at start */

for (i = 0; i < 5; i++)
    {
    printf ("%d",my__array[i]);
    }
```

To begin with, arrays are just lumps of memory: they contain garbage (the values are not initialized to zero). The values in the array could be set to zero using:

```
for (i = 0; i < 5; i++)
    {
    my__array[i] = 0;
    }
```

Arrays can be made up of any type of variable: float, double, char, short etc.. but the elements in the array are always of the same uniform type throughout.

How arrays are stored in C

Declaring an array is a simple way of getting a fixed amount of memory, bigger than a single variable. The compiler does the hard work of claiming the memory from the Amiga's memory manager and the memory is already 'typed' and mapped out. This means that the array is made up of a number of variables, all of the same type (int, char, float etc.), which are layed out side by side, in order, so that they can be accessed using the indices in square brackets as above.

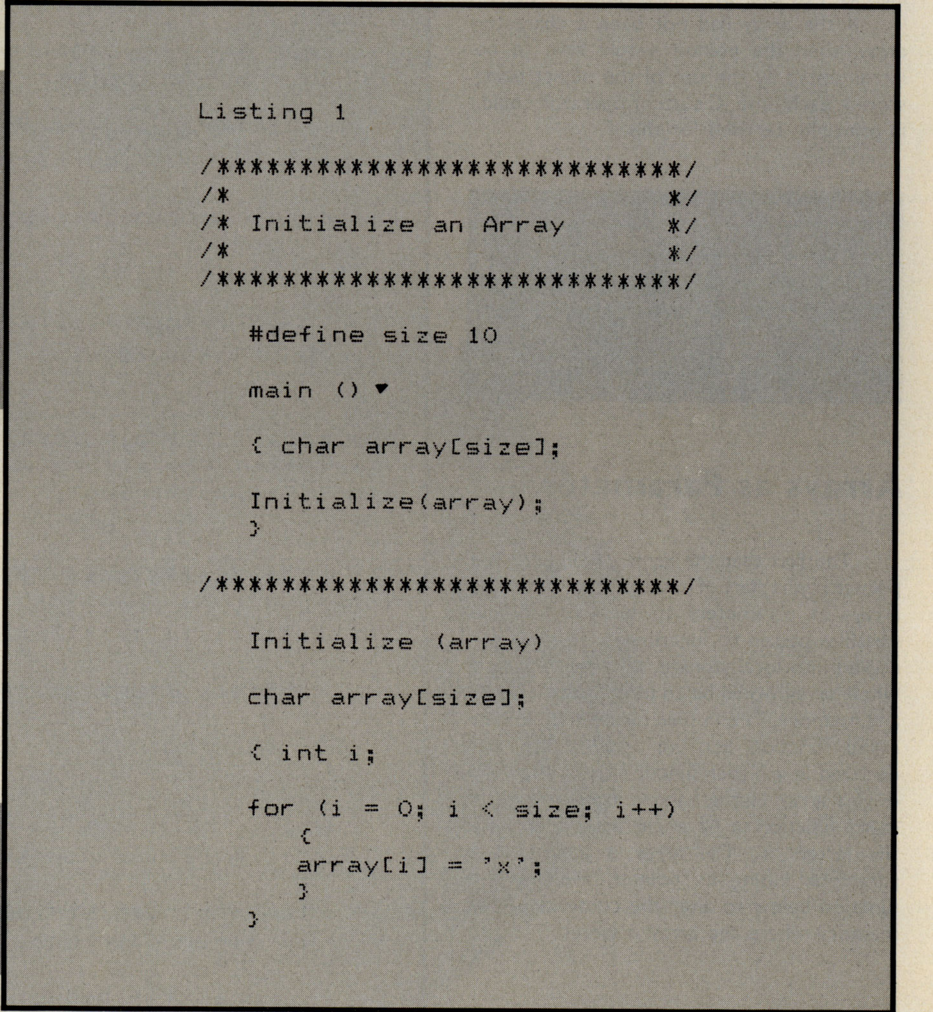
In the memory, the storage pattern is very straight forward: an array is stored as a complete 'contiguous' (unbroken) lump of memory and the individual elements are stored one after the other, as closely as possible. Normally they are stored literally end to end, one byte after the next, but this is not always the case as some variable types have to be aligned to start at particular addresses. (This means that some memory can be wasted by using an array, as certain bytes will simply be missed out in order to align values according to the requirements of a computer's processor.)

Arrays and pointers have a special relationship. The name or identifier of an array (such as *my__array*) is actually a pointer to the first element in the array. This is particularly important for dealing with strings (which are no more than arrays of characters).

The special feature about arrays of characters is that each character takes up only one byte, so the addresses of neighbouring characters in an array are only different by one. This immediately suggests two possible ways of printing out an array of characters. Define:

```
char ch__array[20];
char *chptr;
int i;
```

Now, assuming that the character array has



Listing 1

```
/*******************************/
/*                             */
/* Initialize an Array         */
/*                             */
/*******************************/

   #define size 10

   main ()

   { char array[size];

   Initialize(array);
   }

/*******************************/

   Initialize (array)

   char array[size];

   { int i;

   for (i = 0; i < size; i++)
      {
      array[i] = 'x';
      }
   }
```

been initialized with some characters, here are two equivalent ways of printing them out:

```
for (i = 0; i < 20; i++)
     {
     printf ("%c",ch__array[i]);
     }
```

and using the fact that ch__array is a pointer to the first character:

```
chptr = ch__array;

for (i = 0; i < 20; i++)
     {
     printf("%c",*chptr);
     chptr++;
     }
```

In the second case, we have introduced a character pointer *chptr* which is first set of be the same as *ch__array* (that is, it is set to point to the first character in the array). Then the loop prints out 'the contents of' the character pointed to by *chptr* and increments the pointer by one byte, so that as it goes around the loop, the address pointed to increases by one byte each time.

If the array had not been a character array, then the pointer would have to be incremented by the size of the object being stored, each time. C's *sizeof* operator could, in principle, be used for this:

```
intptr = int__array;

for (i = 0; i < 20; i++)
     {
   printf ("%d",*intptr);
   intptr += sizeof(int);
     }
```

Arrays as Parameters

The fact that the name of an array is a pointer to the start of it is exploited for passing arrays as parameters to functions. Unlike variables passed to functions, arrays are not copied into local storage, because copying a whole array would be an utter waste of time and memory. This in turn means that when a called function acts an array which was declared in a higher function, it changes the original array. Listing 1 shows a program which calls a function to fill an character array with the letter 'x'. The array is passed as a parameter, using its pointer, allowing the function below to alter the contents of the memory where the array is stored.

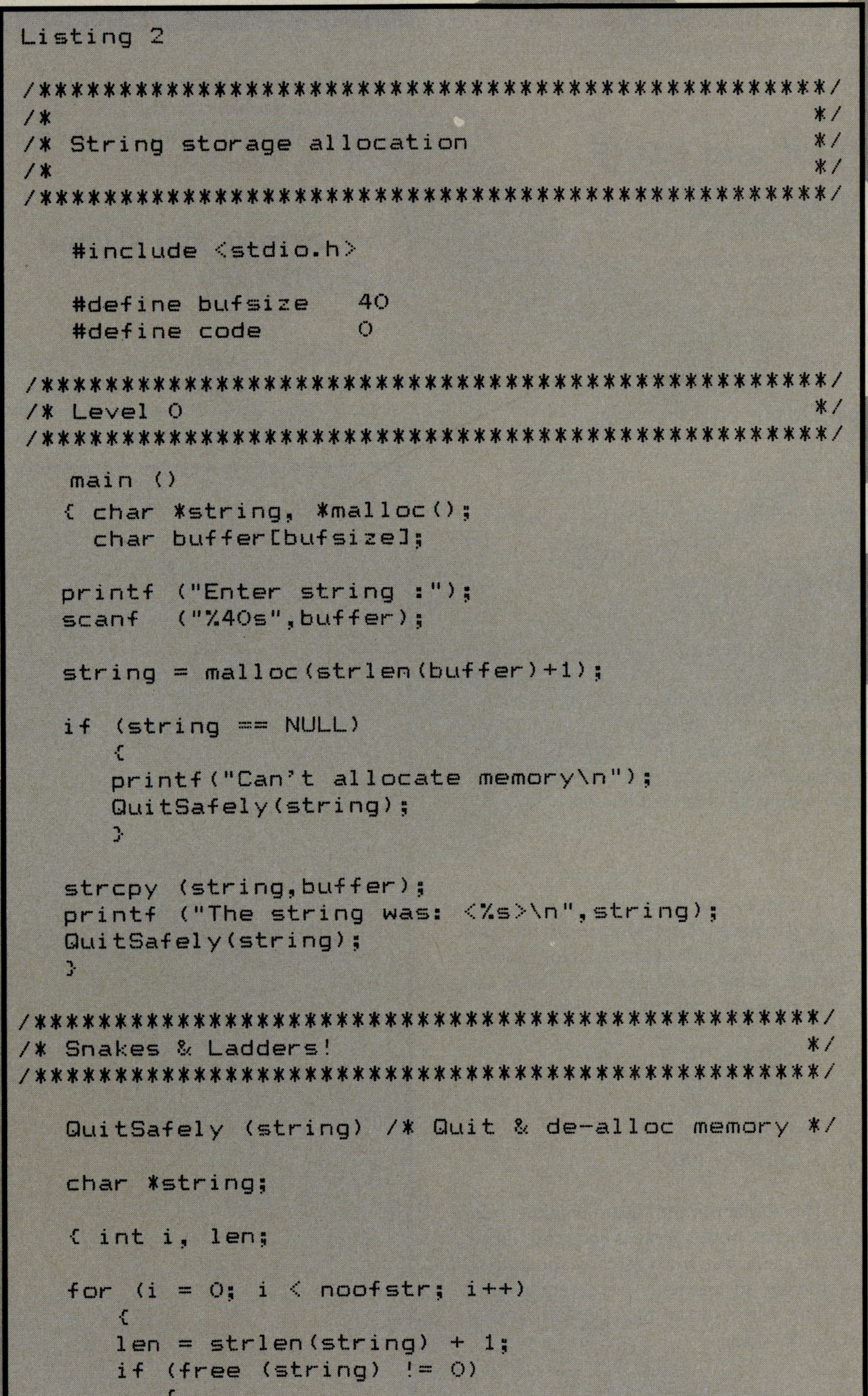
Freedom of Choice

C programmers have an option. When a variable is declared in a C program, the compiler calculates the number of bytes required to store all of the variables in memory and it stakes a claim with the Amiga's operating system for that amount of memory. Once a variable has been declared, however, its size is fixed. Once an array has been sized, it stays that size. Not all data behave in this way, however. You cannot program a computer user to require no more than thirty slots in an array. He/she may be faced with a pile of a hundred figures to store, with the pile growing all the time. Even a half decent program should be able to expand its storage in relationship to what is required.

A better arrangement is for a program to claim memory as it needs it and to use pointers to keep track of it. Programs which exploit this idea to the full can be extremely complex animals, which use many elaborate techniques for linking pointers to make chains of variables. Here we shall deal only with the basic ideas.

Memory Allocation

The C standard library contains a function



Listing 2

```
/**************************************************/
/*                                                */
/* String storage allocation                      */
/*                                                */
/**************************************************/

  #include <stdio.h>

  #define bufsize   40
  #define code      0

/**************************************************/
/* Level 0                                        */
/**************************************************/

  main ()
  { char *string, *malloc();
    char buffer[bufsize];

  printf ("Enter string :");
  scanf  ("%40s",buffer);

  string = malloc(strlen(buffer)+1);

  if (string == NULL)
     {
     printf("Can't allocate memory\n");
     QuitSafely(string);
     }

  strcpy (string,buffer);
  printf ("The string was: <%s>\n",string);
  QuitSafely(string);
  }

/**************************************************/
/* Snakes & Ladders!                              */
/**************************************************/

  QuitSafely (string) /* Quit & de-alloc memory */

  char *string;

  { int i, len;

  for (i = 0; i < noofstr; i++)
     {
     len = strlen(string) + 1;
     if (free (string) != 0)
        {
        printf ("Debug: free failed\n");
        }
     }

  exit (code);
  }
```

called *malloc*, short for 'memory allocation', which allows users to request to take control of a certain number of bytes of memory. *malloc* tries to comply with the request and if it succeeds, returns a pointer to the start of such a suitable block of memory. If it does not succeed it returns the value NULL. Notice that by using a pointer, the programmer is protected from the cumbersome knowledge of where the memory happens to be (which is probably quite irrelevant). The Amiga's memory manager decides whether or not the memory is free. The syntax is:

```
pointer = malloc(numberofbytes);
```

malloc is useful in a number of situations. The simplest is for storing strings typed in by the user. See listing 2 for an example. *malloc* has to be declared as a function which returns the type 'pointer to character', so you must include the line:

```
char *malloc();
```

in the declarations section of every function which uses *malloc*. Although it is declared in this way, it does not mean that you cannot obtain storage for data other than characters, it is simply a technicality. Once a character pointer has been obtained, the cast operator '(typename)' can be used to cast it into the proper type. For example, suppose you want to allocate memory for a *FILE* structure, you would write:

```
FILE *fp;
char *malloc();

fp = (FILE *)malloc(sizeof(FILE));
```

The cast for pointers is (type *). To claim 50 bytes for integers, you would write:

```
int *intptr;
char *malloc();

intptr = (int *)malloc(50);
```

Once this stage has been reached, the request for memory will have returned with either success or failure. It is essential to check the result for a *NULL* pointer, using:

```
if (intptr == NULL)
  {
  printf ("Failure! Out of memory?");
  /* safe exit */
  }
```

and to exit safely. There is no sense in trying to store information using the pointer if none was obtained. The result will simply be the Amiga fireworks display and a Guru Meditation message.

Once a block of memory becomes redundant, it should be returned to the free memory pool so that other programs can use it. This is done with the antithesis of *malloc*, called free(). The syntax is:

```
free (character pointer to block);
```

so a block of character storage would be freed by:

```
free(chptr);
```

and a *FILE* block would be freed by:

