

All Electron User  
programs also work  
on the BBC Micro with  
OS 1.2 and Basic II

A Database Publication

# electron

## user

Vol.2 No.6 March 1985 £1

How to fill  
those awkward  
shapes—fast!



### GAMES!

Chilling arcade  
action... deadly  
race against  
time... fun  
with words for  
early learners

### UTILITIES!

More tips for easy  
animation... creating giant  
letters... printing graphics displays

Acorn's Plus 3  
First facts about the official  
Electron disc system

# TAKE CONTROL!

WITH THE NEW COMMANDER 3  
JOYSTICK INTERFACE FROM  
BUD COMPUTERS.

**COMMANDER 3**  
Joystick INTERFACE

## FEATURES:

- PCGO standard Joystick memory position
- Includes Software Utility for maximum games compatibility
- Custom made case
- Fully compatible with Auto-Fire Joysticks
- 18 months guarantee



GREAT  
VALUE  
£14.95

Available from all good computer retailers or direct, post free, from Bud Computers.

Please send me:  Commander 3(s) at £14.95     Quickshot II Joystick(s) at £9.95  
All prices include VAT and Postage & Packing

Name \_\_\_\_\_

Address \_\_\_\_\_

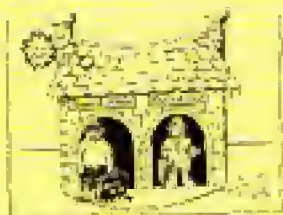
Tel. \_\_\_\_\_

Bud Computers Ltd., FREEPOST, 196 Milburn Road, Ashington, Northumberland NE63 1BR  
Trade and export enquiries welcome.

**Bud**  
Computers  
AHEAD WITH IDEAS

## News

All that's new in the expanding world of the Electron. **5**



## Percy

Beat the burning fuse and save the house. **18**

## Screeendump

Two easy ways of outputting screen pictures on a printer. **20**

## Notebook

A simple program simply explained. **23**



## Filler

A machine code fill routine to colour in those awkward shapes. **24**

## Hardware

A look at the Slomo VDU controller. **29**

## Scrapbook

The pages where Electron Users share their short, simple, fun routines. **30**

## Acorn's Plus 3

We take a first look at the official disc system for the Electron. **34**



## Merlin's Cave

At last - the first of Merlin's columns. **35**

## Triangle Turnover

Lines in action **36**

## Software Surgery

All you want to know about the latest in software from our frank reviewers. **37**



## Fred's Word Game

Learning becomes fun as you help Fred with his sentences. **41**

## Big Letters

Brighten up your programs with our large letter utility. **44**

## Beginners

If you don't know about logical variables THEN read this. **8**



## Pascal

An in-depth look at the language, its uses and Acornsoft's S-PASCAL. **11**

## Animation

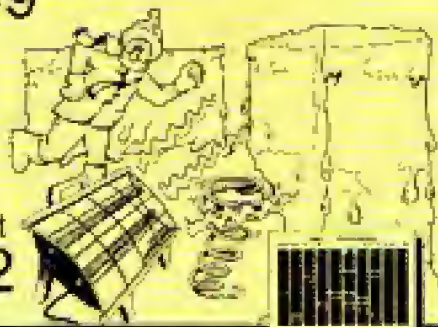
The final article in our series on Basic graphics action. **13**

## Micro Messages

The pages you write yourself. A selection from our mailbag. **49**

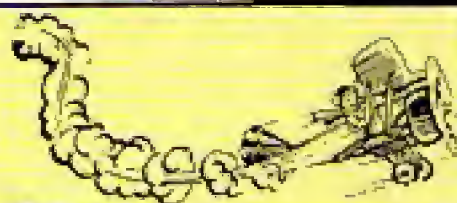
## Mr. Freeze

Arcade ice cube action to get you hot under the collar. **52**



## Guidelines

Rules of thumb for those trying their hand at Electron programming. **57**

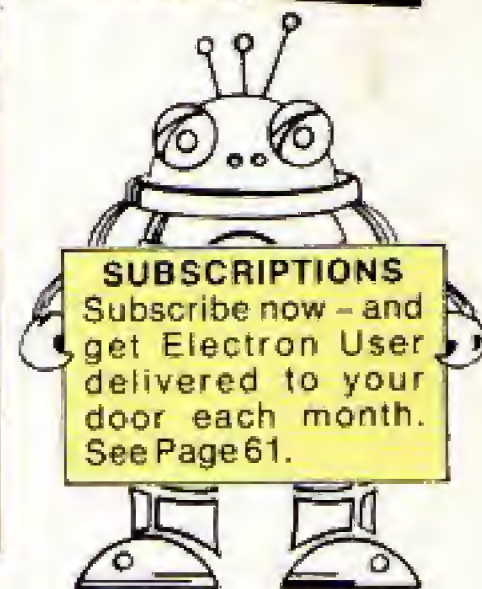


## Pigs

Even pigs might fly in Electron User. **58**

## Picasso

Modern art comes to your micro. **59**



**SUBSCRIPTIONS**  
Subscribe now - and get Electron User delivered to your door each month. See Page 61.

Managing Editor  
**Derek Meakin**  
Features Editor  
**Pete Bibby**  
Production Editor  
**Peter Glover**  
Layout Design  
**Heather Sheldrick**  
Advertisement Manager  
**John Riding**  
Advertising Sales  
**John Snowden**  
Editor in Chief,  
Database Publications  
**Peter Brameld**

**Published by Database Publications Ltd**  
Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.  
Telephone: 061-456 8835 (Editorial) 061-456 8383 (Administration) 061-456 8500 (Advertising)  
Subscriptions: 061-480 0171 Telex: 857664 SHARET G. Prestel: 614568383.

News trade distribution: Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Circulation 0293 27053.

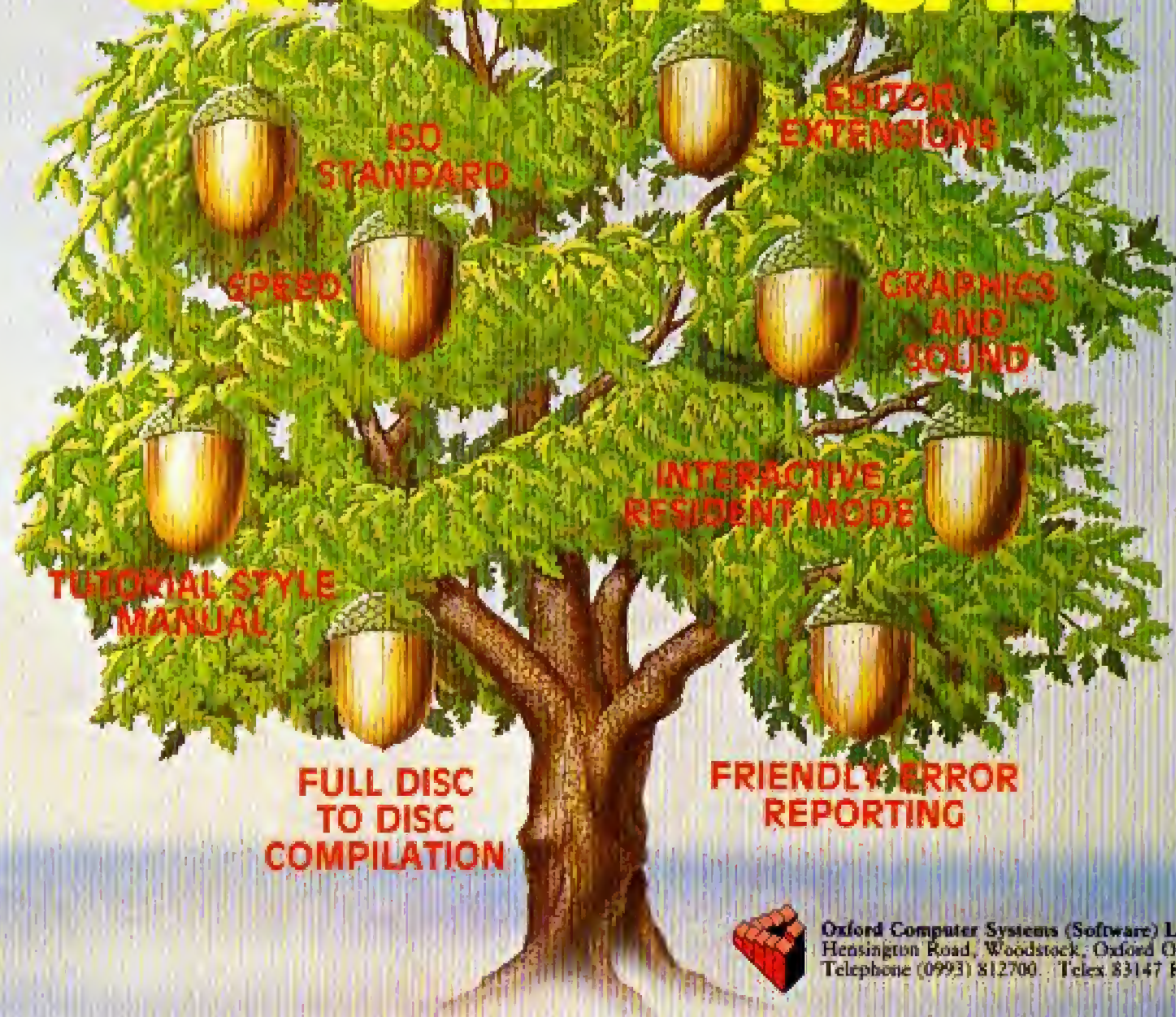
Electron User is an independent publication. Acorn Computers Ltd, manufacturers of the Electron, are not responsible for any of the articles in this issue or for any of the opinions expressed.