```
free((char *)fp);
```

and so on.

String Example

Listing 2 provides a simple illustration of a program which fetches a string from the user into a waiting buffer, allocates some memory to store the string, copies the string into that storage and prints it out. All of the essential features of memory allocation are present in the example without the complications introduced by type casting. Try modifying the example to cope with ten strings. The solution to this problem can be found in my book *C: A Dabhand Guide*, by Dabs Press.

Next Issue

Making room for strings is a very small part of what can be done with *malloc* and pointers. In the next issue, the idea of memory management and storage is taken a step further by introducing some sophisticated types of variables known as structures. These are the basic objects which the Amiga's operating uses to drive the special features of Intuition and AmigaDOS.

YA

Figure 1: A pointer is a variable which points to the start of some data

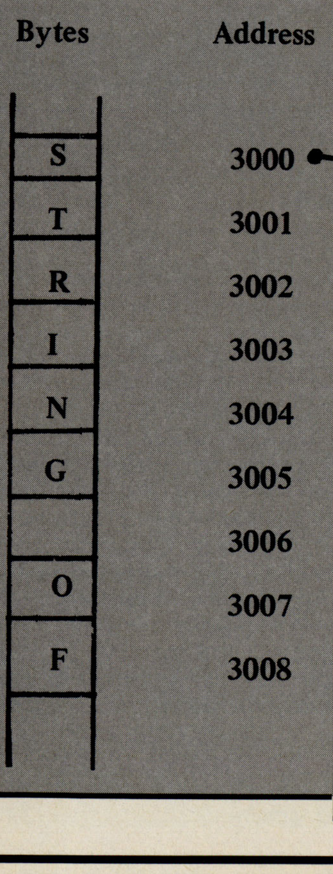


Figure 2: An integer array. The elements are stored next to each other. Each slot is the site of an integer

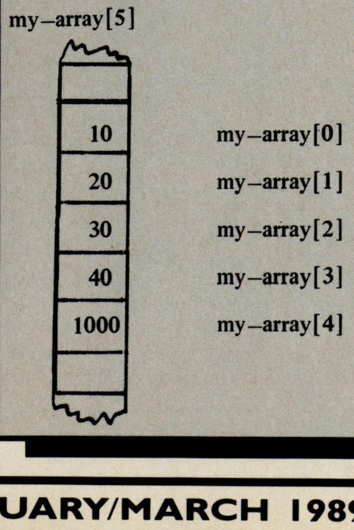
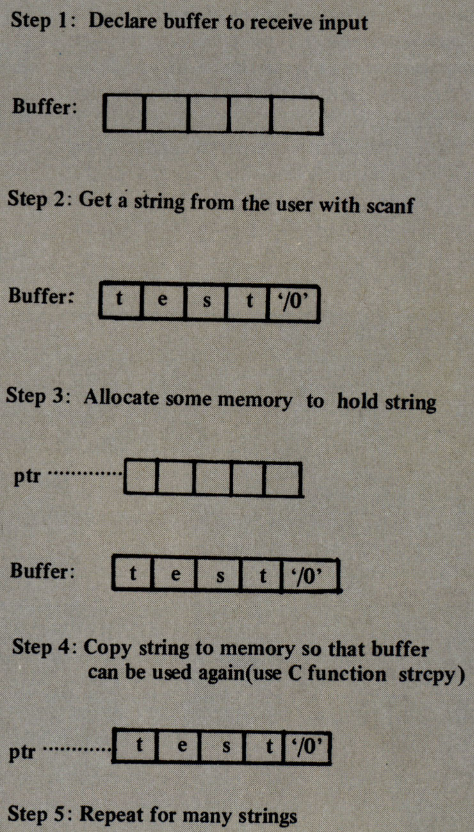


Figure 3: Memory allocation(see Listing 2)



Step 1: Declare buffer to receive input

Step 2: Get a string from the user with scanf

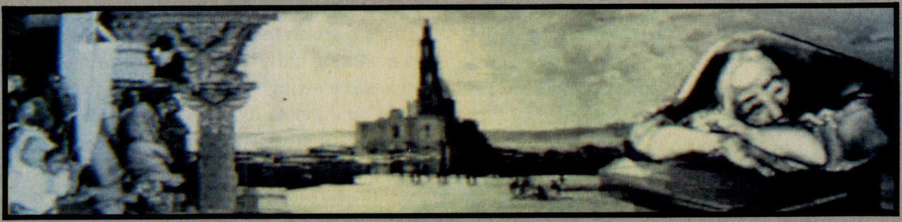
Step 3: Allocate some memory to hold string

Step 4: Copy string to memory so that buffer can be used again(use C function strcpy)

Step 5: Repeat for many strings

Lewis Tilley indulges in what he calls "Splashing at Ten-league Canvasses"

Photolab



▲**MEXICAN TRIPTYCH**▲: *A drawing, a photo and a montage blended into one image 960 pixels across*

■ **Kipling could not have known how** close to the truth of 1988 he would be with a line he wrote in the poem which starts "When Earth's last picture is painted, and the tubes are twisted and dried. ..." In the second stanza, he writes, " And those that were good shall be happy: they shall sit in a golden chair; They shall splash at a ten-league canvas with brushes of comets' bair."

True, not all Amiga artists have golden chairs, but *Deluxe PhotoLab* has given them the tools to be good, happiness and a screen canvas nine thousand, nine hundred and ninety nine pixels in both directions with brushes of comets' hair, cougars' hair or any kind of hair you choose to capture and manipulate with the hairiest brush and paint menus and sub menus yet devised.

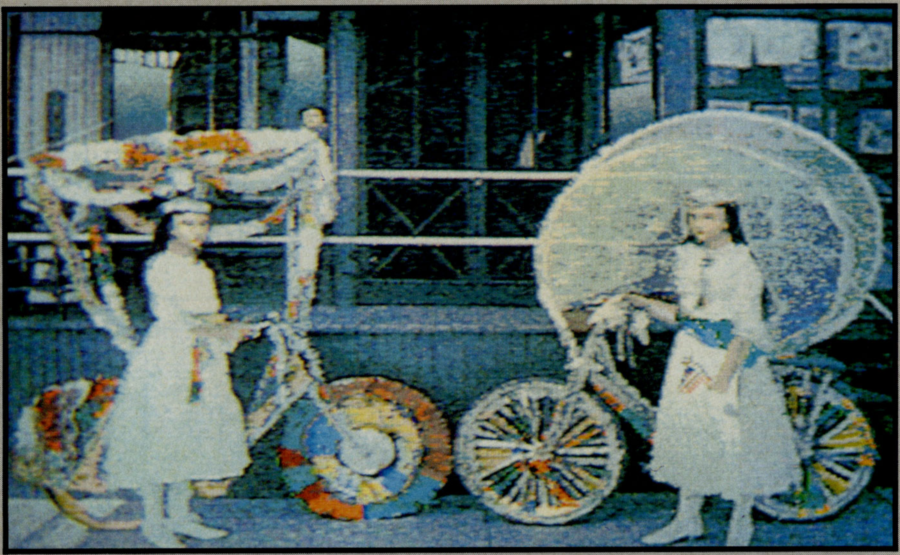
How do you expand a good paint program so that it is a *great* paint program without so slowing it down in its execution that the operator feels that he is walking through a slowmotion dream sequence? Why you divide it up into three separate programs and package them together. I like the way this is done in *Photolab*. The three icons which appear in the Workbench open window are a triangle for the Paint program, a circle for Color program and a rectangle for the Print program. That's good graphic design.

Deluxe Photolab contains not only a paint program, which functions very much like Electronic Arts' own classic *DPaint II* with the addition of a HAM mode, but also includes a pixel manipulator called Colors designed along the lines of *Butcher and Pixmate* and a print software program named *Posters* which lives up to its name.

This three course, graphic feast has more items in its menus than does a Caribbean Cruise Liner's main dining room. I found it necessary to slow down the usual top-speed function of my mouse control so that I wouldn't flip past my choice among the closely spaced selections offered.

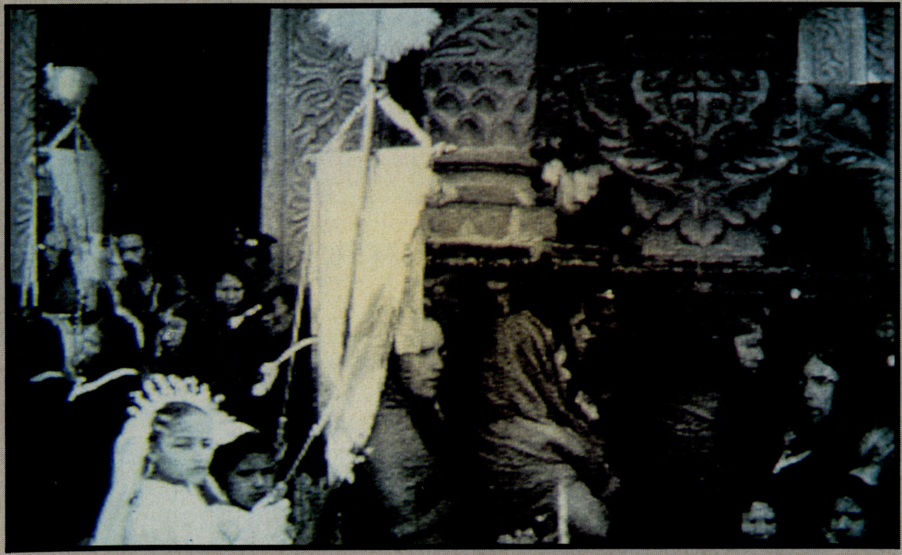
Simplifying the files

If you have been working for the past year or two with Amiga graphics, you are probably faced with pictures, brushes and images in your files which were done in all the modes and resolutions in which the Amiga makes it possible for us to work. I was becoming more and more frustrated by the elaborate system of multiple programs which I used in creating an image. Mind you, I shall continue to use all the elegant programs to which I have become accustom: *DPaint II* will execute perspectives, *DigiPaint* will tint like Titian and *Photon Paint* will spread a brush on cubes, cones and balls. However, the perfect partner for combining all these good old friends is now *Photolab*.



▲**US DIPTYCH**▲: *Here I used Photolab to colorise a photo of the town of Aurora, Missouri in 1900*

▼**GIRL WITH A BANNER**▼: *The basis for a montage of church architectural details*



What is more, the different resolutions may be placed on two different screens and worked on together by switching back and forth between them with the upper right hand double boxes. One of the most difficult habits from DPaint to break is the swap- screen 'J' key. *Photolab*'s "J" control, unfortunately, has not been programmed to swap screens. After you have brought up a screen, you may click anywhere on it to activate it.

Sometimes it is very difficult to grab the right place at the top of a screen in order to pull it up ... or down. With a brush on the end of your pointer, it seems to be even more difficult. I would have preferred a better way to swap screens.

▲OLD WOMAN▲: A charcoal drawing digitised in hi-res with DigiView, then combined using Photolab

The storage miracle

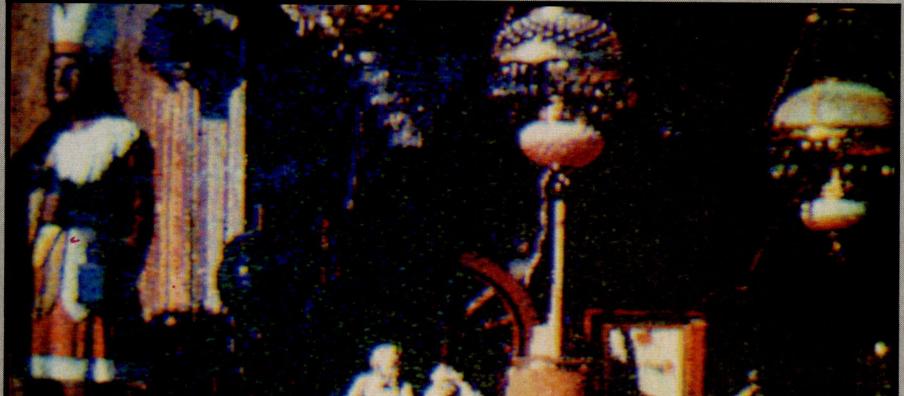
Where are all these screens coming from? Unlike DPaint and most other paint programs which use only the 512K of Chip RAM for storing graphic images, *Photolab* stores in whatever free RAM you might have. Aren't you happy that you bought those extra Megabytes of RAM while the prices were still low? This feature is responsible for the expanded Page Size (up to 9999 × 9999 pixels IF you have the memory).

A Megabyte Amiga will give you a 800 × 800 page. Three Megabytes seem to give me a workable 1280 × 1600 page size while using HAM and Interlace. (This was established by the empirical approach to mathematics). Be warned, however, that you cannot SAVE the whole page of 16 screens on one disk. I could save a 9 screen page to a disk taking up 328,566 bytes plus 979 for the icon, but since the 320 × 400 full screens in HAM and interlace generally take up 50K each and the icons and other little niceties on a disk eat away at the storage I was just lucky to receive a warning when I tried to save 16 at once. Loading saved images back into the program gives a little problem, too. If you stray from the normal page size, screen size and resolution/interlace which is expected, you can sit a long, long time waiting for the images to load.

How to keep loading sweet and short? Since there is no automatic feature for loading the correct display mode your file needs...nor the right amount of space in page size it needs, either keep good notes or stick to HAM interlace for convenient and top image quality. I must, in all fairness, admit there is a record kept which is easily accessible to tell you the space you need. Underneath the cleanest icon replica of your image there is saved the page size it loads into; however, since it is buried back there on the workbench with the other icons, I never see it when I am loading from within the program. Another trick is to always use the "Load at" load. It will tell you how much space you will be needing.

I keep an Amiga 2000 at the University of Southern Colorado (it is the one which has three Megabytes). In Georgia, where I am now for the winter, my Amiga 1000 has an "Insider" Meg to give it one and a half total. I miss the extra power when working with *Photolab*.

▲SCROOGE BUYS A TURKEY▲: Originally a HAM image. The Photolab Colour program translated it to lo-res



▼CIGAR STORE INDIAN▼: Digitised from a transparency before becoming part of the US Diptych

The Tools

So, the canvas is 20 leagues across, but what about the paint brushes of comets' hair? The tools are really not all that different from what you are using with *Photon Paint* and *Deluxe PaintII* ... except for a Curve Drawing tool which makes it down right sinful to draw the human body. I've had my share of sin with the simple one curve tool which other paint programs have, but this one does it like a French curve — two curves. First you set the two points at the beginning and the end of the straight line. Lift your left finger and slide the mouse to pull the first curve out. Now press down your finger again while sliding the curve at the opposite end of the line to a position which will please you. When you release this time, the curve draws itself to the placement you have selected. Between the raising and lowering of fingers, however, the double curve seems to feed on itself, pulling the line away from the other curve.

As difficult as it may sound, you'd better get accustomed to this method of drawing for here come the drawing packages for use with laser printers and *Postscript*. This setting of four and more points is the way they do it in *Professional Draw*.

ResizeDraw tool in the Brush menu saves a lot of time. You draw a rectangle the size you want the brush while it is still attached to the tip of your mouse-finger.

The small row of colors at the top of the screen is inadequate for the selection and use of colors in HAM. So? Punch the screen button upper right ... whoops, did you get the workbench screen? Sometimes I do; so take the safer way with me and touch the "P" key on your keyboard. Here comes the BIG palette screen! It is half the size as a full screen and contains row after row of little pans of color. Some, well, most are empty black when you flip to this palette page, but by copying from the basic palette at the top which duplicate the one on your main painting screen you can transfer, manipulate and, what is most important, SPREAD one color into another for a most exciting blend of colors accessible immediately from HAM mode and this Paint Set Palette of 128 colors. Remember, this was one of the major omissions in the otherwise excellent *Photon Paint*. The new update to *Photon Paint* is said to have it.

Do I personally find this palette, grand as it is, satisfactory on a second, hidden page? Would I paint in the living room of my house but keep the palette in the kitchen? Perhaps an artist raised on computers can pull off this trick, but I have known the convenience of a palette just before me on the tray beneath my easel. It will take some getting use to this hidden palette trick.

The "p" for palette key is much used by me. So is the "v" for "Volumes" key which summons the names of the volumes in your disk drives which contain your directories and picture files. Gone is the comforting gadget to punch up DF0: , DF1: and RAM:

The MENU at the top of your painting screen offers a very strong group of options. As a matter of fact, there is one menu entitiled, "Options." It has my next favorite key. "r" key repeats the last action for as many tines as you choose to press it. Like a well executed wash in an old fashioned architectural rendering, you may slowly build up a darker tone by overlaying many lighter tones.

The pull down sub-menu under MODES/PAINT is very accurately calibrated for the amount of a selected color you wish to apply. It is a little upside down, however in its listing. Solid color is at the top. Below Solid (full strength) is LOW MIX (a very faint glaze) followed in ascending strength order by Mix (1/4 strength), Average (1/2), and Blend (3/4).

Shade is similar to *Photon Paint*'s control once you have requested the Shade Control under the Options Menu. There is a point to move around to where you want maximum density of a brush and the horizontal and vertical control bars are there in an expanded form which lets you determine to which of the edges you want them relating. I must confess that for me the visual presentation of the effects in *Photon Paint* are easier to work with when using this feature.

Here is a list of some of the other of the many, many manipulative features found under the Paint Modes submenu:

Subtract, Add, Scale, Scale2, Max and Min, XOR, OR, and AND. Then there is the HLF which greys down everything in the image by taking away half the saturation of the colors in the area selected and B&W which takes away all color.

And this is only a beginning of features in menus and submenus. There are four fill control selections: Solid Color, Brush Pattern, Trace Edges and Gradient (with dithering). You specify offset to Brush Patterns and choose what pixels are to be changed that are close in value to your fill area by selecting Fill Offset and HAM Closeness respectively.

There is also a BRUSH Mode submenu which specifies the normal brush as Matte, a one color brush names Color, a drawing brush of pattern called Pattern and Store which places a rectangle around your brush.

The real BRUSH MENU enables you to do all those things you want a brush to do plus a fancy Resize Draw mentioned earlier and a Grab Last which picks up the last drawing done on the screen. Hold down the Shift Key while executing Grab Last and you get what was underneath the last object drawn. Remap command lets you take full advantage of the *Photolab*'s ability to work between two different screens at the same time. It will true-up the color differences between a brush with one palette and a picture with another.

COLOR

COLOR is the easiest way to convert HAM images to other formats I have yet used. A control panel more than half a screen high loads your HAM image behind itself where you may move it up and down while you evaluate the changes about to be made. Load an image in so that the menus are activated.Just below the top of the COLOR MENU line is shown the CURRENT INFO: which gives the mode, depth of planes, width and height of the screen. You may easily change any of these if the program allows a change from the visible and invisible submenus. I stress the IF because I crashed the program several times trying to change a HAM into a HiRes interlace image.

The stepping down works like a charm. From the top menu with the right mouse button depressed you are rewarded with a SET TO: submenu. Any present display mode is offered. I like the decisions of color modifications the program makes from the HAM down to Medium and Low res. Over on the side of the control board is a vertical row of single letters. These controls are rewarding when you use them to change an image overall. Especially enriching is the "V" for Value which picks up contrast as the value lowers.

Color manipulations are just as easy once you figure out that the charts of individual colors need to be locked down, leaving the one you wish to manipulate free to increase or decrease the amount of the letter (on-the-left) chosen RGBcolor by clicking the UP-TRIANGLE and DOWN-TRIANGLE gadgets to the right of the chart window.

I may find that using the PAINT program of *Photolab* slows the present rapid painting style I have developed using PhotonPaint, DigiView and DeluxePaintII, but COLOR, in the *Photolab* package, has become an extension of image manipulation which has supplemented the more difficult for me to understand competitors.

Printing

The last of *Photolab*'s programs, *Poster*, wipes away all other printing programs I have ever used. Since it was introduced to my modest OKIMATE 20 color printer, the Oki has started thinking it is a little brother to the costlier big boys. The second graphic choice in preferences offers the 1.3 workbench choices. HalfTone makes a cleaner color print for me than the default mode. The visual size chart shows you exactly what size you are printing and where the paper divisions are going to be. This grid of eight and a half by eleven inch pages totals 176 pages. You may place your image on each page vertically or horizontally.

The aspect ration may be changed so that you may squeeze or stretch an image in its printing...or print it normally with its aspect ration maintained at any setting. The size of your print out? It ranges an unbelievable one inch by one inch up to ten feet by ten feet. My muralist training at last may well come in handy with the Amiga.

Summary

Photolab is at the leading edge of paint programs with this version; pushing, sometimes straining to the AMIGA'S limits. I hope my few personal aversions will not keep you from the adventure of operating at a level which it offers. If you have had experience with other programs such as *DPaint II*, then go for *Photolab*. The collection of three-in-one makes it a package unmatched today. ***YA***

Title: **Deluxe Photolab**
Supplier: **Electronic Arts**
Langley Business Centre
11-49 Station Road
Langley
Nr Slough
Berks SL3 7YN
Tel: **(0753) 49442**
Price: **TBA**

Assembly Language Programming

Hungering for a little 68000 cuisine but don't quite fancy anything on the menu? Then read on and let Peter Lawrence show you how you can write your own

■ **In this installment I'm going to con**tinue looking at calling Intuition functions from assembly language by considering menus. First a little background to whet the appetite.

The Aperitif

As I'm sure you are aware that Amiga menus offer a convenient way of allowing user input. Menu selections are of two types: actions and attributes. Actions are commands which, when selected, are executed. Attributes on the other hand are options which may be switched on and off. They stay in effect until they are cancelled. In addition a series of menu attributes may be deemed mutually exclusive which means that selecting one attribute automatically implies the cancellation of all the others.

Menus are attached to a window so that the particular series of menus (known as a menu strip) which appears when the right mouse button is pressed is determined by which window is active at the time.

The Main Course

A menu strip is a series of menus which appear side by side at the top of the screen, each below a separate heading.

As far as the application program is concerned, a menu strip is a linked list of structures known as Menu structures. The menu structure is described below in the usual format.

```
WORD     mu__LeftEdge, mu__TopEdge      {0,2}
WORD     mu__Width, Height              {4,6}
UWORD    mu__Flags                      {8}
APTR     mu__MenuName, mu__FirstItem    {10,14}
WORD     mu__JazzX, mu__JazzY           {18,20}
WORD     Mu__BeatX, mu__BeatY           {22,24}
```

NextMenu — contains a pointer to the next menu structure in the menu strip. If this is the last then this field should be set to NULL.

LeftEdge, TopEdge, Width and Height — all describe the select box of the menu's heading. In the current release of Intuition TopEdge and Height are both ignored.

Flags — can have the following values:

MENUENABLED (=1) determines whether this particular menu is enabled and hence selectable.

MIDRAWN (=$100) indicates whether the menu's item are currently displayed to the user. It is maintained by Intuition and may be tested by the application program should it need this information.

MenuName — is a pointer to a NULL terminated string of ASCII characters which specify the heading which is to appear above the menu.

FirstItem — points to the first item in the menu as described below.

JazzX, JazzY, BeatX and BeatY — If you're wondering what these odd, rather musical sounding fields really do then join the club because I've got no idea. They are, in fact, used by Intuition for purely internal housekeeping reasons and so need not concern us.

Each menu consists of a series of items from which the user may select. Each one of these items is represented within the system by the MenuItem structure.

Note that LeftEdge is relative to the left edge of the menu and not the left edge of the screen.

Flags — can be set as follows.

CHECKIT (=1) identifies the menu item as an attribute which will be marked with check mark when it is selected.

CHECKED (=$100) is set by Intuition whenever the item is selected, assuming it is an attribute item. It can be initialised by the user when the menu is first created.

ITEMTEXT (=2) when set tells Intuition to render this item on the menu as text defined by an IntuiText structure pointed to by the ItemFill field. However, if this flag is not set then the item is rendered as an image and ItemFill should point to an Image structure.

COMMSEQ (=4) specifies that this item may also be selected by a command key sequence (the right Amiga key held simultaneously with another key).

ITEMENABLED (=$10) specifies if the item is enabled and hence selectable. It may be altered by the user only by using the OnMenu and OffMenu functions (see below for details).

HIGHFLAGS these 4 flags, HIGHCOMP (=$40), HIGHBOX(=$80), HIGHIMAGE(=0) and HIGHNONE(=$C0) determine how the item will be highlighted when selected. They are the same as the corresponding flags we have discussed in the past with relation to gadgets.

MutualExclude — this 32 bit field applies only to attribute items and specifies which other attribute items should be automatically deselected when this item is selected. Each bit of this field corresponds to an item in the menu, bit 0 is the first item, bit 1 the second and so on.

SelectFill — Similar to ItemFill, this field points to either an Image or IntuiText structure depending upon the setting of the ITEMTEXT flag. This is only valid if the HIGHIMAGE flag is set.

Command — is used if the COMMSEQ flag is set indicating that a command key sequence can be used. This field contains the key which, when combined with the right Amiga key, selects this menu item.