Electron User welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise, the return of material cannot be guaranteed. Contributions accepted for publication will be on an all-rights basis.

Subscription rates for 12 issues, post free:  
£12 UK  
£13 Eire IIR (£16)  
£20 Europe  
£20 Rest of world (surface)  
£40 Rest of world (airmail)

© 1985 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles or listings.

# OXFORD PASCAL



Oxford Computer Systems (Software) Ltd.  
Heasington Road, Woodstock, Oxford OX7 1JR, England  
Telephone (0993) 812700 Telex 83147 Ref. OCSL

## Compilers like these don't grow on trees

### Oxford Pascal is Fast

Oxford Pascal compiles down to FAST COMPACT P-code, giving you the real speed and power of Pascal, together with the ability to compile very large programs.

### Oxford Pascal is Standard

Oxford Pascal is a full extended implementation of Standard ISO Pascal. This means that you can compile any Pascal program (subject to size), written on any computer, anywhere.

### Oxford Pascal is Compact

Because it compiles into P-code, Oxford Pascal reduces programs into the most compact form possible. In fact it allows you to pack more code into your BEEB than any other language, and should your programs become too large, you can still use the CHAIN command to overlay limitless additional programs without losing data.

### Graphics & Sound Extensions

In addition to the entire Pascal language, Oxford Pascal features a whole range of Graphics (all modes) and sound extensions designed to make maximum use of the BBC Computer. Oxford Pascal also provides numerous extensions such as hexadecimal arithmetic and bit manipulation instructions.

### Oxford Pascal in Education

In Education, Oxford Pascal is fast becoming a *de facto* standard. It is already the most popular Pascal on the Commodore 64, and will soon be released for the Spectrum and the Amstrad. In fact, Oxford Pascal will soon be available for 90% of the computers installed in the U.K., and is already available in German, French, Swedish, and American versions. Students and teachers alike find that it makes sense to use a standard implementation of Pascal across the whole range of educational micros. Call us for details of our generous educational discounts.

### Manual

Both these compilers come with a manual which has been carefully designed, not only as a quick reference guide, but also as a full

tutorial for those new to Pascal.

### Resident and Disc Compiler

Oxford Pascal comes in two forms:

**For Tape Users...** Oxford Resident Pascal. A compiler located largely in ROM which is available at any time. Programs can be written and compiled on the spot without disc or tape access, and compilation is fast enough to make using the compiler much like using the BASIC interpreter. Thus, learning Pascal is a simple interactive process. Some 15K of memory is available for user programs, the remainder being reserved for compiled object code.

**For Disc Users...** Oxford Disc Pascal offers all the above PLUS... a full disc compiler which is capable of using the WHOLE memory for Pascal object code, it is supplied with a powerful LINKER, allowing you to break large programming tasks down into separately compilable, easily-manageable files.

### Friendly Error Messages

Many compilers produce little more than an error and line number to help correct mistakes in Pascal programs. Oxford Pascal however, gives you one of 49 friendly and informative error messages. Messages which not only indicate the reason for an error, but also print out the line in question with a pointer to the exact position where the error was detected. Run-time errors are reported using line-numbers from the original source-program, with a full explanation of how the error occurred.

### Powerful Editor

With Oxford Pascal there is no need for you to learn how to use a new Editor. Pascal programs can be entered in exactly the same way as BASIC programs, without the need to learn any new commands. When you are used to using Pascal, you will find our extensions to the Standard

Editor even more useful. What is more, Oxford Pascal allows you to mix BASIC and Pascal together, in much the same way that you can mix BASIC and assembler. In fact you can, if required, mix all three together...BASIC, Pascal and assembler...in one program.

### Stand Alone Code

Unlike other compilers, Oxford Disc Pascal allows you to compile on the BBC and then relocate your program so that it will run on the BBC and on the Electron. The relocated program will run without a Pascal ROM and can be loaded and run from tape or disc just like any other program.

This means that you can distribute or sell your software freely and without the need for ROMs, to run either of the above machines.

Price/availability matrix

	BBC 'B'	ELECTRON	C64	SPECTRUM
DISC	£49.95	Not vend.	£49.95	Available April 1st 1985
CASSETTE	£39.95	£59.95 Inc. Cartridge	£22.95	

All prices are inclusive of VAT  
Please add £2.00 for postage  
and packing

### Oxford Compilers — The Future

During the next year, we at Oxford will be releasing a series of language implementations such as C, and Modular 2, for the BBC, and other popular micros.

These compilers are being built, using the most modern techniques in automated compiler construction, and will bring to the micro-user, a level of robustness and efficiency, only now becoming available to mini and mainframe users.

Oxford...  
the Compiler  
Compilers.

Oxford Pascal Order form. Please make cheques payable to OCS Ltd.  
Please return the my copy of Oxford Pascal. I enclose £  
including £2 postage and packing.  
My computer is BBC 'B'  ELECTRON  C64  (please tick appropriate)  
I would like my compiler supplied on DISC  CASSETTE

Name \_\_\_\_\_  
Address \_\_\_\_\_  
Postcode \_\_\_\_\_  
Telephone \_\_\_\_\_

# electron user NEWS

## Acorn slashes price of Electron by £70

A RECORD boom in sales of the Electron has enabled Acorn to slash the price of the machine by a dramatic £70.

This effectively reduces the cost from £199 to £129 and is expected to prove a shot in the arm to sustain the current high level of sales.

The price cut follows news that 100,000 Electrons had been sold in the lead up to Christmas.

In all the Electron sold a total of 190,000 units during 1984, as compared to 230,000 for its stablemate, the BBC Micro.

Now it is confidently

predicted that Electron sales will easily outstrip those of Acorn's once undisputed flagship during 1985.

Announcing the price cut, Acorn's managing director, Chris Curry, said: "We also expect a big increase in Electron sales to education.

### Attractive

"With the much lower price, the addition of Econet later this year, and lots of new edu-

cational software coming along, it now becomes a very attractive proposition for schools".

Curry was quick to reply to a report in The Sunday Times that the BBC Micro was "sinking into oblivion" and quoted major High Street chains who said the machine was their top selling computer.

He said The Sunday Times story – the culmination of a series of



inaccurate and damaging reports – had begun to resemble a vendetta against the British computer industry.

And he strongly criticised the paper's assumption that Japanese domination of the British micro industry was assured.

He said: "Delivering these self-inflicted wounds can have no other result than to give our foreign competitors an advantage".