```
APTR     mi__NextItem                     {0}
WORD     mi__LeftEdge, mi__TopEdge        {4,6}
WORD     mi__Width, mi__Height            {8,10}
UWORD    mi__Flags                        {12}
LONG     mi__MutualExclude                {14}
APTR     mi__ItemFill, mi__SelectFill     {18,22}
BYTE     mi__Command, mi__Filler          {26,27}
APTR     mi__SubItem                      {28}
UWORD    mi__NextSelect                   {32}
```

NextItem — points to the next MenuItem structure in the list. For the last item it should be zero.

LeftEdge, TopEdge, Width and Height — describe the select box of the menu item.

SubItem — points to another linked list of MenuItem structures which specify subitems associated with the main item.

NextSelect — is filled by Intuition when the item is selected by the user. Your program should check this field. If it is equal to MENUNULL (=$FFFF) no unprocessed menu selections remain, otherwise the field indicates a user selection which should be processed by the program. More detail of how to do this is presented later in this article.

Along with these new structures there are several new Intuition library functions to be introduced.

SetMenuStrip {-264} — attaches the menu strip, the first Menu structure of which should be pointed to by register A1, to the window whose Window structure is pointed to by A0.

ClearMenuStrip {-54} — detaches the menu strip from the window whose Window structure is pointed to by A0.

ItemAddress {-144} — takes the address of the menu strip

(i.e. the first menu structure in the list) in A0 and the MenuNumber in D0 and then returns with the address of the selected MenuItem structure in D0. If the menu number passed to it does not correspond to an existing menu item then the function returns a zero. It is important to test for this result because attempting to process a non-existent menu item will almost certainly crash at least your task, if not the whole system. This can easily happen if, for example, the user selects a menu then releases the menu button while still pointing to the menu's header. Your program will still receive a MENUPICK message but the menu number in the message will be invalid.

It is advisable to always use this function when interpreting menu numbers thus ensuring upward compatibility of your software.

OffMenu {-180} — when A0 points to a window structure this function will disable the menu or menu item indicated by D0. D0 contains not a pointer to the appropriate structure but something called a menu number. This is a 16 bit number where the lower five bits specify which menu is involved, the next six bits indicate the menu item involved and the high 5 bits specify the subitem selected.

OnMenu {-192} takes the same arguments but enables the item rather than disabling it.

To set up menus on a window for use in a program we need to follow the following steps.

- Set up a NewWindow structure with the MENUPICK flag set then open the window.
- Design and link together the Menu and MenuItem structures to form a menu strip.
- Attach the completed menu strip to a window using the SetMenuStrip function.

To respond to menu selections your program should,

- Wait for messages of type MENUPICK.
- Examine the im__Code field of the IntuiMessage structure returned to find the menu number.
- Translate this menu number into a MenuItem address using the ItemAddress function and take the appropriate action.
- Check the mi__NextSelect field to see if multiple items were selected and if so act accordingly.
- Reply to the message.

Dessert

It's time for something we can really get our teeth stuck into so let's look at a practical example.

The listing presented here sets up a small window on the Workbench screen and attaches a menu strip to it. This consists of two menus each with three items.

The first of these menus consists entirely of action items which in this case are set up to do this, that and the other — literally. When any of these items are selected the program determines which item it was and then processes the request by calling a subroutine specifically written for the purpose. In this case all these three routines do is print a simple message on the screen via the Intuition function PrintIText. They could, of course, do whatever you like.

This menu also demonstrates the use of keyboard command sequences for menu action items. The three items can be selected without using the mouse merely by holding down the right Amiga key and then pressing the appropriate command key. In this program the three items are selectable using the A, B and C keys respectively. If you run this program you will notice that the menu items include a symbol to the right of the test describing their function. These symbols indicate the command key alternatives which have been set up for the menu item and are constructed by Intuition automatically whenever you specify an item to have a command key sequence. You should always make sure when designing your menus that the items select box is wide enough to include this symbol as well as the descriptive text or image which you have supplied.

The second menu consists entirely of attributes, although in your own programs you are free to mix attribute and action items within the same menu. When these items are selected a large tick appears to the left of the item to demonstrate this fact. This image is supplied by Intuition and so, once again, you must be sure to leave space for it. If you would prefer a different image for marking selected items this can be set up when the window is opened. Simply set up an Image structure to describe the desired check mark and store a pointer to it in the nw__CheckMark field of the NewWindow structure before calling the OpenWindow function. This was briefly covered in part three of this series.

This menu also illustrates how to use the mutual exclude flag. The three items in the menu describe what might philosophically be called *States of Being*. Specifically these are **Having Cake**, **Eating Cake** and **Feeling Hungry**. Clearly these states are not entirely independent upon one another and so this has been reflected in the way the mutual exclude flags were set. If you run the program you will find that while it is possible to **Have Cake** and **Feel Hungry** you cannot **Eat Cake** and **Feel Hungry** (greed is not tolerated in our philosophical world). In accordance with the laws of all universes whether philosophical or otherwise, you cannot **Have Cake** and **Eat It**.

Cheese & Biscuits

That about rounds up our treatment of the Intuition library. While we have certainly not covered all aspects of this rather vast topic it is about time we moved on to look at some other aspects of the Amiga's operating system, and there is certainly a lot to be going on with.

Next time around we'll be gorging ourselves on the DOS library and looking at how to handle files management and I/O, including taking input from the keyboard and sending output to the screen.

I wonder if version 1.3 will have a wine card too! *YA*

LISTING 1
TIME FOR DINNER

```
*  Constants and Flags  *

AbsExecBase        = 4
LIBRARY_VERSION    = 33
CLOSEWINDOW        = 512
MENUPICK           = $100
MENUENABLED        = 1
ITEMENABLED        = $10
ITEMTEXT           = 2
COMMSEQ            = 4
CHECKIT            = 1
HIGHCOMP           = $40
JAM1               = 0
JAM2               = 1
MENUNULL           = $FFFF
NULL               = 0

*  Library Offset Vectors  *

*-Exec Library-*

LVOOpenLibrary   = -552
LVOCloseLibrary  = -414
LVOWaitPort      = -384
LVOGetMsg        = -372
LVOReplyMsg      = -378

*-Intuition Library-*

LVOOpenWindow    = -204
LVOCloseWindow   = -72
LVOSetMenuStrip  = -264
LVOClearMenuStrip= -54
LVOItemAddress   = -144
LVOPrintIText    = -216

*-Offsets Into Structures-*

wd_UserPort      = 86
wd_RPort         = 50
mg_CLASS         = 20
mg_CODE          = 24
mi_UserRoutine   = 34

*  This mi_UserRoutine is not part  *
*  of the standard MenuItem         *
*  structure but this field has     *
*  been added to contain an APTR    *
*  to the routine which will        *
*  handle user selections of the    *
*  menu item.                       *

*  Main Programme  *


CODE

Start:

*  First open the Intuition Library  *
                lea     IntuitionName,a1
                move.l #LIBRARY_VERSION,d0
                move.l AbsExecBase,a6
                jsr    LVOOpenLibrary(a6)
                move.l d0,IntuitionBase

*  If Intuition failed to open then exit  *

                beq.l  IntuitionFail

*  Open a window  *

                lea    NewWdw,a0
                move.l IntuitionBase,a6
                jsr    LVOOpenWindow(a6)
                move.l d0,Wdw

*  If it fails then clean up and exit  *

                beq.L  WindowFail

*  Attach the menu strip to the window  *

                move.l Wdw,a0
                lea    Menu1,a1
                move.l IntuitionBase,a6
                jsr LVOSetMenuStrip(a6)

*  Wait for an IDCMP message  *

Waiting:        move.l Wdw,a0
                move.l wd_UserPort(a0),a0
                move.l a0,MPort
                move.l AbsExecBase,a6
                jsr    LVOWaitPort(a6)
                move.l MPort,a0
                jsr    LVOGetMsg(a6)
                cmpi.l #0,d0
                beq.s  Waiting

*  Check if it was a menu message  *

                move.l d0,a1
                move.l a1,Message
                move.l mg_CLASS(a1),d1
                cmpi.l #MENUPICK,d1

*  If it is then go to the menu routine  *

                beq.s  MenuHandler

*  If not then assume it is a Close Window *
*  message.  Start by replying to the      *
*  message.                                *

                move.l Message,a1
                jsr    LVOReplyMsg(a6)

*  First remove the menu strip  *

                move.l Wdw,a0
                move.l IntuitionBase,a6
                jsr    LVOClearMenuStrip(a6)

*  Close the window  *

                move.l Wdw,a0
                move.l IntuitionBase,a6
                jsr    LVOCloseWindow(a6)

*  Close the Intuition library  *

WindowFail:     move.l IntuitionBase,a1
                move.l AbsExecBase,a6
                jsr    LVOCloseLibrary(a6)

IntuitionFail:  rts

*  This routine handles user menu selections  *

MenuHandler:

*  Determine which menu item was selected  *

                move.l d0,MenNum
                lea    Menu1,a0
                move.l Message,a1
                move.w mg_CODE(a1),d0
                move.l IntuitionBase,a6
                jsr    LVOItemAddress(a6)

*  If a NULL pointer is returned then there was  *
*  no menu item selected                         *

                move.l d0,a0
                beq.s  NoMenu

*  Find the user routine to service the  *
*  particular menu item selected and     *
*  call it                               *

                move.l mi_UserRoutine(a0),a0
                jsr    (a0)

*  Now reply to the message  *

NoMenu:         move.l AbsExecBase,a6
                move.l Message,a1
                jsr    LVOReplyMsg(a6)

*  Go back and wait for the next message  *

                bra.l  Waiting

*  Process a user request to "Do This"  *
*  The routine prints a message         *

Action1:        move.l Wdw,a0
                move.l wd_RPort(a0),a0
                lea    MessageText1,a1
                clr.l  d0
                clr.l  d1
                move.l IntuitionBase,a6
                jsr    LVOPrintIText(a6)
                rts
```

```
*  Process a user request to "Do That"  *
*  The routine prints a message         *

Action2:        move.l Wdw,a0
                move.l wd_RPort(a0),a0
                lea    MessageText2,a1
                clr.l  d0
                clr.l  d1
                move.l IntuitionBase,a6
                jsr    LVOPrintIText(a6)
                rts

*  Process a user request to "Do The Other"  *
*  The routine prints a message              *

Action3:        move.l Wdw,a0
                move.l wd_RPort(a0),a0
                lea    MessageText3,a1
                clr.l  d0
                clr.l  d1
                move.l IntuitionBase,a6
                jsr    LVOPrintIText(a6)
                rts

*  In this case we do nothing if the menu  *
*  was an attribute.  The system uses the  *
*  flags field in the MenuItem structure   *
*  to automatically record the attributes  *
*  selected.  These flags can be tested    *
*  whenever required and not only when a   *
*  menu message is being processed.        *

DoNothing:      rts

*-------------------------------------------

*  Initialized Data  *

                 EVEN

IntuitionName:   dc.b 'intuition.library',0

                 EVEN

*  Define the window  *

NewWdw:          dc.w 250,60,136,26
                 dc.b -1,-1
                 dc.l CLOSEWINDOW!MENUPICK
                 dc.l $2100E
                 dc.l 0,0,WdwTitle,0,0
                 dc.w 0,0,0,0,1

WdwTitle:        dc.b 'Menus',0

                 EVEN

*Define the menu strip as a pair of menus  *

Menu1:           dc.l Menu2
                 dc.w 0,0
                 dc.w 100,10
                 dc.w MENUENABLED
                 dc.l MenuTitle1
                 dc.l Menu1Item1
                 dc.w 0,0,0,0

Menu2:           dc.l NULL
                 dc.w 100,0
                 dc.w 100,10
                 dc.w MENUENABLED
                 dc.l MenuTitle2
                 dc.l Menu2Item1
                 dc.w 0,0,0,0

MenuTitle1:     dc.b 'ACTIONS',0

                EVEN

MenuTitle2:     dc.b 'ATTRIBUTES',0

*  An IntuiText structure macro  *

IntuiTextStruc:  MACRO    ;Text No., Left, Top
                 EVEN

Text?1:          dc.b 2,3,JAM1,0
                 dc.w ?2,?3
                 dc.l 0,String?1,0

                 ENDM

                 EVEN

*  Define the individual menu items  *

Menu1Item1:   dc.l Menu1Item2
              dc.w 0,0
              dc.w 150,10
              dc.w ITEMTEXT!COMMSEQ!ITEMENABLED!HIGHCOMP
              dc.l 0
              dc.l Text11,NULL
              dc.b 'A',0
              dc.l NULL
              dc.w MENUNULL
              dc.l Action1

              IntuiTextStruc 11,6,1

              EVEN

Menu1Item2:   dc.l Menu1Item3
              dc.w 0,10
              dc.w 150,10
              dc.w ITEMTEXT!COMMSEQ!ITEMENABLED!HIGHCOMP
              dc.l 0
              dc.l Text12,NULL
              dc.b 'B',0
              dc.l NULL
              dc.w MENUNULL
              dc.l Action2
```

```
              IntuiTextStruc 12,6,1

              EVEN

Menu1Item3:   dc.l NULL
              dc.w 0,20
              dc.w 150,10
              dc.w ITEMTEXT!COMMSEQ!ITEMENABLED!HIGHCOMP
              dc.l 0
              dc.l Text13,NULL
              dc.b 'C',0
              dc.l NULL
              dc.w MENUNULL
              dc.l Action3


              IntuiTextStruc 13,6,1

              EVEN

Menu2Item1:   dc.l Menu2Item2
              dc.w 0,0
              dc.w 150,10
              dc.w ITEMTEXT!CHECKIT!ITEMENABLED!HIGHCOMP
              dc.l 2
              dc.l Text21,NULL
              dc.b 0,0
              dc.l NULL
              dc.w MENUNULL
              dc.l DoNothing

              IntuiTextStruc 21,20,1

              EVEN

Menu2Item2:   dc.l Menu2Item3
              dc.w 0,10
              dc.w 150,10
              dc.w ITEMTEXT!CHECKIT!ITEMENABLED!HIGHCOMP
              dc.l 1!4
              dc.l Text22,NULL
              dc.b 0,0
              dc.l NULL
              dc.w MENUNULL
              dc.l DoNothing

              IntuiTextStruc 22,20,1

              EVEN

Menu2Item3:   dc.l NULL
              dc.w 0,20
              dc.w 150,10
              dc.w ITEMTEXT!CHECKIT!ITEMENABLED!HIGHCOMP
              dc.l 2
              dc.l Text23,NULL
              dc.b 0,0
              dc.l NULL
              dc.w MENUNULL
              dc.l DoNothing

              IntuiTextStruc 23,20,1

              EVEN
String11:     dc.b 'Do This',0
              EVEN
String12:     dc.b 'Do That',0
              EVEN
String13:     dc.b 'Do The Other',0

                        EVEN
String21:               dc.b 'Having Cake',0
                        EVEN
String22:               dc.b 'Eating Cake',0
                        EVEN
String23:               dc.b 'Feeling Hungry',0

                        EVEN

*  IntuiText structures defining the  *
*  messages to send to the user when  *
*  an item from the "ACTIONS" menu    *
*  is selected.                       *

MessageText1:           dc.b 2,3,JAM2,0
                        dc.w 12,15
                        dc.l 0,Message1,0

MessageText2:           dc.b 3,2,JAM2,0
                        dc.w 12,15
                        dc.l 0,Message2,0

MessageText3:           dc.b 1,2,JAM2,0
                        dc.w 12,15
                        dc.l 0,Message3,0

Message1:               dc.b ' OK Done This ',0
Message2:               dc.b ' OK Done That ',0
Message3:               dc.b 'Do It Yourself',0

*  Storage for Program Generated Data  *

                        EVEN

IntuitionBase:          blk.l 1
Wdw:                    blk.l 1
MPort:                  blk.l 1
Message:                blk.l 1
MenNum:                 blk.l 1