*IN an appeal to schools Chris Curry said: "It is essential you support suppliers in stamping out software piracy. If it is not significantly reduced, we are in serious danger of wiping suppliers out.*

*"The result will be less software for schools, and what there is, of a poorer quality. Nobody will win".*

## SAY IT WITH ELECTRON FLOWERS!

ELECTRON technology will soon be helping people say it with flowers. A new machine from British Telecom's business systems equipment division – the Merlin M2105 – has been built around the Electron motherboard.

Interflora, the association of flower retailers, plans to place one of these terminals in each of its 2,500 UK member's

shops. It will replace the telephone as a means of transmitting orders around the country.

A six month pilot scheme involving 50 Interflora members will take place this year, after which the association hopes all its outlets will go on-line with the new system.

The Merlin M2105 is a small desk top machine with

built-in modem, auto dial and answer, VDU, dot matrix printer and associated software.

### Response

It uses the T-Link communications network protocol with file transfer between terminals over the public telephone network.

Not user programmable, it

has a limited voice response library, powerful built-in text editing facilities and text messaging and communications software as standard.

In another pilot scheme, under the auspices of the National Health Service, the Merlin M2105 is being used in the Healthnet message service within a local area health authority.

# Now Mini Office is teaching aid

MINI Office, the chart topping business software package recently released for the Electron by Database Software, has been selected as a national teaching aid.

It will now be incorporated as part of a series of special courses held throughout the UK to link education with industry.

The training program is organised by the Careers Research and Advisory Centre (CRAC) for sixth form students, undergraduates and

careers and business studies teachers.

CRAC is a registered charity funded by most of the UK's blue chip companies ranging from the Abbey National through to Marks and Spencer and Williams & Glyn's Bank.

"Our brief is, to increase understanding of business enterprise, the role of management and the kind of skills required", says Maureen Curson, CRAC's course manager.

"So we are very interested in Mini Office

to help get our message over".

Mini Office is a professionally written suite of four programs – a word processor, spreadsheet, database and graphics – which converts an Electron into an inexpensive office tool.

It comes with a 32 page operating guide which acts as an easy to understand tutorial.

However it is the revolutionary pricing of the package – just £5.95 – which has guaranteed it being a runaway success. For business software packages often carry price tags of several hundred pounds.

But CRAC has something else in mind for Mini Office apart from being a valuable teaching aid.

"We also intend to use it to help streamline our own office", Maureen Curson told *Electron User*.



## Crystal prize

ELECTRON users who buy Bourne Educational Software's award-winning Osprey program from Boots have a chance of owning a beautiful crystal sculpture.

The competition is split into two sections, with prizes for the best individual and group project folders based on the history of the Osprey and its return to Scotland.

Alternatively entrants can produce a study of a local bird of prey compared to the Osprey.

The first prize in each section is a specially commissioned Osprey in lead crystal by Swedish sculptor Mats Jonasson.

There will be 10 runner-up prizes of copies of *The Book of British Birds* produced by the Royal Society for the Protection of Birds. Closing date for entries is June 30.

## Upgrade service

A NEW loader program adopted for Acornsoft games has meant that the only joystick interface that will work with these products is Acorn's own.

But Power Software says it has found

a way round the problem for owners of its Electron joystick interface.

Power is offering a free upgrade service for end users that will enable the joystick to be used with Acornsoft's latest releases.

# Does your micro go bump in the night?

IT seems that Electron computers – along with ghouls and ghosties – may be among the things that go bump in the night.

A scientific body which normally investigates strange phenomena ranging from the Loch Ness monster to UFOs has turned its attention to the machines.

Roger Morgan of the Association for the Scientific Study of Anomalous Phenomena (ASSAP) has written to *Electron User* for help with his research.

"Can I appeal to readers for any information, at first or second

hand, no matter how bizarre, concerning unexplainable malfunction or unexpected output", he asks.

Contacted at his London home, he explained: "We are looking for things like strange messages suddenly appearing on screens".

ASSAP, founded three years ago, has some 300 members across the country who devote much of their spare time to serious investigation of the par-

anormal and related fields.

It was recently called in to investigate reports of hauntings at Marylebone magistrates court and has developed an infra red video recorder to assist in its work.

Why has ASSAP suddenly become interested in computers?

"We feel they are a valid subject in the light of the fact we have collected some very interesting data from things run on electricity", says Roger

Morgan.

As a town planner, he regularly works with a computer and this has led him to believe there is a possibility that the machines may lend themselves to acting as mediums.

Secretary of ASSAP is Dr Hugh Pincott who also believes computers may well act as vehicles for psychic phenomena.

"A particular interest of mine is regressive hypnosis where people reveal what apparently happened to them in

past lives", he says.

"Now one of the areas under investigation is the possibility of a cosmic database.

"Of course there may be nothing in it. But we have had enough reports to suggest that it is a valid subject for scientific research.

"And we believe that somewhat more ordinary computers may fall into the same category".

Is there anyone out there – whether Electron user or even the computer itself – who can help? If so please contact Roger Morgan, 15a Kensington Court Gardens, London W8 5QF.

# Education isn't getting a fair deal - Kosmos

**SOFTWARE** publisher Keith Spence says Electron users who want to buy educational programs aren't getting a square deal from retailers and distributors.

Spence, managing director of Educational software house Kosmos, says many shopkeepers are deliberately misleading customers about the availability of programs for the Electron.

Now he has started a campaign to persuade computer dealers and wholesalers to support educational software for the machine that has become the number three best selling micro.

The move was prompted by a letter to Spence from user Sylvia Powers of Birmingham saying: "I have been most disappointed and annoyed to find such little software in the

shops for the Electron.

"We are told by shops that the Electron is not a popular machine so they do not stock the software".

## Problem

Spence, who says he gets dozens of similar letters and phone calls each week, countered angrily: "The problem of unavailability of educational software in the shops is one we've been trying to combat for years.

"Basically, shops - including the High Street multiples - don't like using their precious shelf space for educational programs. They prefer to fill the shelves with games because games sell in greater

quantities and bring in more profit.

"The problem is compounded by similar views held by most software wholesalers. This means that even if shops want to stock educational software they will more often than not have to buy it direct from the software houses.

"They would prefer not to do this because it means more accounts, more invoices, more cheques to issue.

"Shops give the most amazing excuses for having no educational software in stock. Often their stories are a complete fabrication or distortion of the truth.

"This is illustrated by the fact Mrs Powers was

told the Electron is not a popular machine, which is of course nonsense.

"The Electron is still the best machine available for anyone who wants something more for their children than games".

Spence advises Electron users: "Decide what you want to buy and ask your local shop

to obtain it. Don't accept any excuses and don't take "no" for an answer.

"If you have no success or if you prefer a faster response, contact the software house direct.

"Meanwhile we shall continue battling to get educational software for the Electron on the shelves", he added.

## Micro medical course

CHANNEL 4's "The Living Body" series has inspired Martech to produce a software package based on the weekly science programmes.

Six programs, together with a 32 page full colour booklet produced by a leading medical science publisher, will offer Electron owners the opportunity to become more familiar with the workings of

their body.

Titles are "Getting to know your insides", "Building a blood system", "Heart operation", "So you think you can breathe?", "Adventure in digestion" and "Keeping going". Price for the complete package is £19.95.

## Where IS the Plus2?

**MYSTERY** surrounds the Plus 2, the "missing link" in the Electron expansion story.

The first unit to be released was the logically named Plus 1, which allowed the Electron to use Centronics printers, ROM cartridges and also provided an analogue to digital port.

Now Acorn has released its disc system for the Electron, the impressive Plus 3. But as yet there is no sign of the Plus 2 - and Acorn isn't giving any information about it.

There is some speculation that the missing link may be the promised RS423 inter-

face. This telecommunications port is boasted of in the Plus 1 Help message even though it isn't part of the Plus 1.

However informed opinion feels that this will still be part of the Plus 1 and that the Plus 2 will be some other interface.

Now the mystery may be solved. In the Plus 3 user guide both the tube - a high speed data link used with second processors - and the Econet networking interface are mentioned. Neither yet exist for the Electron. Could these make up the missing link? Acorn's reaction?

"No comment".



**THESE** are the new 3 1/2 in and single and dual 5 1/4 in disc drives to be launched by Cumana for the Electron.

The interface will cost £149.95, the 100k 3 1/2 in drive with interface £299.95 and the 100k 5 1/4 in drive with interface £289.95.

## Challenge on TV

JUDO expert and TV Superstars champion Brian Jacks has challenged Electron users to eight testing events - arm dips, squat thrusts, canoeing, cycling, football, swimming, archery and the 100 metres.

Brian, who holds the world record for arm dips of 54 in 30

seconds is appearing regularly on BBC-TV's "Micro Live".

The new software release, Superstar Challenge, from Martech will be featured on the programme on March 8 and 9.

The game requires a strategic approach as well as quick reactions.

# Logical variables really work —but only on condition

LAST month saw us braving the mysteries of IF ... THEN statements. We saw how we could use them together with the relational or logical operators in Table 1 to get our programs to make choices.

Using these we could make the outcome of a program depend on whatever was input into it. We saw that the IF ... THEN statement took the form:

IF condition is true THEN perform the rest of the line.

If the condition wasn't true then the rest of the line was ignored. The Electron, albeit dependent on the program, had shown it was capable of making choices determined by the program's data.

Program 1 sums up what we covered last month, using all six logical operators to subject six numbers from a DATA statement to a battery of IF ...

```
10 REM PROGRAM 1
20 MODE 1
30 FOR loop=1 TO 6
40 READ test
50 PRINT "Loop ";loop "
test is ";test
60 IF test=1 THEN PRINT
test*=1"
70 IF test<4 THEN PRINT
test*<4"
80 IF test>3 THEN PRINT
test*>3"
90 IF test<=5 THEN PRINT
test*<=5"
100 IF test>=2 THEN PRINT
test*>=2"
110 IF test<>6 THEN PRINT
test*<>6"
120 NEXT loop
130 DATA 3,-2,1,6,4,0
```

Program 1

THEN tests. See if you can understand what's happening.

Why are some lines printed and others are not?

Now that you've reminded yourself of what you learnt last month, take a good look at Program 11. Can you see anything that appears wrong?

```
10 REM PROGRAM 11
20 FOR loop=1 TO 3
30 READ test
40 PRINT "Loop number ";
loop
50 IF test<5 PRINT "less
than five
60 NEXT
70 DATA 6,7,3
```

Program 11

Surely there should be a pair of inverted commas at the end of line 50? Before you correct that mistake however, try running the program as it stands.

Amazingly, it works correctly for two cycles round the loop and only then gives the error message.

Can you explain this strange behaviour?

It's all down to the IF ... THEN statement of line 50. The first time round the loop, test is equal to 6. This means that the condition test<5 is false as 6 is certainly not less than 5.

As the condition is false the Electron doesn't bother looking at the rest of the line but goes onto line 60. In other words it skips over the part containing the mistake. So no error message occurs.

The same is true when test is equal to 7. The condition at the beginning of line 50 is false so, again, the program ignores the rest of the line after the THEN.

Only when test is 3, the third time round the loop, is the condition test<5 true. The

Operator	Meaning
=	equals
<	less than
>	greater than
<>	not equal to
<=	less than or equal to
>=	greater than or equal to

Table 1: Logical operators

micro goes on to obey the rest of the line after the THEN and finds the error, with the resultant error message.

The program works perfectly until the condition is eventually fulfilled and the Electron has access to the part of the line with the error.

This is an important point to bear in mind when you have a program that works at some times and not at others.

The mistake could be lurking behind a THEN, only coming out to play when the conditions are right.

Errors like these, ones that only happen intermittently, can be very hard to spot and correct.

Let's take another look at the form of the IF ... THEN statement.

IF condition is true THEN perform the rest of the line.

Notice that only if the condition is true is the rest of the line performed. If the

condition is false the rest of the line is ignored. So the condition can be either true or false.

When the Electron comes across conditions such as:

age>18

or

price>=cost+cost/10

in IF ... THEN statements, it has to check to see if they are true or false. If age is 20 then the first condition is true. If age is 5 then, obviously, it is false.

Now the Electron, being a computer, doesn't use the words true and false, it uses numbers to record the outcome of a test.

If a condition holds good, the Electron notes it with a -1. If the condition is false it notes it with a 0.

To the Electron there are no conditions that are true or false, just conditions that return either 0 or -1 when the Electron is told to evaluate them.

This can seem a little weird



in theory so, as ever, try it out on your Electron. Enter:

```
age=25
```

and then:

```
IF age>18 PRINT "older than 18"
```

and the message should appear on the screen. The condition `age>18` is true, so the rest of the line after the THEN is performed.

Now try:

```
PRINT age>18
```

and you'll see your micro proudly displaying `-1`. If this surprises you, don't worry, it's quite simple really.

As you already know, if you tell your Electron to

```
PRINT 5+5
```

it will come back with the answer 10. It evaluates the expression `5+5` before it displays the result. Similarly with:

```
PRINT age>18
```

The micro evaluates this conditional expression before displaying the answer. And, since you've already decided that `age` is 25, then the condition `age>18` is true (as 25 is greater than 18) and so `-1` is returned.

Now enter:

```
age=7
```

If you type in:

```
PRINT age>18
```

you'll find that the micro prints out 0, as the condition is now false.

You can even use conditional expressions like these to give values to variables. Try entering:

```
variable=age>18
```

into the Electron. It may look odd, but the micro will accept it. Now, when you type in:

```
PRINT variable
```

you should get the number 0. The Electron has evaluated the conditional expression, found the result to be 0 (as 7 is not greater than 18) and assigned that value to the variable `variable`.

As conditional expressions can only take two values, either 0 or `-1`, this may seem a little pointless, but these logical variables, as they are called, do have their uses as

we'll see later on.

And if you have problems remembering which is true and which is false the Electron will come to your aid.

It has two special variable names TRUE and FALSE. TRUE is always `-1` while FALSE is always 0. Try entering:

```
PRINT TRUE
```

and

```
PRINT FALSE
```

into the Electron and you'll get the values `-1` and 0.

For the meantime however, let's leave logical variables. They're one of those things that seem confusing and pointless until you need them, then you wouldn't be without them.

For the moment try running Program III. You shouldn't have any difficulty seeing how it works.

As you can see, it's more or less the same as the programs we had last month, except for the use of the variable `flag`.

Each time round the loop line 40 sets `flag` to 0.

If the condition in line 60 is true, then `flag` is given the value `-1`. If `flag` is `-1`, then line 70 ensures that one is added to the running total kept

```
10 REM PROGRAM III
20 count=0
30 FOR loop=1 TO 5
40 flag=0
50 READ test
60 IF test>5 THEN flag=-1
70 IF flag=-1 THEN count=count+1
80 NEXT loop
90 PRINT "There are ";count
  " numbers greater than 5
  " in the data list"
100 DATA 6,2,9,3,5
```

Program III

in `count`.

The next time round the loop `flag` is put back to 0 again. (If you feel adventurous, try using FALSE instead of the 0 in line 20.)

Now this is a fairly long-winded way of doing things, you could replace lines 60 and 70 with one line:

```
60 IF test>5 THEN
count=count+1
```

However I wanted you to see `flag` in action. What happened in the program depended on which of two values (0 and `-1`) `flag` took.

As `flag` toggled between the

two values it acted as a marker or flag, signalling whether `test` was greater than 5.

Now look at Program IV, which is very similar to the previous one, but replaces the condition after the IF of line 70 with a single variable `flag`.

```
10 REM PROGRAM IV
20 count=0
30 FOR loop=1 TO 5
40 flag=0
50 READ test
60 IF test>5 THEN flag=-1
70 IF flag THEN count=count+1
80 NEXT loop
90 PRINT "There are ";count
  " numbers greater than 5
  " in the data list"
100 DATA 6,2,9,3,5
```

Program IV

Here `flag` is acting as a logical variable. When it is `-1` (which only happens when `test` is greater than 5) one is added to `count`.