`2
```

CLI — One Ste

Burghard-Henry Lehmann takes you to meet Ed, the CLI (Command Line Interface) editor.

■ **In the last article on CLI we've learned** how to make a disk bootable so that it starts off the Amiga from CLI instead from Workbench. In this fourth article I'd like to show you how you can write your own startup-sequence which will run a program automatically and set up the Amiga in a particular, user-defined way.

For this we have to look first at a new CLI command: **Execute.**

Execute

This series of articles is called "CLI — One Step at a Time", because at the beginning of it I said that CLI is not a programming language in the strictest sense. Most of the time you give a direct command, the computer executes it, then you give the another command, the computer executes that and so on.

But there is, nevertheless, a CLI command which lets you execute a series of CLI commands in sequence. You might say that this is a bit like a program, but you wouldn't use it to write a serious program, because CLI is really not very useful as a programming language. The execute facility only is useful for things like the startup-sequence, for which we will use it in a minute. You wouldn't use CLI to write a wordprocessor or anything as elaborate as that.

Nevertheless, the CLI command "Execute" executes a series of CLI command found under the filename given after the instruction. For example:

execute s/startup-sequence

will execute the startup-sequence on your Workbench Disk, that is in effect, start off Workbench anew.

The Startup-Sequence

The startup-sequence is always stored under the s-directory. This is where the computer expects to find it. It is a series of CLI commands which are executed immediately after the Amiga has been booted on CLI.

To understand this better, let's look at some of the commands contained in the startup-sequence you'll find on a Workbench diks To get a printout of the startup-sequence, enter:

type s/startupsequence".

First there is the simple command **Echo**. This echoes any text that is written after it and put into inverted commas on to the screen. In your own startup-sequence this is useful to show on screen the name of the disk you are starting up and the name of the program that is run automatically. You might also want to give the date the disk has been set up, as is done with the Workbench disk.

For the time being I do not want to go into most of the other commands given in the Workbench startup-sequence, because they fulfill pretty specialised functions and I will explain those in a later article on advanced CLI. But I'm sure you can recognise commands like "date" and "endcli", because I have explained these in earlier articles.

LoadWb is a CLI-command which simply loads Workbench at any given time. In the startup-sequence of the Workbench disk it is of course used to start off Workbench itself.

To run any program automatically after the computer has been started off on CLI, you simply give the usual command somewhere in your startup-sequence. That is, you either just give the name of the program itself, if you just want it to be run or you give the command "run whatever" if you want to retain CLI so that you can use it while the program is running.

Here is an example of a very simple startup-sequence which echoes a line of text, gives the date as contained in the system clock and finally runs a program called "Whatever"

ep At a Time

as a secondary CLI-process:
echo "Our New Disk Whatever Monday, the 28. November 1988" date
run whatever

Please note, you can only give the CLI commands in a startup-sequence which you first have transfered onto the new disk! Otherwise you'll get the ''Unknown command'' error report!

To transfer the CLI commands you'll need onto the new disk, first use the Makedir command to make a c-directory. Then copy the commands you'll need onto the disk, using the Copy command.

Ed — the CLI Screen Editor

How do you write your startup-sequence in practice?

As a matter of fact, you can use any wordprocessor and then save the file under ''s/startup-sequence'' on the new disk.

But you can also use Ed, the CLI Screen Editor, which is a rudimentary wordprocessor and part of CLI.

Its main advantage is that you can call it from CLI and use it to do a few editing jobs on any textfile or use it to write a short textfile, such as a startup-sequence, or a short C- or assembler program. For anything elaborate you want to use a proper wordprocessor or a proper C-editor.

To start Ed, enter "Ed filename", whereby filename is the name of any textfile you want to use Ed on or a new textfile you want to create with Ed.

Please note that Ed works of course only on proper textfiles, that is textfiles which consist of ASCII characters. This is the same as for the Type command which we have already dealt with.

Ed has a cursor which you can move with the cursor keys left and right and up and down to find your way around the textfile. Ed is always in the insert mode, that is, any character you type will be inserted at the cursor position. Ed does not have a type-over mode.

To give a command to Ed, press the ESC-key. This puts the cursor into the command line at the bottom of the screen and you can now enter one of the socalled extended commands.

For example, enter "f/word/'' and Ed will search the textfile for the occurence of "word". In short, "f'' stands for "find".
''q'' will quit Ed without saving the textfile.
''x'' will also quit Ed, but the textfile will be saved.

Figure 1 gives you a list of the extended commands which you can use.

Normally Ed is in the immediate command mode. This allows you to edit the textfile and give direct commands by depressing the CTRL key and a letter key at the same time.

For example, CTRL-A inserts a line, while CTRL-B deletes a line.

Figure 2 gives you the immediate commands used by Ed. *YA*

Figure 1
Ed — Extended Commands

A /s/	Insert line after current line
B	Move to bottom of file
BE	Block end at cursor
BF /s/	Backward find
BS	Block start at cursor
CE	Move cursor to end of line
CL	Move cursor one position left
CR	Move cursor one position right
CS	Move cursor to start of line
D	Delete current line
DB	Delete block
DC	Delete character at cursor
E /s/t/	Exchange ''s'' into ''t''
EQ /s/t/	Exchange but query first
EX	Extend right margin
F /s/	Find string ''s''
I /s/	Insert line before current line
IB	Insert copy of block
IF /s/	Insert file ''s''
J	Join current line with next
LC	Distinguish between upper and lower case in searches
M n	Move to line number ''n''
N	Move to start of next line
P	Move to start of previous line
Q	Quit without saving text
RP	Repeat until error
S	Split line at cursor
SA	Save text to file
SB	Show block on screen
SH	Show information
SL n	Set left margin
SR n	Set right margin
ST n	Set tab distance
T	Move to top of file
U	Undo changes on current line
UC	Equate upper case and lower case in searches
WB /s/	Write block to file ''s''
X	Exit, saving file

Figure 2
Ed — Immediate Commands

CTRL-A	Insert line
CTRL-B	Delete line
CTRL-D	Scroll text down
CTRL-E	Move to top or bottom of screen
CTRL-F	Flip case
CTRL-G	Repeat extended command line
CTRL-H	Delete character left of cursor (Backspace)
CTRL-I	Move cursor right to next tab position
CTRL-M	Return
CTRL-O	Delete word or space
CTRL-R	Cursor to end of previous word
CTRL-T	Cursor to start of next word
CTRL-U	Scroll text up
CTRL-V	Verify screen
CTRL-Y	Delete to end of line
CTRL-[	Escape (enter extended mode)
CTRL-]	Cursor to end or start of line

Figure 3
Ed — Special Keys

Backspace	Delete character to left of cursor
Del	Delete character at cursor
Esc	Enter extended command mode
Return	Split line at cursor and create a new line
Tab	Move cursor right to next tab position

Music On The Amiga

Armed with our knowledge of the Amiga's sound hardware presented in the first part of this series, this time around, Peter Lawrence looks at how to make use of it from AmigaBASIC and by using the exec library's device I/O functions.

■ **Having gained an overview of how the** Amiga audio hardware is put together we're now ready to look at how to use it.

The simplest way to produce audio on the Amiga is to use the **SOUND** and **WAVE** commands found in AmigaBASIC.

The **WAVE** command allows a wave table to be set up and associated with a particular audio channel. It has the form:

WAVE Channel, WaveTable

where *channel* is an integer between 0 and 3 specifying the audio channel to be used (remember that channels 0 and 3 are on the left output and channels 1 and 2 on the right). The wave table is an integer array of at least 256 elements which defines the digital waveform to use.

Alternatively the reserved word **SIN** can be used and the system will internally associate a sine wave with the channel without your needing to set up the array.

Once the wave table has been set up the **SOUND** command produces the desired audio output allowing the programmer to specify frequency in Hertz (20 — 15 000), duration (a real value between 0 and 77 where 18.2 is approximately 1 second), volume (0 — 255) and audio channel (0 — 3). Its form is,

SOUND Frequency, Duration, Volume, Channel

In addition, there are two useful variations of the **SOUND** command which allow polyphonic music to be synchronised. If a **SOUND WAIT** command is issued then all succeeding commands are placed in a queue but not executed until a **SOUND RESUME** command is issued.

The example in listing 1 demonstrates this by playing a simple series of 4 chords (C, Am, F and G for the musically inclined among you).

While this ability to synchronise multiple notes and to create user defined waveforms makes AmigaBASIC one of the more versatile dialects of Basic. As far as music is concerned, though, it is still grossly inadequate for any serious musical applications. The problem is that although the SOUND command allows the programmer to specify all three of the characteristics of a sound which we discussed in the first article (remember: amplitude, pitch and timbre) in real sounds none of these characteristics remains constant for the entire duration of the sound. As an example, when a note is played on a guitar the amplitude of the wave produced rises very rapidly to its maximum level and then gradually decreases and the note dies away. Different instruments behave differently. Similarly the timbre and even pitch of the note played will both vary over time.

Listing 2 gives an example of how this sort of thing can be handled by AmigaBASIC by splitting each note into series of consecutive notes of much smaller duration and gradually changing the characteristics of the sound as each of these shorter notes is played. In this way the program plays out a series of notes with very slow attack (amplitude rises slowly) and each of which is enhanced with a little vibrato. However, due to the inherent slowness of interpreted Basic this is about the limit. In addition polyphony is difficult to achieve using this technique, partly because queueing up so many short notes with a **SOUND WAIT** command will rapidly cause the heap to overflow and partly because executing a string of **SOUND** commands to several channels introduces a short delay in between each note which produces a rather severe tremelo effect.

So if Basic is no good to us serious users then how do we get our hands on all of this hardware from another language such as C? Read on!

The Audio Device

The audio hardware of the Amiga is treated by the operating system as a standard device and can thus be accessed by applications programs using the special device related functions included in the exec library.

Before looking at the specifics of the audio device lets spend a little time considering the generalities of Amiga device I/O.
To get a device to work for you (and this includes the keyboard, the serial port, the printer and several other Amiga devices as well as the audio device) it is necessary to set up a line of communication between the device and your application program. This is done by means of an I/O request structure. The actual form of this structure will depend upon which device you are dealing with but as a central core to all such structures is the I/ORequest structure presented below.

```
struct IORequest {
     struct      Message     *io__Message;
     struct      Device      *io__Device;
     struct      Unit        *io__Unit;
     UWORD       io__Command;
     UBYTE       io__Flags;
     BYTE        io__Error;
```

The IO structure used for the audio device is an extended form of this structure with extra, device-specific fields appended (as is the case with the other devices). The IOAudio structure looks like this.

Before any I/O can take place the relevant structure must be initialised, partly by the system and partly by the application program. The system's part in this initialisation is

```
struct IOAudio      {ORequest      *io__WriteMsg;
    struct          IORequest    ioa__Request;
    WORD            ioa__AllocKey;
    UBYTE           *ioa__Data;
    ULONG           ioa__Length;
    UWORD           ioa__Period;
    UWORD           ioa__Volume;
    UWORD           ioa__Cycles;
    struct          Message ioa__WriteMsg;
```

performed by calling the exec library function,

Error = OpenDevice(devName, unitNumber, ioRequest, flags)
D0, A0, D0, A1, D1

The function returns a zero if the device opened successfully.

In the case of the audio device the devName is "audio.device", contained in the variable **AUDIONAME** if you are programming in C. The ioRequest field points to the IOAudio structure to be initialised and, for the audio device, the other two fields are generally assigned to zero.

Before a device I/O is performed the user must create a message port and attach it to the IORequest structure by pointing the io__Message field at it. The user must also initialise the io__Command and io__Flags fields as well as any extra fields specifically related to the device in question. Once this is done the I/O is started by calling one of the appropriate exec functions such as **DoIO**, **SendIO** or **BeginIO**.

Audio Device I/O

To make use of the audio device there are three major commands to consider.

ADCMD__ALLOCATE sends a request to the system to allocate one or more audio channels for the program's use. Since, in a multitasking environment one can never sure which channels will be free this command allows a prioritised list of channel requests, the allocation map, to be passed to the system which works its way down through the list until it finds an allocation request which it can satisfy. This will depend upon whether other tasks are competing for the requested channels and upon the relative priorities which the requesting programs have assigned to their requests.

Each element in the allocation map is a single byte where bits zero to three correspond to channels zero to three. If a bit is set then that channel is requested for allocation. Thus a value of two is a request for channel one alone; a value of five is a request for channels zero and two and cannot be satisfied unless both of these channels are available. If you require a stereo pair of channels but do not mind which channels are specifically allocated then the allocation map should contain the values three, five, ten and twelve.

The allocation map is passed via a pointer in the field ioa__Data and the maps length in bytes must be loaded into ioa__Length.

The priority of the allocation request (between -128 and +127) should go into the ln__Pri in the Node structure which is rather deeply imbedded into the IOAudio structure. The field is referenced in C as ioa__Request.io__Message.mn__Node.ln__Pri.

If the allocation request cannot be satisfied and you don't want to wait around then set the **ADIOF__NOWAIT** flag in ioa__Request.io__Flags and the request will return immediately with an **IOERR__ALLOCFAILED** error if the request could not be satisfied.

ADCMD__FREE is the opposite of the allocation command. Performing this command will release the previously allocated channels back to the system and closes the audio device.

CMD__WRITE is the command used to actually play notes. To use it a second IOAudio structure should be initialised. However, this time the system's initialisation is not performed by calling **OpenDevice**, since the device is already open, but by having your program copy the contents of the ioa__AllocKey, ioa__Request.io__Unit and ioa__Request.io__Device across from the IOAudio structure used for the allocation command.

In addition to creating a MessagePort for this structure the wave table location should be placed in ioa__Data field with the tables length in ioa__Length. Volume similarly goes into the ioa__Volume field.

The ioa__Period field controls the pitch of the note produced by specifying how many system clock intervals should elapse between each sample. If this period value is multiplied by the length of the wave table then this gives the number of clock intervals per cycle of the waveform. This system clock runs at a frequency of 3579545 intervals per second and dividing the number of intervals per cycle into this figure gives the frequency of the resultant waveform in Hertz. That is,

```
Frequency = 3579545 / ioa__Period *
ioa__Length
```

If you are changing the period and volume values of the channels and wish the new values to take effect immediately then the **ADIOF__PERVOL** flag should be set in ioa__Request.io__Flags.

The duration of the note is specified via the ioa__Cycles field which tells the system how many cycles of the waveform to play. You should note here that the length of time for one cycle varies with the frequency and so a given value for ioa__Cycles will result in different durations for notes of different pitch.

As indicated earlier, the way to initiate these I/O requests is by a call to one of the exec I/O functions. For ADCMD__FREE requests then the function

```
Error = DoIO(ioRequest)
D0, A1
```

This function performs the I/O specified by the structure passed as its single argument and, when the I/O is finished, returns with a zero value in error if the request was successful.

This function does not generally suffice for the other audio commands because it affects certain flags which may result in your not getting the request which you wanted. To circumvent this an alternative function should be used. That is,

```
Error = BeginIO(ioRequest)
D0, A1
```

and functions similarly to the DoIO function except that it returns as soon as the request is initiated and does not wait for the request to complete. If your program has nothing better to do then a call to

```
Error = WaitIO(ioRequest)
D0, A1
```

will wait for the I/O to complete before waking your task and resuming execution. This is the software equivalent of twiddling the thumbs but is an efficient way to wait since it frees the processor for other things while you are waiting.

Finally, when you've finished with the audio device it should be closed using the

```
CloseDevice(ioRequest)
A1
```

assuming that this has not already been done via a **ADCMD__FREE**.

We've rushed through quite a lot of information so far and no doubt some of it is still a little unclear to you. The best way to consolidate it all is with an example.

Listing 3 presents a short program written in Lattice C which sets up a 2 byte wave table (a square wave), requests a channel, any one will do, and proceeds to play a short sequence of notes. Examine the program carefully to get a feel for how it all fits together and then have a go at modifying it or perhaps, if you're not a C programmer, try translating it into your favourite language.

Further Considerations

Auto Allocation: When the audio device is first opened it is possible to automatically allocate the desired channels without using a ADCMD__ALLOCATE IORequest. If, when the device is opened the ioa__Length field is non zero then OpenDevice assumes that the IOAudio structure passed to it has already been set up with an allocation map and so attempts the allocation.

Channel Stealing: If, after your program has had its channels allocated, another task requests one or more of these channels and specifies a higher priority than your own then these channels will be stolen from you. When an IORequest is sent to a stolen channel then the io__Unit field, which is a copy of the allocation map element representing the original, successful allocation, will have the appropriate bit reset and the request will return an AUDIO__NOALLOCATION error.

ADCMD__SETPREC: This I/O command allows a program to change the priority associated with a particular allocation without having to perform another allocation request.

ADCMD__PERVOL: allows the period and volume of a currently executing CMD__WRITE to be changed immediately thus allowing complex sequences of changes to be made without splitting the note into a series of shorter notes as was the case with the Basic example discussed earlier. If the **ADIOF__SYNCCYCLE** flag is set in io__Flags then the change will occur at the end of the currently playing cycle of the waveform.

CMD__STOP and **CMD__START**: These commands are the equivalent of the **SOUND WAIT** and **SOUND RESUME** instructions. After a **CMD__STOP** all **CMD__WRITE**s are queued until a **CMD__START** is received. Also, any notes which were playing when the **CMD__STOP** took effect will be stopped. When the **CMD__START** is issued these stopped notes will continue from the beginning of the next cycle of the wave table.

Double Buffering

A very important requirement for serious sound synthesis is the ability to change the timbre of a sound smoothly over time without getting a noticeable gap between each part of the sound as occurs when doing complex synthesis in Basic. This is achieved with a technique known as double buffering.

Two wave tables are required along with two IOAudio structures and the following sequence of events is followed.

- Set up the waveform into wave table 1.
- Issue a CMD__WRITE using this wave table and IOAudio 1 for a small number of cycles.
- Set up the next part of the waveform in wave table 2.
- Issue a similar CMD__WRITE using wave table 2 and IOAudio 2.
- Wait for the first CMD__WRITE to finish.
- Continue the waveform in wave table 1.
- Repeat the CMD__WRITE for IOAudio 1 and wave table 1.
- Wait for the second CMD__WRITE to finish.
- Repeat this sequence from step 3 until the

sound is finished.

At the end wait for the two outstanding CMD__WRITEs to complete then continue.

This technique is analogous to that we discussed in AmigaBASIC but the speed of a compiled language such as C allows it to work much more smoothly. Of course, if you need to do a lot of processing between each **CMD__WRITE** you may need to go into assembly language to get the necessary speed.

What's Next?

Well, hopefully that's given you a good basis for your own experiments with Amiga sound production although we have not by any means covered everything. In the future we'll have a look at a few more advanced techniques including directly accessing the hardware registers to improve efficiency (and all the problems that entails). However, next time we're going to take a little diversion and look at another very important aspect of music on the Amiga not yet considered, MIDI. What is it, what can it do and how can we use it? In particular, how can we save ourselves a fortune by writing out own MIDI software? It's not particularly difficult if you understand how MIDI works and, of course, after the next installment, you will. ***YA***

LISTING 1

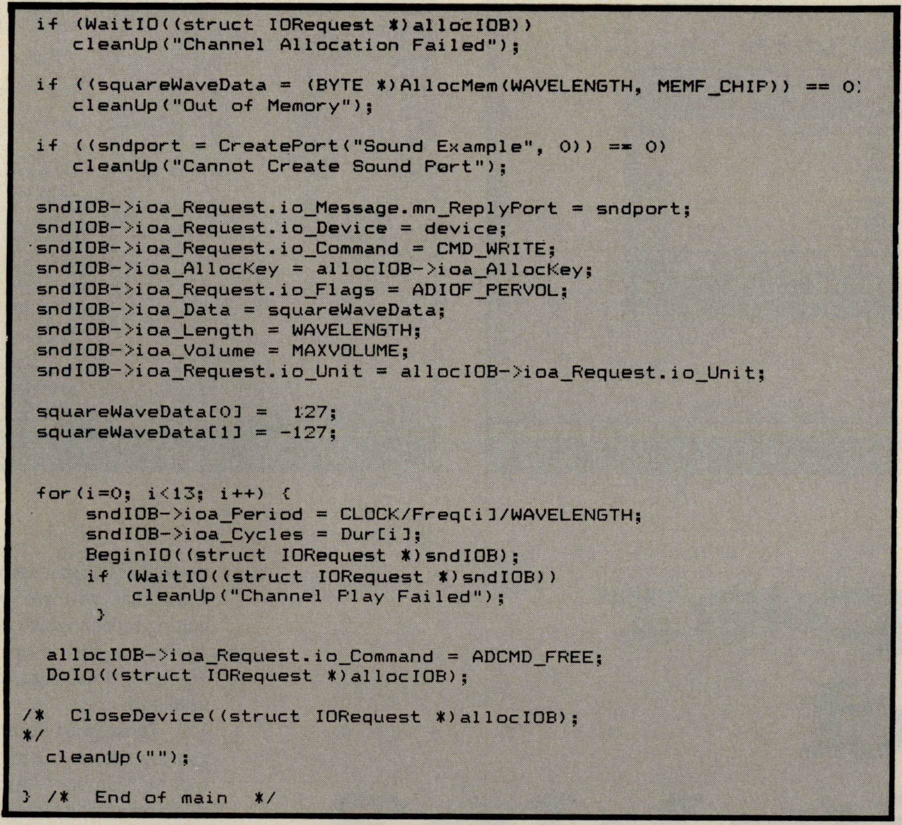
```
DIM Freq(2,3),W%(255)

'  Read the Frequency Data
'  For Each Channel.

FOR Bar = 0 TO 3
  FOR Chan = 0 TO 2
    READ Freq(Chan,Bar)
  NEXT Chan
NEXT Bar

'  Create a 256 Byte
'  Square Wave.

FOR i = 0 TO 127
  W%(i) = -127
NEXT i
FOR i = 128 TO 255
  W%(i) = 127
NEXT i

'  Assign the Square Wave to
'  Audio Channels 0, 1 & 3.

FOR Chan = 0 TO 2
  WAVE Chan,W%
NEXT Chan

'  Play the Four Chords
'  in Sequence.

FOR Bar = 0 TO 3
  SOUND WAIT
  FOR Chan = 0 TO 2
 SOUND Freq(Chan,Bar),20,255,Chan
  NEXT Chan
  SOUND RESUME
NEXT Bar

END

'  The Frequency Data

DATA 261.63,329.63,392
DATA 220,261.63,329.63
DATA 220,261.63,349.23
DATA 246.94,293.66,392
```

LISTING 2

```
WAVE 0,SIN
FOR Note = 1 TO 7
  READ x
  FOR i = 0 TO 50
    SOUND x+5*(i MOD 5),.5,5*i,0
  NEXT i
NEXT Note
'
DATA  261.63, 523.25, 493.88
DATA  392.00, 329.63, 261.63
DATA  261.63
```

LISTING 3

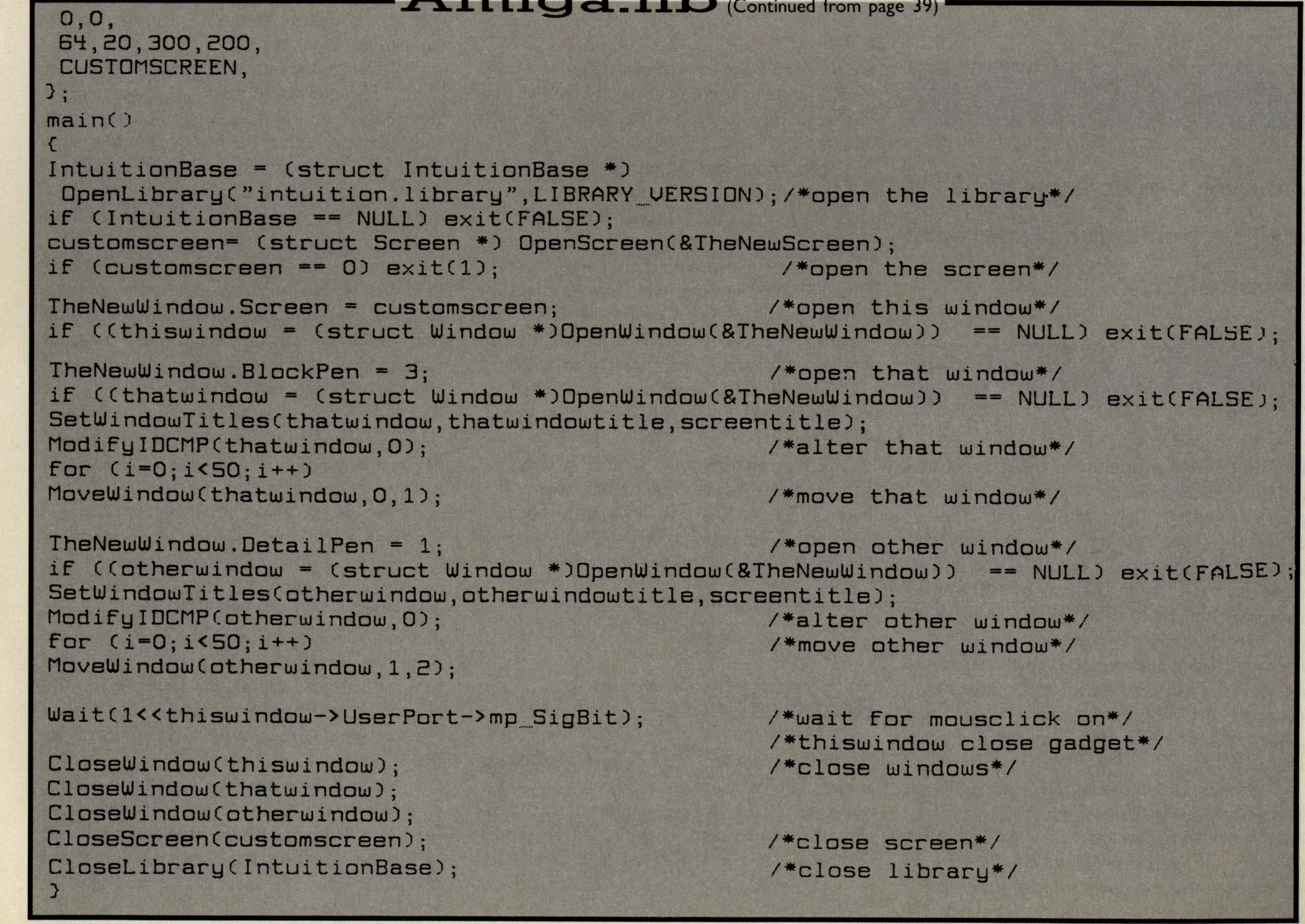
```
#include <exec/types.h>
#include <exec/memory.h>
#include <hardware/custom.h>
#include <hardware/dmabits.h>
#include <libraries/dos.h>
#include <devices/audio.h>

#include <proto/exec.h>

#define  LEFTOF     1
#define  RIGHTOF    2
#define  RIGHT1F    4
#define  LEFT1F     8

#define  WAVELENGTH    2
#define  CLOCK         3579545
#define  MAXVOLUME     64
#define  SOUNDPREC    -40

extern struct MsgPort *CreatePort();


UBYTE  allocationMap[] = {

       LEFTOF,
       LEFT1F,
       RIGHTOF,
       RIGHT1F
   };

struct IOAudio *allocIOB = 0;
struct IOAudio *sndIOB = 0;
struct Device *device = 0;
struct MsgPort *port, *sndport = 0;
BYTE *squareWaveData = 0;

UWORD  Freq[13] = {440,554,660,554,440,554,660,554,
                   440,554,660,554,440};
UWORD  Dur[13]  = {220,277,330,277,220,277,330,277,
                   220,277,330,277,880};
int i;


void cleanUp(message)
TEXT *message;

{
   puts(message);
   if (squareWaveData != 0)
      FreeMem(squareWaveData, WAVELENGTH);
   if (sndport != 0)
      DeletePort(sndport);
   if (port != 0)
      DeletePort(port);
   if (device != 0)
      CloseDevice((struct IORequest *)allocIOB);
   if (sndIOB != 0)
      FreeMem((BYTE *)sndIOB, sizeof(struct IOAudio));
   if (allocIOB != 0)
      FreeMem((BYTE *)allocIOB, sizeof(struct IOAudio));
   exit();
}  /*  End of cleanUp  */


void main()
{

  if(((allocIOB = (struct IOAudio *)AllocMem(sizeof(struct IOAudio),
        MEMF_PUBLIC | MEMF_CLEAR)) == 0) ||
        ((sndIOB = (struct IOAudio *)AllocMem(sizeof(struct IOAudio),
        MEMF_PUBLIC | MEMF_CLEAR)) == 0))
    cleanUp("Out of Memory");


  if (OpenDevice(AUDIONAME, 0, (struct IORequest *)allocIOB, 0) != 0)
     cleanUp("Cannot Open Audio Device");
  device = allocIOB->ioa_Request.io_Device;


  allocIOB->ioa_Request.io_Message.mn_Node.ln_Pri = SOUNDPREC;
  if ((port = CreatePort("Sound Example", 0)) == 0)
     cleanUp("Cannot Create Allocation Port");
  allocIOB->ioa_Request.io_Message.mn_ReplyPort = port;
  allocIOB->ioa_Request.io_Flags = ADCMD_ALLOCATE;

  allocIOB->ioa_Request.io_Flags = ADIOF_NOWAIT;
  allocIOB->ioa_Data = allocationMap;
  allocIOB->ioa_Length = sizeof(allocationMap);


  BeginIO((struct IORequest *)allocIOB);
```



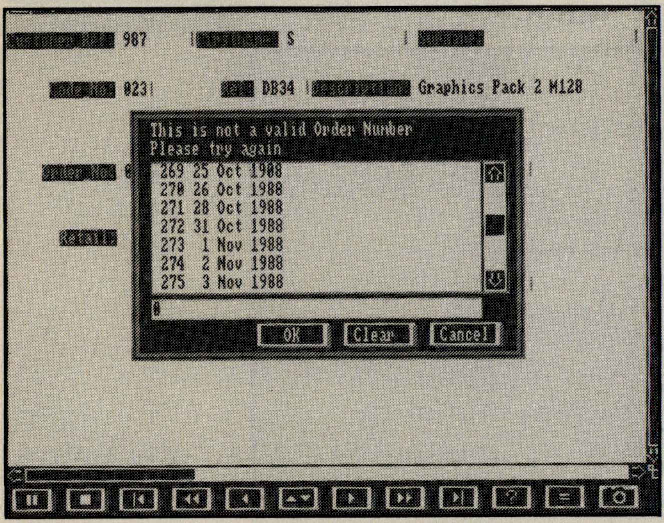
```
 if (WaitIO((struct IORequest *)allocIOB))
    cleanUp("Channel Allocation Failed");

 if ((squareWaveData = (BYTE *)AllocMem(WAVELENGTH, MEMF_CHIP)) == 0)
    cleanUp("Out of Memory");

 if ((sndport = CreatePort("Sound Example", 0)) == 0)
    cleanUp("Cannot Create Sound Port");

 sndIOB->ioa_Request.io_Message.mn_ReplyPort = sndport;
 sndIOB->ioa_Request.io_Device = device;
 sndIOB->ioa_Request.io_Command = CMD_WRITE;
 sndIOB->ioa_AllocKey = allocIOB->ioa_AllocKey;
 sndIOB->ioa_Request.io_Flags = ADIOF_PERVOL;
 sndIOB->ioa_Data = squareWaveData;
 sndIOB->ioa_Length = WAVELENGTH;
 sndIOB->ioa_Volume = MAXVOLUME;
 sndIOB->ioa_Request.io_Unit = allocIOB->ioa_Request.io_Unit;

 squareWaveData[0] =  127;
 squareWaveData[1] = -127;


 for(i=0; i<13; i++) {
     sndIOB->ioa_Period = CLOCK/Freq[i]/WAVELENGTH;
     sndIOB->ioa_Cycles = Dur[i];
     BeginIO((struct IORequest *)sndIOB);
     if (WaitIO((struct IORequest *)sndIOB))
        cleanUp("Channel Play Failed");
      }

  allocIOB->ioa_Request.io_Command = ADCMD_FREE;
  DoIO((struct IORequest *)allocIOB);

/*  CloseDevice((struct IORequest *)allocIOB);
*/
  cleanUp("");

} /*  End of main  */
```

Amiga.lib

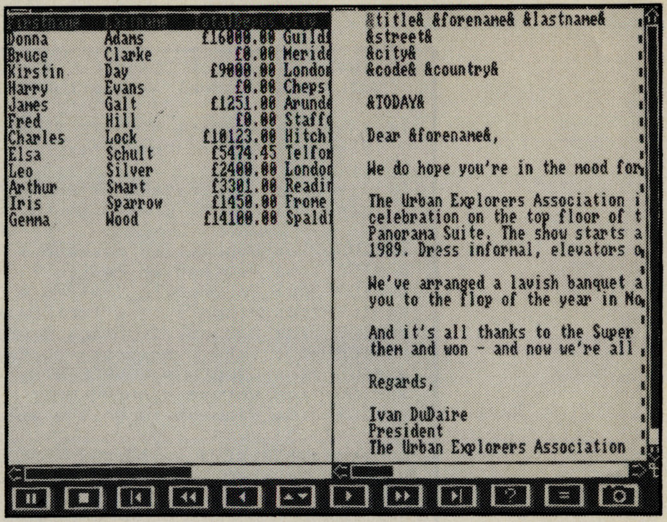
(Continued from page 39)



```
 0,0,
 64,20,300,200,
 CUSTOMSCREEN,
};
main()
{
IntuitionBase = (struct IntuitionBase *)
 OpenLibrary(”intuition.library”,LIBRARY_VERSION);/*open the library*/
if (IntuitionBase == NULL) exit(FALSE);
customscreen= (struct Screen *) OpenScreen(&TheNewScreen);
if (customscreen == 0) exit(1);                        /*open the screen*/

TheNewWindow.Screen = customscreen;                  /*open this window*/
if ((thiswindow = (struct Window *)OpenWindow(&TheNewWindow))  == NULL) exit(FALSE);

TheNewWindow.BlockPen = 3;                           /*open that window*/
if ((thatwindow = (struct Window *)OpenWindow(&TheNewWindow))  == NULL) exit(FALSE);
SetWindowTitles(thatwindow,thatwindowtitle,screentitle);
ModifyIDCMP(thatwindow,0);                           /*alter that window*/
for (i=0;i<50;i++)
MoveWindow(thatwindow,0,1);                          /*move that window*/

TheNewWindow.DetailPen = 1;                          /*open other window*/
if ((otherwindow = (struct Window *)OpenWindow(&TheNewWindow))  == NULL) exit(FALSE);
SetWindowTitles(otherwindow,otherwindowtitle,screentitle);
ModifyIDCMP(otherwindow,0);                          /*alter other window*/
for (i=0;i<50;i++)                                   /*move other window*/
MoveWindow(otherwindow,1,2);

Wait(1<<thiswindow->UserPort->mp_SigBit);            /*wait for mousclick on*/
                                                     /*thiswindow close gadget*/
CloseWindow(thiswindow);                             /*close windows*/
CloseWindow(thatwindow);
CloseWindow(otherwindow);
CloseScreen(customscreen);                           /*close screen*/
CloseLibrary(IntuitionBase);                         /*close library*/
}
```



```
987        S
023        DB34        Graphics Pack 2 M128

This is not a valid Order Number
Please try again
269 25 Oct 1988
270 26 Oct 1988
271 28 Oct 1988
272 31 Oct 1988
273  1 Nov 1988
274  2 Nov 1988
275  3 Nov 1988
0
```



```
Donna     Adams    £16000.00 Guild
Bruce     Clarke       £0.00 Merid
Kirstin   Day       £9000.00 Londo
Harry     Evans        £0.00 Cheps
James     Galt      £1251.00 Arund
Fred      Hill         £0.00 Staff
Charles   Lock     £10123.00 Hitch
Elsa      Schult    £5474.45 Telfo
Leo       Silver    £2400.00 Londo
Arthur    Smart     £3301.00 Readi
Iris      Sparrow   £1450.00 Frome
Gemma     Wood     £14100.00 Spald
```

```
&title& &forename& &lastname&
&street&
&city&
&code& &country&

&TODAY&

Dear &forename&,

We do hope you're in the mood for

The Urban Explorers Association i
celebration on the top floor of t
Panorama Suite. The show starts a
1989. Dress informal, elevators o