The IF of line 70 expects a conditional expression, and finds a variable which has the value of `-1`. To the Electron `-1` means the same as true, so it executes the rest of the line and adds one to `count`.

When `flag` is 0 (as happens if `test` is not greater than 5), the IF finds a 0, interprets it as meaning that the condition is false and the rest of the line is ignored.

So you can use logical variables, alternating between 0 and `-1` to keep track of the results of conditional expressions.

Although in the examples above they aren't really of much benefit, when we get to more complicated IF... THEN statements, they come in very handy.

But before we do, try replacing line 60 in Program IV with:

```
60 flag=test>5
```

and see if you can understand what's happening.

So far, there's only been one conditional expression between the IF and the THEN.

We've only been able to deal with conditions such as "If it's raining then I'll stay at home".

Here the choice is simple,



## From Page 9

it's either raining or it's not.

However in real life we often come across more complicated conditions such as: "If it's wet or windy then I won't come", or "If it's warm and sunny then I'll be there".

Notice that now all sorts of possibilities open up. It could be sunny but not warm or it could be both wet and windy.

It's often like this in programming. We don't just want to know if a number is positive, we want to know if it's larger than 100 and also divisible by 2.

The action of a game can depend on whether the time limit is up or you've used up your last laser base or both.

In other words, we need to know the results of two or even more conditions before we can choose which action to take.

Let's take the case where we say: "If it's warm and sunny then I'll come".

What we're doing is saying that we will come only if both conditions are true. We'll come if it's both sunny and warm at the same time.

If it's sunny but cold we won't be coming. If it's clouded over but warm we still won't come.

We want it to be sunny AND warm before we'll make a move. Both conditions must be met before we take action.

This could be summarised as:

```
IF both the first condition
is true
AND the second condition is
true
THEN I'll do it
```

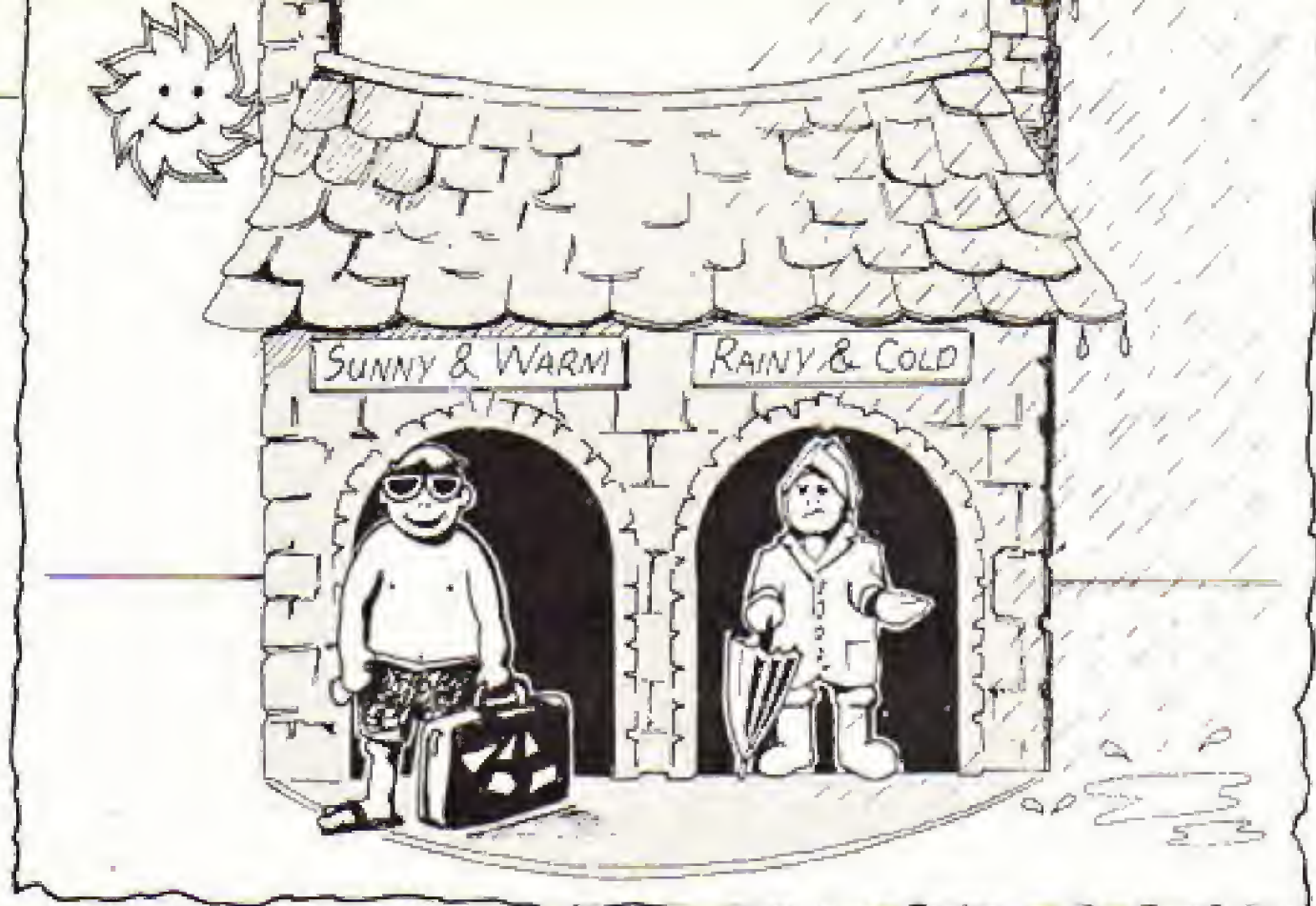
As you can see, we make these types of decisions all the time. If the shop is open and I've got enough money then I'll buy a loaf.

You can see that it would be useful if the Electron could make decisions like this and it can, using the AND logical operator.

Program V shows it in action.

As you can see from the listing, the program consists of a FOR ... NEXT loop which cycles five times.

Each time round the loop line 30 tells you to enter two



"If it's wet and windy then I won't come"

```
10 REM PROGRAM V
20 FOR loop=1 TO 5
30 PRINT "Enter two num
bers"
40 INPUT first,second
50 IF first>=0 AND secon
d=3 THEN PRINT;"is p
ositive and the second numb
er is 3"
60 NEXT loop
```

Program V

numbers while line 40 stores them in *first* and *second*.

Line 50 has the familiar looking IF and THEN but there are two conditions in it. It tests both numbers before coming to the THEN.

The first condition is *IF first >= 0*. This checks to see whether *first* is either greater than or equal to 0. In other words it checks if *first* is positive.

If it is then the condition is true and the Electron records this with a -1.

The second condition, *second=3* is only true if the second number is 3. Again, if the condition is true then it's evaluated as -1.

Both conditions however, are linked by the logical operator AND. This tells the Electron that before it can proceed with whatever comes after the THEN, both the first condition and the second have to be true.

For the message to be printed out not only must the first condition be fulfilled, the second one must also be correct.

In this case the first number has to be positive at the same

time as the second is equal to three. Then, and only then, will the rest of the line be obeyed.

You can run Program V as many times as you want but the message only appears when the first number you enter is positive and the second is equal to three.

The point about the AND operator is that both of the conditions have to be true before the rest of the line is completed.

It's no good the first condition being true while the second is false or vice versa. They've both got to be fulfilled or nothing happens.

Take a look at Program VI which also uses two conditions joined by an AND.

As you can see, we have our usual FOR ... NEXT loop cycling five times.

Each time round the loop line 40 reads in a number from the data in line 80 and stores it in *test*.

Line 50 then subjects *test* to two tests, *test >= 10* and *test < -20*.

The first condition is true if *test* is equal to or greater than

10. The second condition is only true if *test* is equal to or less than 20.

However both conditions are joined by the AND logical operator, so the message after the THEN isn't printed until both conditions hold true.

In other words, for the rest of the line after the THEN to be obeyed *test* must lie between 10 and 20.

So the AND has linked the two conditions into one major condition that is only true when the two subsidiary conditions are true.