We've arranged a lavish banquet a
you to the flop of the year in No

And it's all thanks to the Super
then and won - and now we're all

Regards,

Ivan DuDaire
President
The Urban Explorers Association
```

Super and Suitable?

Andrew Brown tests the bold claims made for Superbase Personal 2

■ **The newest addition to the Superbase** stable, *Superbase Personal 2* comes in a jazzy grey box with a multicoloured logo splashed across the front, lots of nice pictures on the back, and plenty of bold claims to match. But what does it offer in the way of features to someone looking for a database management package? Is it a real upgrade to *Superbase Personal*, filling in the gap up to *Professional*, or is it just a cosmetic exercise? To answer that question I decided to put the program to a real world task and see if it was equal to the situation.

But first, for those people who have been living in a cave for the last two years Superbase is a database program for the Amiga (and other computers) that makes full use of the windowing environment provided by Intuition. Boasting the much vaunted 'VCR' like control panel it is claimed to be as easy to use as playing a tape. It also features the unique ability to load and display pictures and text files associated with databases.

The most significant new additions in *Personal 2* are the addition of live relational links between files using the LOOKUP function, and the inclusion of a rudimentary text editor for use with mail merge and text files. Possibly the most welcome new feature though is the addition of proper RAM disk support. Just specify the RAM: identifier when loading a file, and it will copy the datafile into the Ram disk. Then when you have finished for the day it will copy it back onto the drive it came from.

Other improvements aid the entry of data, and allow data to be manipulated as it is entered. Less useful to my mind is the addition of a communications facility, especially as Precision has also upgraded the Import/Export facility.

Practical test

For my test of its capabilities I decided to create a simple Sales Ledger system comprising three files, an address list, product descriptions, and an invoice line that hopefully would combine data from all three files.

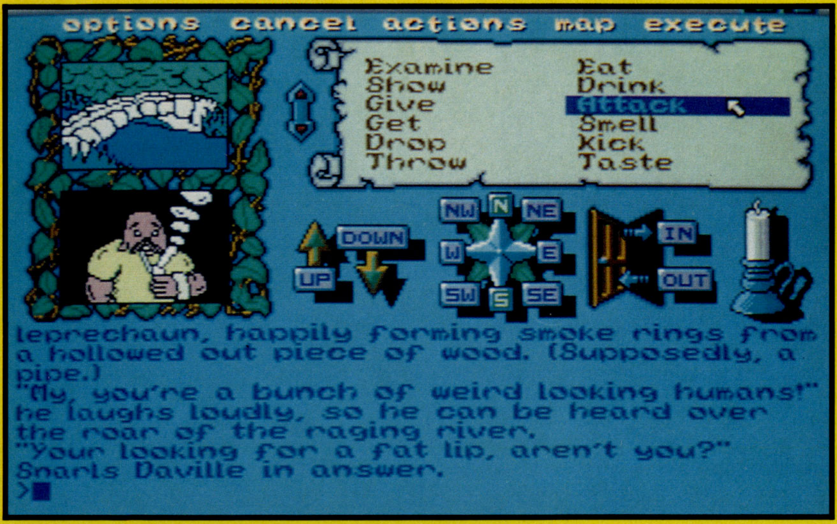
The first thing to do in each case is to specify the name of the file and then up to three passwords associated with it. Having done that you are presented with the file editing screen which allows you to specify the attributes for each field, whether text, numeric, time, date, constant, calculated, or validated.

These last two are particularly important as they allow you to create new fields from old, eg Number*Amount could be a calculation formula attached to a field called Total. Validation is even more important as it allows you to insert data from other files in a field with LOOKUP. However you may come to grief here, as the rules for using LOOKUP are very strict, and not at all well described in the somewhat terse manual.

I managed to come to grief well before that, as believing myself an experienced *Superbase* user I blasted along defining fields and formulae with thoughtless abandon. This proved to be a mistake, as when I tried to reload my saved files *Superbase* curtly informed me that it couldn't find a file or field, and then it threw me out again. No matter, I thought, I will use the Text Editor to rename all invalid field names. No way! It would allow you to load and edit the .sbd file which contains all the field definitions but as soon as you tried to save anything it just gave an out of memory error! Although the situation would presumably not arise on a 1Mbyte plus machine, this is a serious bug and should be ironed out as soon as possible. In addition to this if you use the full page gadget on the Text Editor window you can't save anything at all, and it won't go back until you quit and reboot.

To cut a long story short I managed to get my system going, but the paraphernalia of *Superbase* does not really help at all. You still need to spend plenty of time thinking about index creation, and how to link your files together. At the same time I longed for the power of *Superbase Professional*, but wished that the whole process was less cumbersome, and more intuitive.

In short *Superbase Personal 2* is the latest step in the evolution of this database from Precision Software. A more functional tool than its predecessor, it still lacks the power and flexibility to make it a full scale Database Manager. *YA*



leprechaun, happily forming smoke rings from a hollowed out piece of wood. (Supposedly, a pipe.)
"My, you're a bunch of weird looking humans!" he laughs loudly, so he can be heard over the roar of the raging river.
"Your looking for a fat lip, aren't you?" Snarls Daville in answer.
>

Fin Fahey ventures into the grim but romantic land of Anar to seek the Sword. Will the evil Suzar thwart his plans?

Legend of the Sword

On first sight, this looks very much like a roleplaying game. *Legend of the Sword* starts with you and your party on board ship, faced with a quest after... Well, something or other. The game is none too explicit about its aims. But there the resemblance to roleplaying stops — this game is really an adventure since you have no control over the other members of your party beyond the fact that they follow you everywhere like lost sheep.

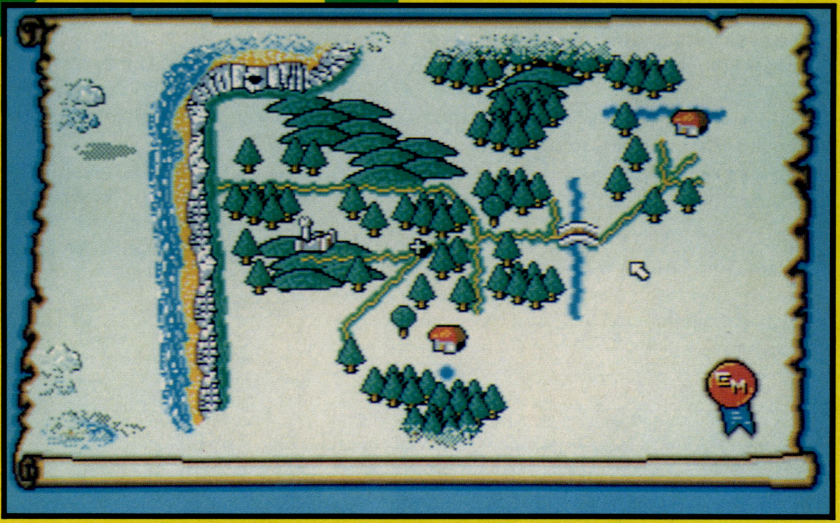
What holds the interest is that *Legend...* is one of the few adventures that does your mapmaking for you. As you progress(having by now got off the ship), more and more chunks of a prettily depicted map get filled in, saving wear on the old HB and graph paper. This led me to wonder why, with the advent of memory-heavy machines like the Amiga and that other on that starts with A, why all adventures don't do this kind of thing. Is there some kind of intrinsic virtue on scribbling maps on bits of paper? Will it profit a person in the afterlife? I think not.

That nice feature apart, this game is really a text adventure with several graphics windows depicting the map, your character's actions, the scene and so on. Very handy for anyone who is illiterate and wishes to be a spectator(actually, they do add a lot of atmosphere).

Being used to RPGs, I find the general surliness and unco-operativeness of my party a bit annoying. They're complete idiots, generally drunken to boot. If you need a simple silver coin you have to kick one of them in the face to get it. In the end, I was left wondering why they'd come.

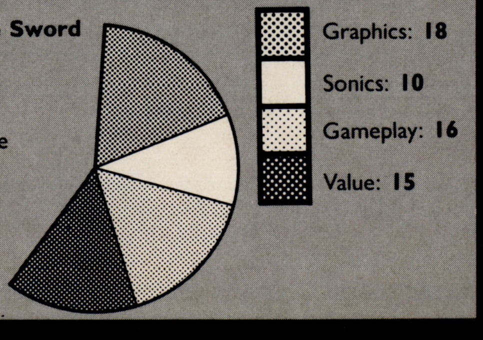
There's little enough to say about the game plot, except that it involves the usual panoply of scrolls, trolls and proles. I didn't however get very far in. The novella is of an exceptionally high standard for one accompanying a computer game — the characters sit around and tell each other stories in a sort of Canterbury Tales tavern scene. The stories are very strange — which hints at some hidden weirdness yet to come if you dig deeper. Or maybe not...

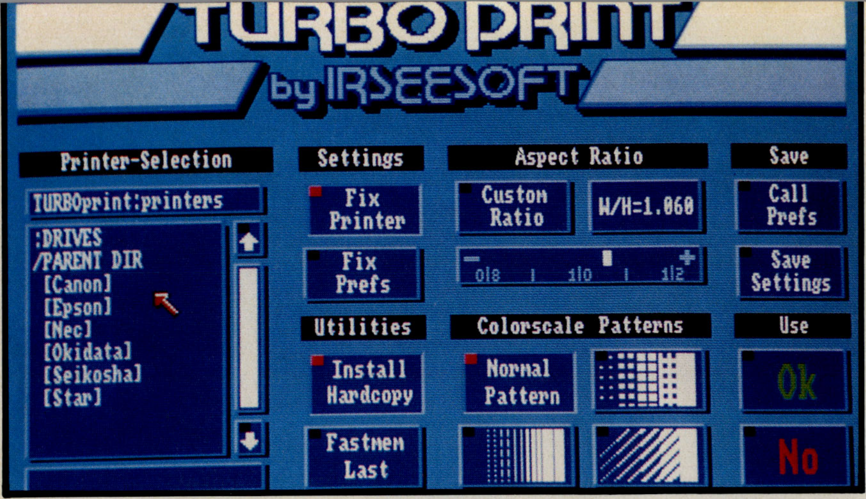
This is a competent and good-looking adventure. For some reason, possibly too much time spent playing *Ultima IV*, it didn't hold my interest. I feel it's a case of treading some well-hallowed but tired ground, albeit with mouse control. *YA*



LEGEND OF THE SWORD

Title: **Legend of the Sword**
Supplier: **Rainbird**
Wellington House
Upper St Martins Lane
London WC2H 9DL
Tel: **01-379 6755**
Price: **£24.95**





Printer Preference

Isn't the Amiga already well-endowed enough with printer support? Anne Owen explains the advantages of Preferences 1.3 and looks at a German turbo charger.

■ **I knew that some of my printouts,** especially to Canon PJ1080A and HP Laserjet could be improved upon speedwise but I started out disappointed and frustrated by this utility because the documentation did not make its operation at all clear.

A full evening's investment in experimentation and manual deciphering bore fruit however and the new printer drivers provided by TurboPrint proved fast and in fact quite simple to use. When booted, TurboPrint installs itself and remains available until you turn off the machine. A Workbench disk is then prompted for. The Turbo Preferences are set from a control panel, selectable from an icon.

Most items are merely turned on and off. The printer driver is selected from a file requestor and all selections can be saved. The settings remain even when you boot another program and you can print "snapshots" of any screen with a CTRL-ALT-P combination.

The Preferences of the application you are using, say Dpaint, will also have to be adjusted, for instance to colour. If selected, the TurboPrint driver is automatically accessed from the application.

Printer drivers

Because the software originates in Germany the "standard" printers are somewhat unfamiliar, often the "equivalents" are the names you and I would recognise.

Conclusions

It is difficult to justify the purchase of TurboPrint for improvements over Preferences, especially with 1.3 around now. The additional printer support might be just what you need of course, especially at only £25.00. The Canon colour printer for instance is supported and the driver is somewhat better than the public domain drivers I have experienced both for speed and colour representation. The Hewlett Packard Laserjet driver — hidden in the Star directory as the Laser 8 printer driver — did not improve dramatically on speed, which is what I was looking for.

Supplier: **George Thompson Associates**

Preferences 1.3

You may have been wondering what the second printer graphic screen was all about on your 1.3 Workbench. Well here goes with a brief description of what can be achieved:

Density 1-7 Some printers can print in different densities. The lower the density selection the faster the printout.

Center Image Your screen picture won't correspond exactly to your paper size so this will centre your printout, or not if you turn it off.

Scaling You have a choice of fraction (default) or integer. The latter will ensure that every pixel on the screen will be printed as an even number of dots on the printer. Good for text fonts or line drawings being graphically printed. Overrides ANY aspect ratio correction.

Color Correction Reduces the 4096 colours of an Amiga screen to a mere 3172 printable colours.

Left offset Sets how many tenths of an inch the printout will appear from the left edge of your paper.

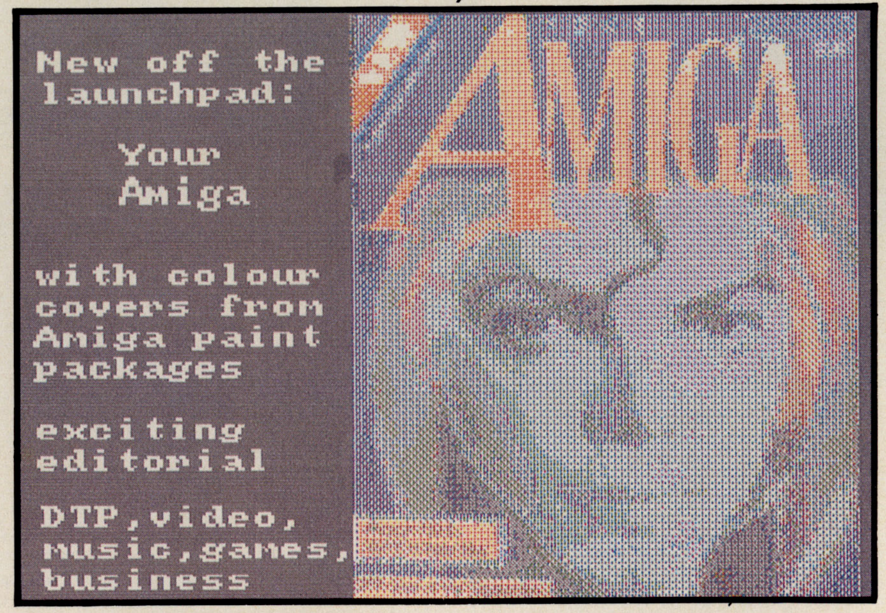
Dithering The options are ordered, halftone and Floyd- Steinberg error distribution. Wow! Give 'em a try. The halftone method is similar to that used to that used in newspapers.

Maxwidth, MaxHeight and **Limits** Set the size of your printout in tenths of an inch. **Bounded** ensures that your printout is no bigger than the limits set.

Absolute makes the printout exactly the size specified in limits. **Pixels** makes pixels the unit of measurement for the max limits. **Multiply** multiplies the pixel width and heighth by the max limits before printing.

Anti-Alias analyses each pixel and its neighbours to try and smooth diagonal lines. Useful for printing fonts and line drawings.

YA


New off the launchpad:
Your Amiga
with colour covers from Amiga paint packages
exciting editorial
DTP, video, music, games, business
AMIGA

HAM ACTING

Place those pixels where you want them

■ **One of the biggest problems when** working in Hold and Modify mode, (referred to hereafter as HAM mode) seems to be the way that, with frustrating frequency, placing a pixel of the desired colour on the screen, alters the colours of other pixels in the same horizontal line.

Although it's usually only two other pixels which are affected, I've observed what must be a knock-on effect which can change the colour of many, many more, ruining the whole 'look' of a picture. After spending much time using Photon Paint, I have discovered a number of ways to minimize this inconvenience.

As you may be aware, when operating in HAM mode, the computer has sixteen registers or memory locations, which *hold* the colour information about each of the sixteen base colours. It is these sixteen base colours which the computer then *modifies* as appropriate, to create the full Amiga palette of 4096 colours. Hence the term Hold and Modify.

When using only the base colours to paint with, I have not experienced any attribute troubles at all. The problems begin when one wishes to use more than sixteen colours, a very common occurrence after all!

If you find that after having placed a pixel on the screen, either its colour or that of its neighbours, has changed to an unacceptable degree, there are several possible courses of action which may be taken to remedy the situation:

■ Change the base colours; try to include the current ink colour or that of the pixels in one of the adjacent squares in your base colour palette using the 'affect screen' option. Experiment to find out which of the total number of colours that you plan to use, produces the most stable modified colours.

With careful planning you can reduce drastic colour problems considerably.

■ Work systematically; I've found that by working in a particular direction, left to right for example, colour problems can sometimes be totally overcome, whilst working in the opposite direction causes endless trouble. Incidentally, I actually find that I've encountered less trouble on the whole, when working from right to left.

■ Try applying the pixels in a different order; Assume for example that you wish to lay a black pixel next to a white one, but on placing the black pixel, the white one changes colour. Try to plan it out so that you lay the white pixel second rather than vice versa, this sometimes solves the problem.

■ Try, try and try again; You may find that by laying a pixel in the same spot repeatedly, or alternately relaying the adjacent pixels and the problem pixels several times, using the desired colours, that you are able to achieve the correct result. This seems to be particularly true when laying a base colour next to a modified one.

■ Can you afford to use a slightly different shade? You often find that by merely changing a colour by just one degree, that you can either eradicate the colour problems, or that the change in adjacent pixels falls to within acceptable limits.

There are numerous other 'tricks' that can make life a little easier when trying to produce your masterpiece, and the more shortcuts that you can learn, the more efficient you can become at using Photon Paint — or any other art packages for that matter! The key to discovering the best routes to achieve your objectives is experimentation. Quite often you'll create some wholly unexpected effects which can be added to your repertoire of techniques. Here are some more hints based on what I've discovered:

When you are producing a picture that includes shadows, reflections, curves, semi-opaque surfaces or any other features which require subtle shading based upon a colour spread (see Figure 1), (a range of graduated shades all based upon the same root colour), you will find it far easier to create a basic grey spread and then colour it later. For example, your picture consists of a white ball under a yellow light; therefore the ball will consist primarily of shades of white and grey tinged with yellow. Create a colour spread ranging from black or dark grey — depending on the depth of your shadows — through to white. Then bearing in mind the intensity of the light that you wish to simulate, add an equal amount of yellow colouring to each shade in your spread, using the RGB sliders. You will then have a colour spread which accurately represents the shades visible on a white ball under such lighting conditions. All that remains is to colour it in!

If, like me, you are not fortunate enough to be naturally artistic, don't be afraid to hack other people's work about to get images that you are not capable of producing yourself. I personally, couldn't draw trees to save my life, however, by looking closely at the ones that other, more artistically gifted people had produced — much of the time using magnify mode to see in detail, the colours and shading techniques that they had used — I've gained a much greater understanding of the construction of 'realistic looking' trees. Mind you, I still can't draw them, but at least I know why now!

You can also use other people's work as a source of brushes and inspiration for use in your own creations. Provided the screens aren't for public display, you won't have any problems with the copyright holders.

Public Domain software can prove a valuable source of reference material for any pictures you may be trying to create, or techniques that you are hoping to learn.

Well, there's a few of the techniques that I've learnt, but there are many, many more, just waiting to be discovered. Remember, be adventurous, it can't hurt to try something new and you may just discover something worthwile along the way.

YA

Amiga System Programmer's Guide

Phew! Stand well back if you have a delicate constitution, this is not a book for the faint hearted.

■ **The *Amiga System Programmer's*** *Guide* is volume 6 of a multi-volume series of Amiga related titles produced by Abacus Software (USA) and Data Becker (Germany). This series covers topics ranging from introductory material for first time Amiga users right through to merciless assaults on the brain cells for those seasoned hackers irrevocably hooked on mental masochism. This book is the mental masochist's nirvana. I guess that what I'm trying to say is that this is a *Techie's* book — but what is a *Techie's* book.

If you're like me, a bit of a hacker who started out writing Basic programs on an 8-bit machine and inevitably ended up moving into assembly language to minimise the speed and versatility limitations of high-level languages, then you have probably realised by now that the Amiga is no exception. Because the Amiga is hailed as such a powerful piece of hardware we end up expecting so much from it that high-level languages once more seem inadequate and the hackers all put their brains back into assembly mode.

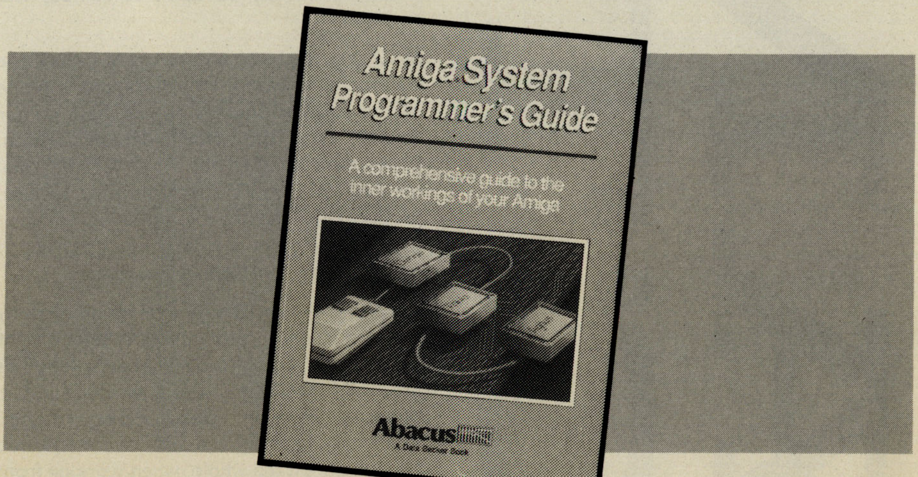
However, on an 8-bit machine, all an assembly language hacker needed was a book on programming the processor concerned and a list of special locations and operating system/ROM routine entry points. On the Amiga this is only the beginning. The fact that the Amiga has a multi-tasking operating system and is loaded with purpose built coprocessors means that the budding assembly language programmer who wants to squeeze the last bit of performance out his machine is faced with a whole new world to learn about. Enter the *Amiga System Programmer's Guide*.

This book take a detailed look at the nuts and bolts of the machine. Although it is always written with the assembly language programmer in mind its authors rarely seem prepared to skimp on detail.

It is divided into four sections the first of which is concerned with the hardware. In addition to demonstrating how to access the hardware registers directly and how to handle copper lists, interrupts and dozens of other machine-specific topics this book first gives a good and quite comprehensive overview of the hardware, right down to the pin connections on the 68000 processor, the custom chips and the peripheral interfaces and timing diagrams for various system components such as the keyboard and mouse controller. Those true Amiga devotees amongst us who realised the machine's merits from the beginning and thus sacrificed a not inconsiderable portion of their net worth to become proud Amiga 1000 owners will be pleased to learn that this book represents one of the few instances where we are not neglected. All of the variations between that wonderful history making machine and its modern, mass-produced offspring are well documented.

It must be said, however, that this book does not confine itself to merely discussing how the hardware is put together, it is liberally interspersed with example programs (available on an optional diskette to save your fingers) demonstrating the practicalities.

In part two attention is turned to the operating system with a similarly detailed account of exec, once more punctuated with helpful example programs. However, in this section much of the emphasis is shifted towards C examples which, although not unreasonable for the tasks treated, is a bit of a shame since it is hard to find good assembly language examples in some of the areas treated in C.



As an example, a common task when programming the Amiga is the creation of a message port for your program. This is, in fact, so common that the standard C library supplies a special function for this purpose and so most documentation does no more than explain how to call this function. The Amiga assemblers, on the other hand, do not have this library function so it would be nice to present a module which assembly language programmers could adapt and use. Instead the authors have chosen to present the C source code for this routine only.

Without going into too much detail about the rest of the book suffice it to say that parts three and four concern themselves with Amiga DOS and Amiga Devices respectively with the programming emphasis returning to assembly language. These two sections are considerably briefer than the first two and concern themselves much more with what the programmer needs to know rather than supplying detail merely for completeness. They are, nevertheless, informative and useful.

Overall my impression of this book is good. It presents some quite detailed and complex information in a logical and sensible manner and, while this sort of information can, to all but the most devoted hacker, seem dull but necessary the book does, I think, manage to avoid becoming a dry catalogue of facts. It nevertheless contains enough information to deserve a pride of place on your bookshelf as a long term reference. One criticism however. Its usefulness as a reference could perhaps be hindered by the fact that a volume containing so much information is supported by a mere five page index. Hardly sufficient for my money.

Overall the book impressed me although it is certainly only for the very serious Amiga programmer or the committed novice who just has to know how it all works.

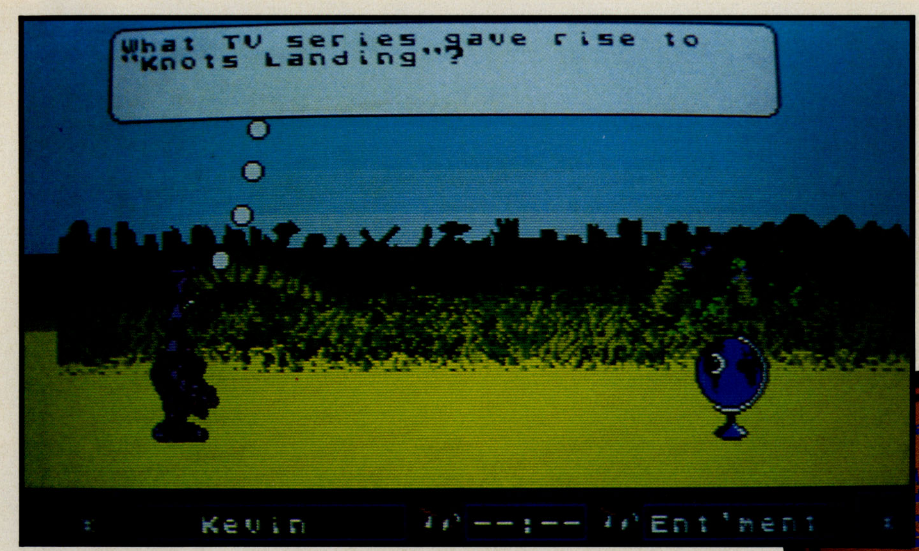
I have not yet had the opportunity to closely examine any of the other books in this series yet but I look forward to it. If they are of a similar standard to this one I am sure they will do deservedly well in the marketplace.

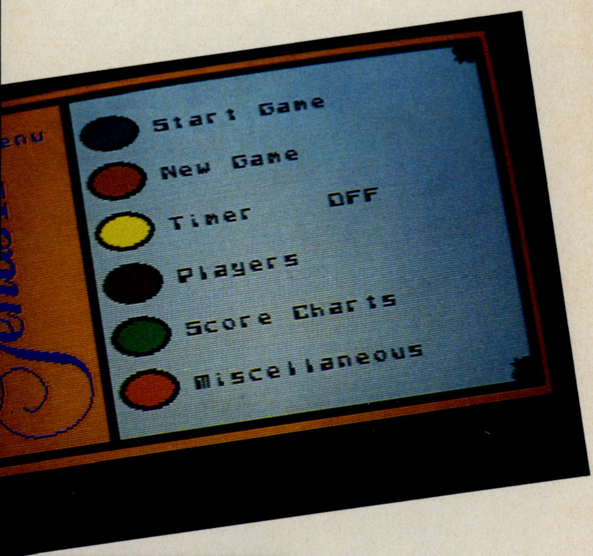
YA

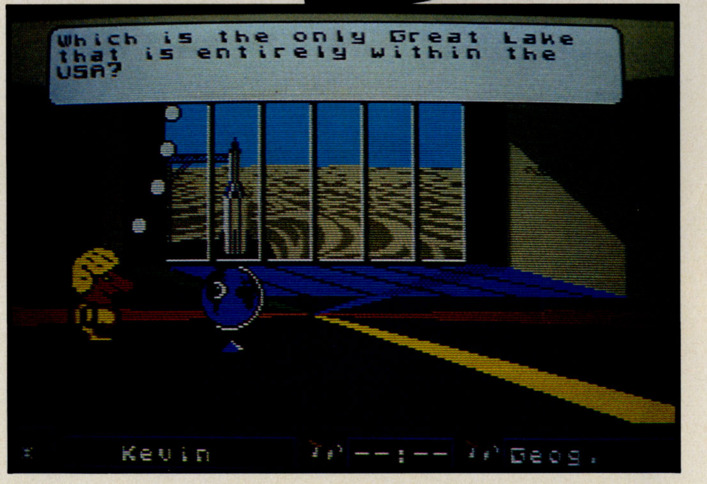
Title: **Amiga System Programmer's Guide**
Price: **TBA**
Publisher: **Abacus**
5370 52nd Street SE
Grand Rapids
MI 49508
ISBN Number:**1-55755- 034-4**

Let's get Trivial

The craze of '87 seems to have been given a new lease of life in '89 thanks to those guys at Domark — Kevin Crosby checks it out


What TV series gave rise to "Knots Landing"?




Which is the only Great Lake that is entirely within the USA?

I guess it's pretty safe to assume that everyone has been exposed to the wonders of good ol' triv by now. I'm sure we've all sat in a circle on the floor in some friend's abode till the wee small hours of the morning asking each other inane questions for the grand prize of six triangular pieces of coloured plastic in our counter tray. It proved to be the Christmas gift a year ago, and the original Domark package based on this game also did very well on most formats of micro.

Now we have *Trivial Pursuit — A New Beginning* — which attempts to turn the board game into more of a computer game than the original Domark Triv.

To do this they have put in a scenario which tells the tale of Earth in the next century as a dying planet. And you thought Triv was just answering silly questions. No, this time you're answering silly questions in order to save the human race. You must fly a spaceship to the nearest habitable planet of Genus II but to get to this wondrous world you must work your way through six galaxies finding objects which correspond to the six question topics. Once on Genus II you are treated to a grilling from the Elders who need to find out if you are indeed trivial enough to fit in with their culture. If you get through that then you've won.

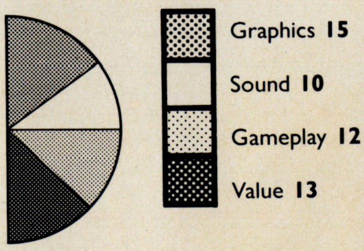
In other words, the board has been replaced by deep space, the little plastic triangles have been replaced by objects and the boring bits like rolling the die or asking the questions have been replaced by 16-bit microchip technology.

But what of the questions? Well to be honest I didn't find them too difficult. They were, in general, along the lines of "Who won the women's singles in Wimbledon '88?" or "Who runs the Fruit and Veg stall in Eastenders?" So is it any good? Technically yes; the sound and graphics certain do the job and there are plenty of features built in like player profiles, timer options, etc. I'm unsure, however, as to whether it actually works as a game. After all it does need several players and traditionally computer games have been something of a solitary pastime. I suppose it's the sort of game that will be bought for Amiga owners, not by them. *YA*

TRIVIAL PURSUIT

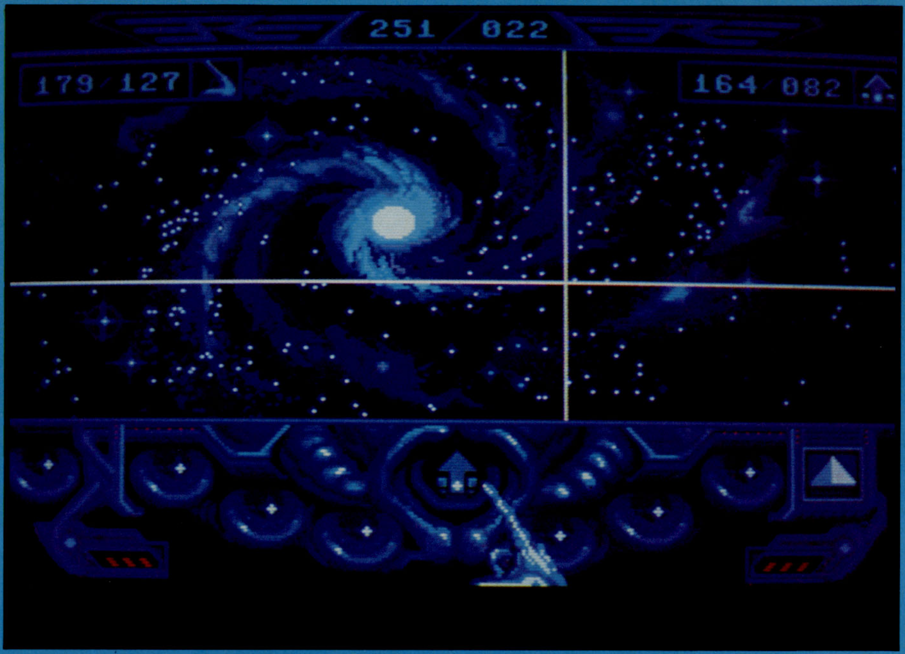
Title: **Trivial Pursuit — A New Beginning**
Supplier: **Domark**
Ferry House
51-57 Lacy Road
Putney
London SW15 1PR

Tel: **01-780 2222**
Price: **£19.95**



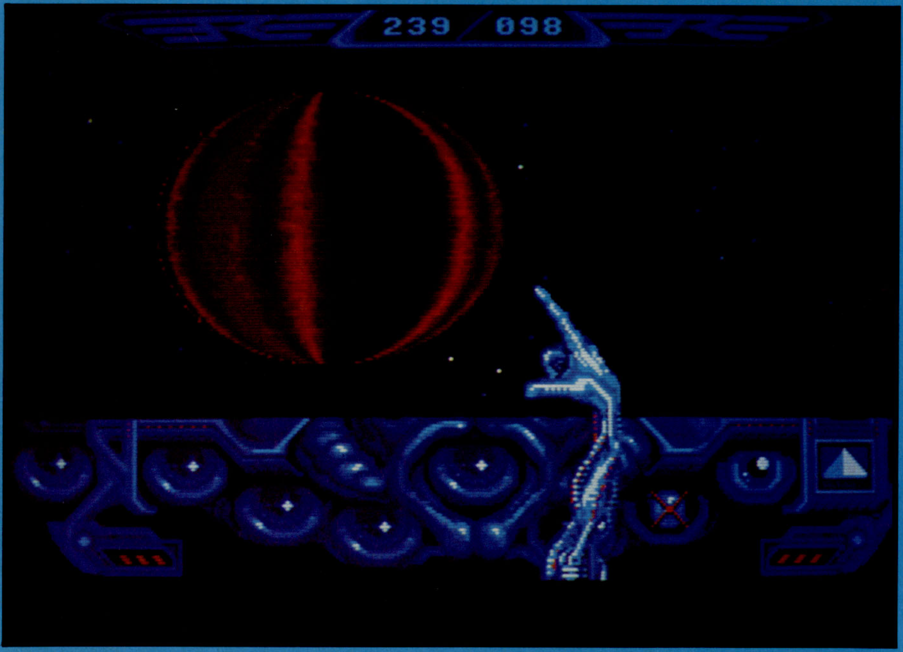
Captain Blood

It's taken a long while for this game to finally make it to the Amiga from the ST. Was it worth the wait? Kevin Crosby finds out



Not your run of the mill space game this. The program, conceived by French Software House Exxos has a very strange feel about it indeed. There's something very French about *Captain Blood*.

For a start the Novella (complimentary story generally used as padding for a weak game costing more than a tenner) does actually enhance the game. Normally the novella serves merely as a source of codewords for the copy protection. If it's not needed for that then it usually goes straight into this particular reviewers bin. However this one did have my interest for a good thirty minutes. Not because it's a noteworthy piece of literature, it isn't but it did serve as a platform for the view into this other world.



"What the Hell's Crosby talking about?", I hear you saying, "Has he been smoking back issues of TV Gamer?". I'd better tell you what the game's about. *Captain Blood* is in fact the alter-ego of science fiction writer cum games programmer Bob Morlock. To cut a long story short. He has a great idea for a game which involves flying from planet to planet looking for alien lifeforms to beam up. However just as Bob runs the program guess what! He is transmuted into the computer as *Captain Blood* — Pilot of the interstellar Ark. Obviously the guys behind this game have two videos in their collection; Alien and Tron. Anyway, whilst piloting this vessel, Blood is cloned off and, in order to survive must be turned into a sort of Bio-android who must search some 32,000 points in the Hydra galaxy for his clones or Numbers as they're referred to. It's a matter of life and death for the Captain as he must locate, teleport and destroy the Numbers to keep his vital fluid level up. No need for me to tell you what happens to Blood if he runs out of fluid!

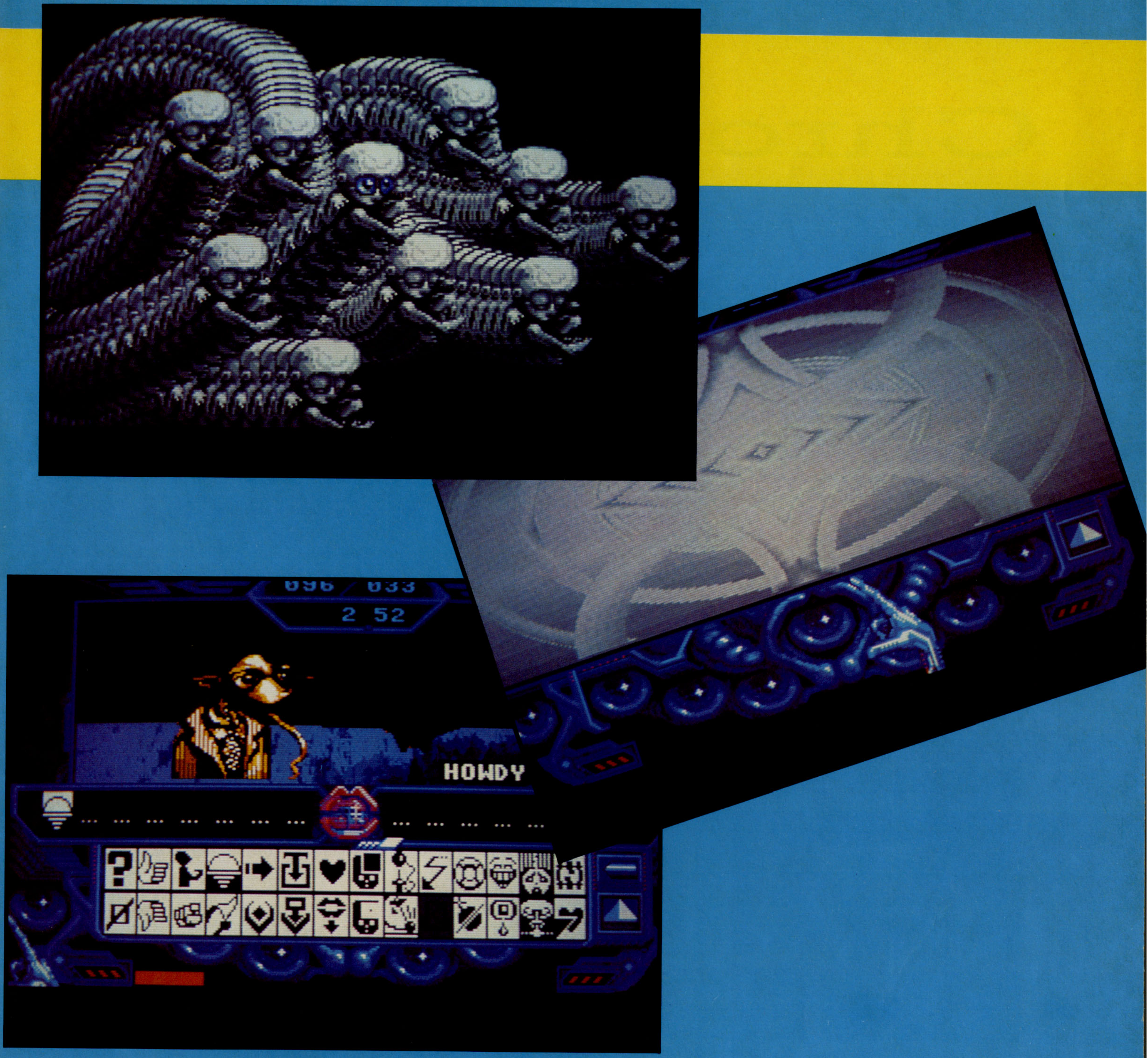
So how does one go about such a task armed only with an Amiga? Well the cockpit of your ship has over ten on-screen icons which control what views you can have of the planet you're orbiting, the Teleporter and destruction circuits therein. Galaxy Map and related Hyperdrive activation, mission selection, communication and of course good ol' Save/Load game function.

Once orbiting a planet you can select one of three missions;

1.Oorxx Geophoto mission — which involves you taking orbital pictures of the terrain below.

2.Oorxx Destroy mission — which involves simply nuking an entire planet. And finally...

3.Oorxx Contact mission — which involves sending one of these living extensions of yourself down onto the planet, to skim the surface in search of



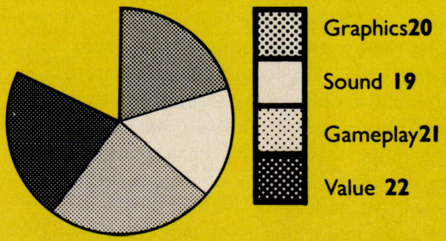
contactable life of some description which can be teleported aboard and moved elsewhere in return for some information on the whereabouts of your Numbers. If on the planet you do find one of the aforementioned clones you need to teleport him up and then disintegrate him in order to gain some life fluid.

The difference here is that creatures have to be willing to be teleported in order for the process to work, so a little friendly coaxing may be required to get what you want. This is achieved with the aid of the UPCOM (Universal Protocol of Communication), an icon-based language which all life forms (even Spectrum owners) can understand.

Makes a pleasant change to see a game appear that doesn't purely rely on total destruction to win. Combine this with some excellent sound and graphics plus a theme tune from Jean Michel Jarre and you've got yourself one hell of a game, which should keep the slightly more intelligent starship troopers among you happy for a good long time. YA

CAPTAIN BLOOD

Title:**Captain Blood**
Supplier:**Infogrames**
Price:**£24.95**

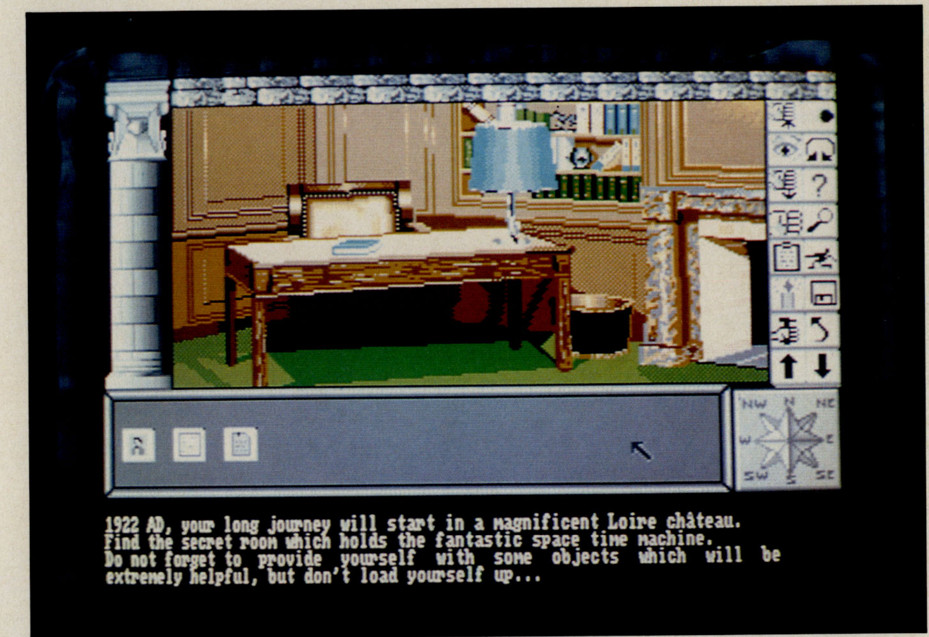


Chrono Quest

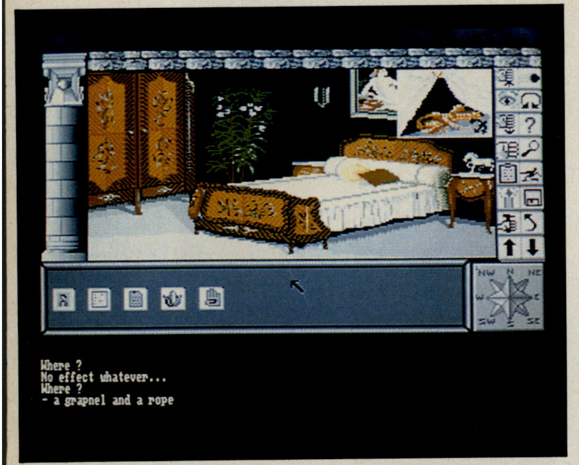
Gordon Hamlett journeys back through time in this latest title from Psygnosis

■ **Psygnosis has established a fair** reputation in the sixteen bit software market with games like *Barbarian and Obliterator*. There new game may come as something as a shock to their many fans and for a variety of reasons.

To start with, *Chrono Quest* is not an arcade game but an adventure. There's nothing wrong with that in itself but when you couple that with the fact that the game was actually written by a third party French company, then problems start to arise.