It might make it clearer if you rewrite line 60 as:

```
50 IF (test)=10 AND
test(<=20) THEN inrange=TRUE
```

Here the two conditions are enclosed in the brackets, making the line look like our old familiar:

```
IF condition is true THEN
perform the rest of the
line.
```

The only difference is that now the condition that decides whether the rest of the line is performed is made up of two subsidiary conditions joined with an AND.

And that's it for this month. Try running Program VI with different conditions in line 50 and different numbers in the data statements and see if you can understand what's happening.

Also notice the use of the logical variable *inrange*.

● Next month we'll be looking at AND again, along with two other logical operators, OR and EOR.

```
10 REM PROGRAM VI
20 FOR loop=1 TO 5
30 inrange=FALSE
40 READ test
50 IF test>=10 AND test<
=20 THEN inrange=TRUE
60 IF inrange THEN PRINT
;"test" is in the required r
ange"
70 NEXT loop
80 DATA 9,12,5,17,23
```

Program VI

PASCAL is the latest in a series of programming languages from Acornsoft. It arose from investigations into possible developments resulting from the inclusion of data structuring facilities in an ALGOL-60 like language.

It was designed around 1970 mainly by Professor Niklaus Wirth working at the Institute for Informatics in Zurich, but also benefited by the inclusion of some of the ideas of C.A.R. Hoare who was also working on data structuring facilities in programming languages.

He published his language in 1971 and named it after the great seventeenth century French philosopher Blaise Pascal, who invented one of the earliest known calculators.

Two years later, in 1973, Hoare and Wirth attempted a formal definition of the language in response to user experience to shed light on areas of uncertainty. This led to a revision and extension of the original language.

As with all computer languages, Pascal was designed for a specific purpose. Niklaus Wirth's main objective was to produce a language better suited to teaching programming than any existing language at the time.

He was successful in his aims and it soon became popular as a teaching language.

Very quickly, user groups sprang up in several countries to exchange information and ideas on Pascal and the language was adopted by the University of California, San Diego in 1973/4 as their main teaching language.

UCSD were responsible for implementing Pascal for a wide range of computers.

One of the main reasons for Pascal catching on so quickly is that it is concise – the rules of grammar can be written down on just four or five pages.

Pascal is fairly simple to learn although complete beginners may have trouble initially as the knowledge required to write your first program is greater than for Basic.

Pascal is a highly structured

# Try S-Pascal and get rid of those spaghetti junctions



By ROLAND WADDILOVE

language with a rigid format that the programmer is required to adhere to. Everything is laid out so neatly and logically that it is difficult to go wrong.

It encourages a style of

programming in which programs are built up step by step from small well defined procedures.

All programs start with the word 'program' followed by the name of the program. All

the constants and variables used must be declared after the title, plus their type – for example, integer.

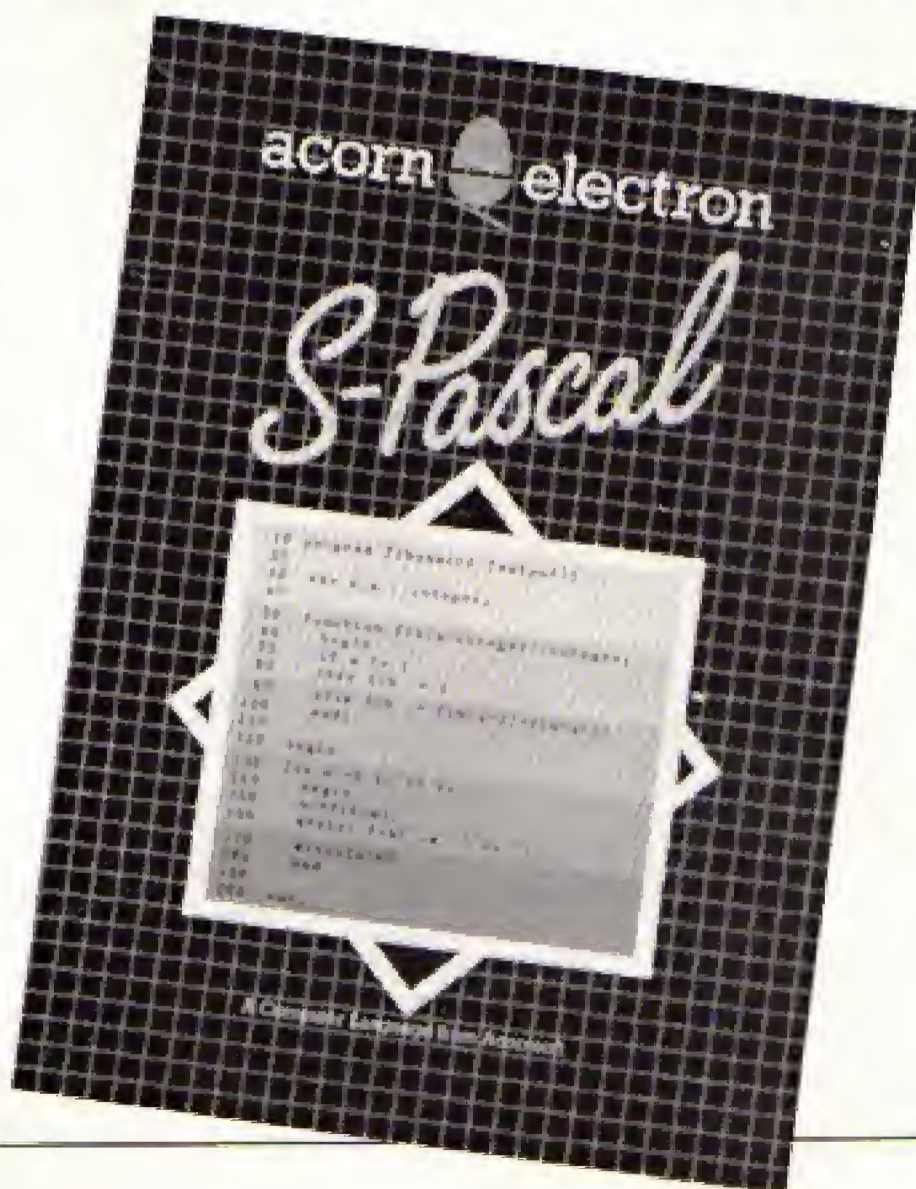
Any procedures used are defined following the variables and constants and the action part of the program commences with 'begin' and finishes with 'end'.

Pascal programs are very readable, being almost self documenting and needing very few comments. The program flow is easy to follow and the structure clear, making alterations, improvements and debugging very simple.

Lisp is quite interesting, Forth is fast and powerful, Basic just a Mickey Mouse toy for kids – but Pascal is a real programmer's language and a delight to use.

Pascal is a compiled language, not an interpreted one like Basic which means that Pascal programs run many times faster than their Basic equivalents.

There are two popular ways of implementing Pascal, each with its own advantages. Either the text of the source program can be compiled to pure machine code – which makes it very fast but specific to that machine – or it can be compiled to P-Code which is



## From Page 11

then interpreted when run, not unlike Forth.

This is slower but more easily transferred to other machines.

Acornsoft's S-Pascal is not a full blown version but contains a subset of Pascal to teach the language and provide an introduction to structured programming.

It is designed for people who know little or nothing about Pascal but are familiar with Basic. It allows short programs of up to 1.25k to be written, compiled and executed.

There are several important differences between this latest language from Acornsoft and the previous ones.

The first is noticed immediately on opening the box – which is slightly larger than normal. Inside is the cassette and manual whereas with the other languages the manual has to be purchased separately on top of the cassette. This makes S-Pascal some £7 cheaper than the others.

The second difference is noticed when S-Pascal is loaded and totally confused me at first – it wouldn't have if I had read the instructions, but then who does?

When loading is complete, after about five minutes, the Electron is still in Basic. The loader can be listed and Basic programs typed in and run – I thought it had not loaded and wondered where the Pascal was.

S-Pascal is a compiler only – not an interpreter – so commands cannot be entered in direct mode. What you get are several new \* commands to enable you to write, compile and run Pascal programs.

To type in a Pascal program \*NEW is entered. Programs can be typed in, edited and listed as with Basic, but using lower case characters so as not to confuse the compiler when it is run with Basic keywords which are stored as tokens.

\*COMPILE will activate the compiler producing code which is stored in a reserved area of memory. It can then be



executed with \*GO.

Pascal programmers will be disappointed with Acornsoft's S-Pascal as there are so many omissions compared to a full implementation and they will feel very restricted with the subset. However this is only designed to be a simple, limited version to give people an insight into how Pascal works.

Most Pascal reserved words are present with procedures, functions and arrays being possible, and all the mathematical operators are available. However, hardly any of the predefined functions or procedures have been included such as SIN, COS, and ABS.

Variables can be character, Boolean or integer, but not real, which explains why many of the functions are not available.

Call has been added – not a standard Pascal word – to allow machine code routines and the operating system to be accessed from within Pascal.

Acornsoft have chosen to compile the source text directly to machine code instead of P-Code as with many implementations.

The code is placed starting at &1100 and there is enough room for about 2.5k. The source text can be saved in the same way as Basic and the object code produced, saved with \*SAVE.

Compiling the source text directly to machine code has several advantages over compiling to P-Code. After compiling, the compiler – actually a Basic program 11k long plus 4k workspace, residing at

&1F00 – is no longer needed.

This means the object code can be \*RUN on its own, or the compiler space used for a Basic program which calls the machine code, or high resolution graphics – for example Mode 0.

Instead of using a Basic compiler program, why not write in Pascal, a far superior language, and compile that? A Pascal compiler is far more powerful than a Basic equivalent, with far fewer restrictions.

Can a Basic compiler cope with multi-dimensional arrays, procedures and functions to which parameters are passed and that have local variables? Acornsoft's S-Pascal can.

The compiler uses a two pass assembly, printing the mnemonics and object code each time, and if the printer is enabled, it can be listed.

Errors are spotted on the second pass and the appropriate line listed with an arrow pointing to the mistake, and a message is printed saying what the error number is and where it occurred in the line.

The error can then be looked up in the manual or on the reference card supplied.

I was curious to find out just how fast Pascal was. How efficient is the machine code? So I wrote equivalent – or near enough – programs in Basic, Forth, Lisp, Pascal and assembly language.

It simply involved setting a variable to zero, then going round a loop 30,000 times, incrementing the variable by one each time.

The speed test results are shown in the panel on this

page.

The test showed Pascal to be up to three times as fast as Basic and marginally faster than Forth, which is generally reckoned to be a fast language itself.

The test also highlighted the incredible inefficiency of the code produced – Pascal taking some eight times longer than the specifically written machine code routine.

This is not a criticism of S-Pascal but is just a fact of life. Compilers cannot hope to be as efficient as a purpose written machine code routine.

Acornsoft has achieved their main objective of producing a simple subset of Pascal for teaching the language and structured programming. The compiler is straightforward to use and the manual is short – 67 pages – but clear, and covers every aspect in detail.

The tape, and manual, contains seven demonstration programs showing what the system is capable of, which is quite a lot.

S-Pascal has a further function as a tool for writing short machine code routines which can be \*RUN or called from within a Basic program. This is probably more useful to the experienced programmer.

Programmers are strongly recommended to look at Pascal – especially those writing so called 'spaghetti' programs full of GOTOs. It will improve their structure no end. If you already write structured programs then learning Pascal will be a doddle.

S-Pascal is a welcome addition to the list of programming languages for the Electron, and if they ever bring out a full blown version on a ROM cartridge you can bet I will be one of the first to get it.

### How fast is Pascal?

THESE are the results of the speed test described in this article:

**Assembler . 1.4 seconds**  
**Pascal ..... 11.3 seconds**  
**Forth ..... 12.5 seconds**  
**Basic ..... 34.9 seconds**  
**Lisp ..... 285.0 seconds**

# Crafty colour can create kinematics!

ALAN PLUME continues his series on how to achieve simple but effective Electron animation

THE last article showed what can be done by using a combination of straight-forward character animation and very simple Basic programs.

This article will show how pleasing animated displays can be achieved by simply using the colour capabilities of the Electron.

The first method is easily demonstrated by displaying a rotating object. Listing 1 shows how it's done.

Here the object to be rotated is a seven-sided polygon drawn in Mode 2, made up of triangles coloured in the Electron colours 1 to 7 (red, green, . . . white).

To make it rotate (or appear

to rotate) all that is necessary is to make the red triangle change to green, the green to yellow, the white to red, and so on continuously.

The speed of rotation can be increased by altering the value of *wait* at line 40. For some values of *wait* you will notice lines moving across the coloured areas.

This happens because the change of colour is not synchronised to the vertical scan of the computer.

The \*FX19 at line 110 waits for the vertical synch to occur before palette switching and has some smoothing effect.

See what happens when you delete this and insert it at

line 125.

The second method can be used to display relatively complex shapes either moving or changing shape.

The obvious way to move an object made up of lines is to draw the object in position 1, calculate a new position, erase the object in position 1, draw in position 2 and so on.

Now to draw an object made from a number of lines takes time in Basic (even on the Electron's big brother!), so that each time we erase the object the screen is momentarily blank.

A solution to this is to draw or erase some lines on the display without affecting others. Then the display will

remain continuous even if not completely smoothly animated.

This effect can be achieved by careful use of the logical actions available in the GCOL statement coupled with palette changes (VDU19).

The first object is drawn using, for example, white. The lines for the second object are calculated and drawn in black in such a way that any overlaps with the first object do not erase it.

Then a palette change is made so that the first object becomes black (matching the background colour) and the second white, again making sure that no erasures occur.

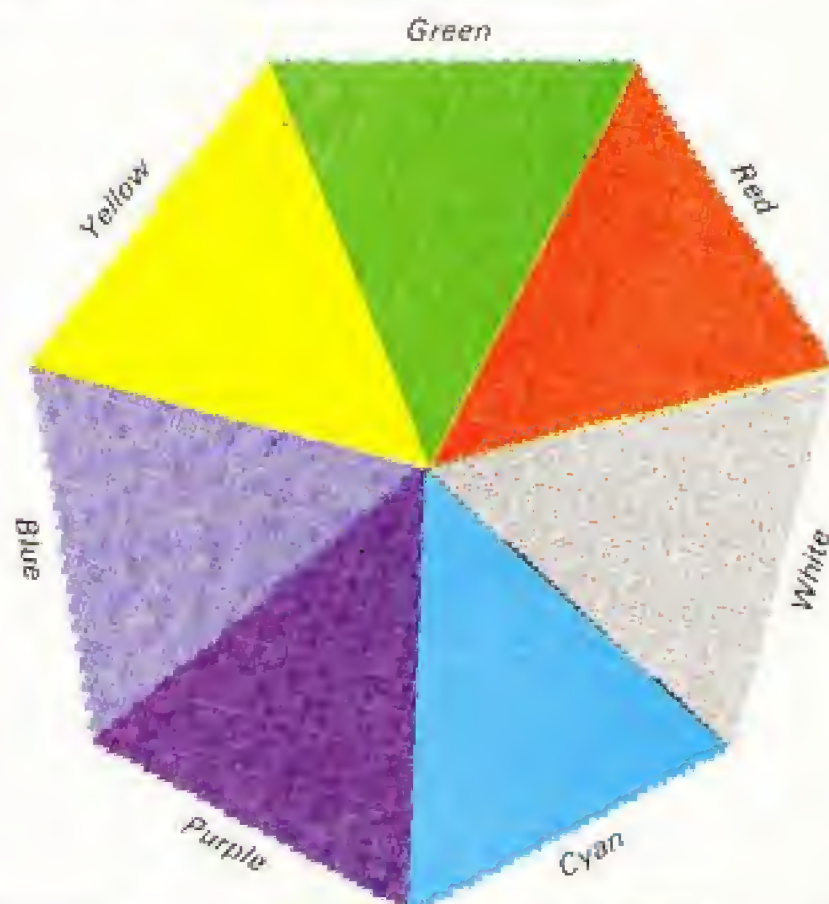


```

5 REM LISTING 1
10 MODE2
20 VDU23,1,0;0;0;0;
30 DIM col% 7
40 wait=10
50 FOR I%=1 TO 7
60   col%?I%=I%
70 NEXT
80 VDU29,640;512;
90 PROCc%rc
100 REPEAT
110 *FX19
120 FOR C%=1 TO 7
130   VDU19,C%,col%?C%;0
140 NEXT
150 FOR C%=1 TO 7
160   I%=col%?C%+1; IF I%
=8 I%=1
170   col%?C%=I%
180 NEXT
190 TIME=0:REPEAT UNTIL T
IME=wait
200 UNTIL FALSE
210 END
220 DEFPROCc%rc
230 a%nc=RAD(360/7)
240 R%=400
250 FOR I%=0 TO 6
260   SCOL0,I%+1
270   MOVE0,0:DRAWR%*COS
(I%*a%nc),R%*SIN(I%*a%nc):J
I%=I%+1:PLOT85,R%*COS(JI%*a%nc),R%*SIN(JI%*a%nc)
280 NEXT
290 ENDPROC

```

Listing 1



## From Page 13

The first object is then erased and the process repeated for subsequent objects. Listing II shows this in action for the letter A.