```
1922 AD, your long journey will start in a magnificent Loire château.
Find the secret room which holds the fantastic space time machine.
Do not forget to provide yourself with some objects which will be
extremely helpful, but don't load yourself up...
```



```
Where ?
No effect whatever...
Where ?
- a grapnel and a rope
```

The story is a familiar one that has already been explored several times in other adventures. The year is 1922. Your father has managed to discover a means of travelling through the very fabric of time itself by means of a machine that he has invented. During the course of his researches though, he has made several enemies and now he has disappeared. To make matters worse, you are being accused of his murder and so set out to prove your innocence and expose the real villain.

The game is entirely controlled by icons. Whereas this should simplify matters considerably, it only serves to make matters worse. Although there is limited text, it is in Franglais rather than English. There are several spelling mistakes and many of the responses are downright misleading — that's if you are lucky enough to get one in the first place, the game is very inconsistent here.

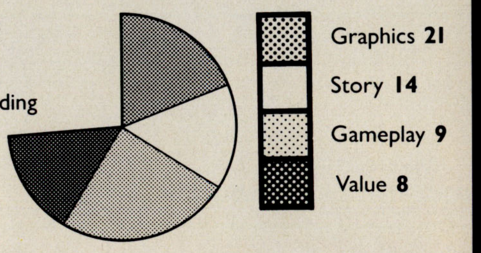
Because of the system of icons, the number of things that you actually do is severely limited. Most of the puzzles require little more than using the correct item in the correct location. Again, this is fair enough if the system actually works. In practice, locating a specific item on the screen is a hit and miss affair. For some items, you have to click on exactly the correct spot, other times, there is a large area of tolerance built in. For example, in the study, clicking on some books gets no response, others inform you that you are looking at a small library whilst some appear to have powers of transmutation as you end up examining a silk lampshade which is somewhat below the books.

Again, frustration sets in as you cannot manipulate items as you would like to. You cannot pick up objects that are eminently portable, even if they prove to be of no use to you. Opening a cupboard with a key results in you being unable to shut that cupboard again — until you move into a different location whereupon you find the door shut of its own accord when you return. Dropping an item becomes a total lottery — you may or may not find it again.

Not everything about the game is bad though. The graphics look excellent and unfortunately are likely to seduce the unwary purchaser. Great graphics do not a good game make. The game itself though is dreadful and represents very poor value at thirty pounds. Give this one a miss. They may not look as good, but there are many better - and cheaper — adventures around. *YA*

CHRONO QUEST

Title: **Chrono Quest**
Supplier: **Psygnosis**
1st floor,
Port of Liverpool Building
Pier Head,
Liverpool
L3 1BY
Tel: **051 207 0825**



YOUR AMIGA

01-437 0699
extension number 281

CLASSIFIED ADVERTISEMENTS

Lineage 58p per word (including VAT) — Minimum £8.70
Semi-Display £10.80 (plus VAT) — Minimum 3×1

Send your requirements to:
Anna Eddie, Classified Department,
Your Amiga, ASP LTD, Golden Square,
London W1R 3AB

MAIL ORDER

AMIGA SOFTWARE CLUB

WIDE RANGE OF SOFTWARE AVAILABLE

Try before you buy. Public Domain programmes available from £2.00.

Send SAE for free details to:

P.O. BOX 155,
Cardiff CF5 4YA

PUBLIC DOMINION SOFTWARE

PUBLIC DOMAIN SOFTWARE

400 DISKS AVAILABLE FOR THE AMIGA

Fish 1-162, Amicus 1-26, Slipped Disk 1-40, Fang Hotmix 1-60 **£3.00 each.** Panorama 1-72, Auge 1-25, APDL 1-40 **£4.00 each.** Please write or phone for details.

THE AMIGA PD LIBRARY,
140 Rushdale Road,
Sheffield S8 9QE
Tel: 0742 588429

MAKE YOURS A HAPPY AND PROSPEROUS NEW YEAR — ADVERTISE IN YOUR AMIGA

REPAIRS

Fixed price repairs for Amiga 500's door to door service.

All approved modifications included free of charge.

Just £60 + VAT for door to door convenience.

GRANADA COMPUTER SERVICES,

Wilkinson House,
Telspace 2, Stafford Park,
Telford, Shropshire TF3 3BD

Tel: 0952 290488 — Ask for administration VISA or cheque

REPAIRS

**AMIGA, ATARI, etc.
Monitors and all add-ons.
Amiga upgrades and Mods**

ALL ESTIMATES FREE
PRICES FROM ONLY £25
Call or post. *Fast turnaround*
Carriage arranged

PVS ELECTRICAL COMPONENTS
244 Deansgate Court
Deansgate, Manchester
M3 4BQ
Tel: 061-831 7086
Telex: 668986
Fax: 061-832 6934

pvs

DISKS

ATHENE DISKS

85p EACH + VAT (98p)

Units of 10 Minimum Quantity 10
DS/DD 135tpc

Unbranded bulk discs by Maxell, Verbatim or KAO. Excellent quality. Full 2 for 1 guarantee.

Disc boxes, branded discs, Amiga A500, 2000
Please call for quantity discounts

**0705 511439 (24hr)
0705 511646 (Fax)**

**Cheques/PO to ATHENE DISKS,
16 Stoke Road, Gosport, Hants PO12 1JB**

SPECIAL OFFERS

PSD COMPUTER SUPPLIES SCOTLAND'S NEWEST AND BEST

FREE! FREE! FREE! FREE!

**HIGH QUALITY MOUSE MATS
FOR PURCHASES OVER £25
OR THREE OR MORE BOXES OF DISKS
All disks supplied with labels and lifetime guarantee**

Box of 10, 3.5″ Branded......	£9.90
Box of 10, 3.5″ Branded	
Sony.............................	£18.50
Verbatim	£17.50
Dysan	£18.66
3M	£19,65

80 × 3.5″ Disk Box	£7.49
Mouse Mats	£4.00
9.5 × 11 Printer Paper, 2,000 sheet......................	£13.95
Also labels and 5.25″ disks	

All prices include VAT — Phone for bulk orders
75p p&p for orders less than £15

PSD SUPPLIES
32 Peat Place, Nitshill,
Glasgow G53 6SJ Tel: 041-881 5947

COPY DEADLINES APRIL/MAY ISSUE DEADLINE: 7th FEBRUARY

**Telephone details to
Anna Eddie on 01-437 0626
or Fax your copy on 01-437 1006**

VIDEO GRAPHICS

WE SPECIALISE IN GRAPHICS, ANIMATION AND DTP SOLUTIONS

BUROCARE

Burocare Computer Systems
211 Kenton Road, Harrow,
Middlesex
Tel: 01-907 3636 Fax: 01-907 0027

TRIANGLE TELEVISION

TRIANGLE TELEVISION
The Amiga Graphics and Animation Specialists

FOR ALL YOUR VIDEO AND 'A.V.' REQUIREMENTS

Slide/Print/OHP Bureau Service. All systems are available and tested on our High Band Edit Suite. Ask about the Triangle Broadcast Genlock.

TRIANGLE TELEVISION
130 Brookwood Rd
London SW18 5DD
Tel: 01-874 3418 Fax: 01-877 1029

SPECIAL OFFERS

T.C. COMPUTERS

TEL: Wantage 02357 60177

AMIGA A500 + TV modulator + software	£349
AMIGA A500 10845 + software	£599
AMIGA A500 + High Res Colour Monitor	£648
AMIGA A500 + £200 of free software + joystick	£399
AMIGA A500 + 2nd 1 mb disk driver software	£439
COMMODORE PC1 - mono+software	£345

Also Atari ST & PC Compatibles including Amstrad available. Enquiries welcome.

Please send SAE for details. Now at:

12a Barnards Way, Charlton Heights,
Wantage, Oxon OX12 7EB

DISKS

Why Take Risks?...
Buy **Branded** disks.

GOLDSTAR 3½" DS/DD DISKS
Boxed in 10s, 100% error free, Branded, Fully Guaranteed, Includes Labels

£8.30 + VAT Per box of 10

PLEASE ADD 87p + VAT FOR P&P
Goods sent same day as order received.

P.V. TUBES (MAIL ORDER DEPT).
104 Abbey St., Accrington, Lancs,
BB5 1EE
Tel: (0254) 36521/32611/390936
Fax: (0254) 395361
PARTNERS: Mrs S. Cucknell/
Mr B. Cucknell.

CLASSIFIED COUPON

YOUR AMIGA, CLASSIFIED ADVERTISEMENT DEPT
No 1 GOLDEN SQUARE, LONDON W1R 3AB

PLEASE DEBIT MY ACCESS/BARCLAYCARD No. □□□□□□□□□□□□□□□□ EXPIRY DATE □□□□

Rates: Lineage 58p per word (VAT inclusive) minimum £8.70 semi-display £10.80 per single column cm. No reimbursement for cancellations. All ads must be pre-paid.

PLEASE USE BLOCK CAPITALS and include Post Code.

Name ....

Address ....

....

.... Daytime Tel. No. ....

Signature .... Date ....

□ For Sale □ Software □ Wanted □ Equipment □ Aerials □ Otherwise State

DEALER GUIDE

WE ACCEPT
ACCESS AND VISA
TELEPHONE BOOKING

TO ADVERTISE WRITE TO

Classified Dept
1 GOLDEN SQUARE
LONDON W1R 3AB

BERKSHIRE

CSE COMPUTERS

HORSESHOE PARK
PANGBOURNE
RG8 7JW
Tel: (07357) 4774

Contact Mr John Arkill

THE AMIGA SPECALISTS

HERTFORDSHIRE

ASK THE EXPERTS...

- ★ We stock a comprehensive range of Amiga Hardware, Software & Accessories
- ★ Wide selection of peripherals
- ★ Guaranteed media at competitive prices
- ★ Ruby-Comm Viewdate & VT100 Terminal software. The Complete Comms Pack for your Amiga or ST

Y2 COMPUTING
111 St Albans Rd, Watford WD1 1RD
Tel: 0923 50161

LONDON

AMIGA A2000 SPECIALIST DEALERS

Video and interactive video programmers and consultants

ARIADNE SOFTWARE LTD
273 Kensal Road,
W10 5DB
Tel: 01-960 0203
Contact: C West

SUSSEX

MAN
COMPUTER MACHINE
MDR INTERFACES LTD

CONTACT US FOR YOUR
AMIGA 2000 and 500
SOFTWARE AND HARDWARE
EXPERT ADVICE GIVEN
ALSO BESPOKE SOFTWARE

Little Bridge House, Danehill,
Haywards Heath, Sussex RH17 7JD
Telephone: Danehill (0825) 790294

EAST SUSSEX

GAMER PERSONAL COMPUTERS

71 EAST STREET
BRIGHTON BN1 1HQ
TEL 0273 728681

Full range of Amiga Hardware and Software in Stock

WEST YORKSHIRE

TRILOGIC

STOCKISTS OF
- ● COMMODORE
- ● PHILIPS
- ● STAR
- ● CITIZEN
- ● SBC

Full range available at cash & carry prices from our new showroom at:

UNIT 1, 253 NEW WORKS RD,
BRADFORD BD12 0QP
Tel: 0274 691115.

See our ad elsewhere in this issue for Amiga accessories

TO ADVERTISE IN YOUR AMIGA DEALER GUIDE SEND YOUR REQUIREMENTS TO

ANNA EDDIE
YOUR AMIGA
1 GOLDEN SQUARE
LONDON W1R 3AB
Tel: 01-437 0626

FREE ISSUES

A subscription to your favourite magazine is the best way of making sure you never miss an issue.

And from now until 28th February 1989 you can get extra copies ABSOLUTELY FREE, by taking advantage of our special Christmas subscription offer. With a monthly title for example, this means you get 15 issues for the usual price of 12.

Order your subscription today using the coupon below and you will receive the best in reading entertainment right into the 1990's! This offer is also open to subscribers wishing to extend/renew their current subscriptions.

Standard subscription rates ▶ ▶ ▶

Monthly titles (15 for the price of 12)	UK	Europe	Middle East	Far East	Rest of World
A & B Computing	£18.00	£27.30	£27.60	£31.10	£28.30
Aeromodeller	£23.40	£28.20	£28.40	£30.20	£28.70
Antique Clocks	£27.00	£32.40	£32.60	£34.70	£33.00
Citizens Band	£16.80	£20.70	£20.85	£22.35	£21.10
Electronics Today International	£18.00	£22.20	£22.40	£24.00	£22.70
Ham Radio Today	£16.80	£21.30	£21.50	£23.20	£21.80
Military Modelling	£16.80	£23.60	£23.85	£26.45	£24.30
Model Boats	£16.80	£21.20	£21.30	£23.00	£21.60
Model Railways	£15.00	£20.90	£21.10	£23.30	£21.50
Photography	£15.00	£22.00	£22.30	£25.00	£22.80
Photoplay	£13.20	£17.90	£18.10	£19.90	£18.40
Popular Crafts	£18.00	£23.30	£23.50	£25.40	£23.80
Radio Control Model Cars	£16.20	£21.10	£21.30	£23.10	£21.60
RCM&E	£15.60	£21.60	£21.80	£24.00	£22.20
Radio Modeller	£15.60	£21.20	£21.40	£23.60	£21.80
Scale Models International	£16.20	£20.80	£21.00	£22.70	£21.30
Video Today	£15.00	£20.20	£20.40	£22.30	£20.70
Which Video?	£15.00	£19.40	£19.50	£21.20	£19.80
Woodworker	£16.80	£24.00	£24.20	£26.90	£24.70
Your Commodore	£15.60	£23.25	£23.50	£26.40	£24.00
Quarterly titles (5 for the price of 4)					
Practical Wargamer	£ 7.80	£ 9.90	£10.00	£10.70	£10.10
Radio Control Scale Aircraft	£ 9.00	£11.10	£11.20	£12.00	£11.30
Alternate monthly titles (8 for the price of 6)					
Commodore Disk User	£15.00	£18.00	£18.20	£19.30	£18.40
Radio Control Boat Modeller	£ 8.10	£10.70	£10.80	£11.70	£10.90
Your Amiga	£ 9.00	£11.80	£11.90	£13.00	£12.10
Fortnightly title (28 for the price of 24)					
Model Engineer	£28.80	£37.80	£38.00	£41.50	£38.70

Please commence my subscription to .... with the .... issue.

I enclose my cheque/money order for £.............................. made payable to ARGUS SPECIALIST PUBLICATIONS

or debit my Access/Barclaycard number □□□□□□□□□□□□□□□□ Card expiry date ....

Signature .... Name ....

Address ....

.... Postcode ....

Please return this coupon with your remittance to:

Infonet Ltd. (YAFI/1) 5 River Park Estate, Billet Lane, BERKHAMSTED, Herts. HP4 1HL

ANCO

AMIGA

ATARI ST

COMMODORE 64

AMIGA	£14.95
ATARI ST	£14.95
COMMODORE 64	£ 9.95 (Cass)
	£12.95 (Disk)
SPECTRUM	£ 9.95
AMSTRAD	£ 9.95
MSX	£ 9.95
IBM PC EGA-CGA	£19.95 (3¼ & 5¼ same box)

ANCO SOFTWARE LTD, UNITS 9-10, BURNHAM TRADING ESTATE, OFF LAWSON ROAD, DARTFORD, KENT DA1 5BH TEL: 0322 92513