Both of these methods, simple palette switching and

drawing/erasing, can be the basis of very effective animated displays.

I hope that the two programs have whetted your appetite and that you will explore more deeply into animation using Electron Basic.



```

5 REM LISTING II
10 MODE1
20 VDU23,1,0;0;0;0;
30 READ npts
40 DIM xI(npts),yI(npts)
,pI(npts)
60 FOR pt=1 TO npts
70 READ pI(pt),xI(pt)
,yI(pt)
80 NEXT
100 scaleI=2
110 VDU29,0;512;
120 VDU19,3,3;0;19,1,3;0;
130 colI=1
140 xincI=0
141 GCOL0,colI:PROCdrawit
(xincI)
145 VDU19,colI,3;0;19,3-c
olI,0;0;
146 FOR posI=1 TO 80
150 xincI=xincI+16
160 colI=colI EOR 3
170 GCOL1,colI:PROCdra
wit(xincI)
175 #FI19
180 VDU19,colI,3;0;19,
3-colI,0;0;
185 GCOL2,colI:PROCdra
wit(xincI-16)
200 NEXT
205 END
210 DEFPROCdrawit(AI)
220 FORIX=1TONpts
230 PLOTpI(IX),xI(IX)*
scaleI+AI,yI(IX)*scaleI
240 NEXT
250 ENDPROC
260 DATA3
270 DATA4,-8,64,5,8,64,
5,40,-64,5,24,-64,5,12,-2
0,5,-12,-20,5,-24,-64,5,
-40,-64,5,-8,64
280 DATA4,0,32,5,8,-4,5
,-8,-4,5,0,32
    
```

Listing II

EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES



# The Definitive Adventures for the Electron...

## The Wheel of Fortune

This game is a classic puzzle adventure with all the features you'd expect from EPIC...

### PLUS

- Intelligent moving characters with varying moods. **And you can talk to them too!**
- Multistatement language and speech interpreters.
- Runs in real time.
- **250** locations and over **30,000** characters of text. Only Epic's compression techniques can pack so much into the Electron.

"Having now tried all the Epic Adventures, they must be the yardstick by which all other adventures for the Electron should be judged."

ELECTRON USER

"The Wheel of Fortune for the BBC and Electron is a highly-recommended state-of-the-art adventure."

SHIELDS GAZETTE

"This has to be the adventure of 1984. It really is superb."

MICRONET 800

"The definitive adventure. Highly recommended."

ELECTRON USER

**Our other three adventures have also received superb reviews in Electron User. They each contain approximately 230 locations and 25,000 characters of text.**

TO: EPIC SOFTWARE, DEPT. E, 10 GLADSTONE ST., KIBWORTH BEAUCHAMP, LEICESTER LE8 0HL Please Rush Me -

	CASSETTE	DISC	STATE:
.....qty THE WHEEL OF FORTUNE	£9.95	£11.95	BBC/
.....qty CASTLE FRANKENSTEIN	£7.95	£9.95	ELECTRON
.....qty THE QUEST FOR THE HOLY GRAIL	£7.95	£9.95	(Delete)
.....qty THE KINGDOM OF KLEIN	£7.95	£9.95	No Graphics on Electron

POSTAGE & PACKING FREE FOR 2 OR MORE. ADD 50p FOR ONE.

I enclose Cheque/P.O. to the value of ..... (Payable to 'Epic Software')

NAME .....

ADDRESS .....

POST CODE .....

All programs available for immediate despatch. Dealer enquiries welcome. Help Service. Send letter if you don't want to cut magazine.

EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES...EPIC ADVENTURES

# FIRST BYTE ELECTRON JOYSTICK INTERFACE

Printer Interface  
**OUT NOW**  
Uses normal BBC printer commands  
No software required!

**WORKS  
WITH  
MOST GAMES  
£19.95**



## ELECTRON JOYSTICK INTERFACE

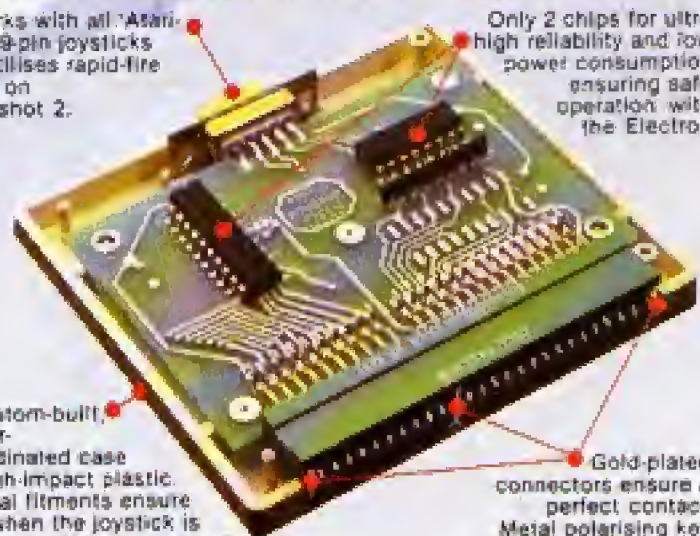
Electron users! This is the add-on everyone wants. It's the new Electron switched joystick interface from First Byte - available now with free conversion tape that vastly extends your game range right away.

The interface operates with all 'Atari-style' 9-pin joysticks, and its many advanced design features put it way out in front for quality and reliability. That's why, to date 15 major software houses are already bringing out games that work directly with the First Byte Electron Joystick Interface - and many more are sure to follow.

## Look at these advanced design features.

Works with all 'Atari-style' 9-pin joysticks and utilises rapid-fire mode on Quickshot 2.

Only 2 chips for ultra-high reliability and low power consumption ensuring safe operation with the Electron.



Custom-built, colour-co-ordinated case in high-impact plastic. Special fittings ensure that when the joystick is plugged in, the case takes the strain, not the soldered joints.

Gold-plated connectors ensure a perfect contact. Metal polarising key and nylon end caps ensure positive locking.



First Byte Computers,  
10, Castlefields,  
Main Centre, Derby.  
DE1 2PE  
Tel: Derby (0332) 365280

A GENUINE FIRST BYTE  
ADD-ON

AVAILABLE FROM WHSMITH AND MOST COMPUTER DEALERS

### MAIL ORDER FORM

Please rush me the following items:  
 First Byte Joystick Interfaces at **£19.95** \_\_\_\_\_  
 First Byte Printer Interfaces at **£34.95** \_\_\_\_\_  
 TOTAL \_\_\_\_\_

I enclose a cheque made payable to FBC Systems Ltd.,  
 I wish to pay by  Access  Visa Expiry date \_\_\_\_\_

Card No.

Signed \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_ Tel \_\_\_\_\_

# QUICK TO LEARN

## THAT'S...

# MINI OFFICE

**SPREADSHEET**

A	B	C	D
NAME	AMOUNT	PERCENT	DATE
1			
2	WATER	50.00	05.10
3	FUEL	40.00	05.10
4	FUEL	40.00	05.10
5	PHONE	10.00	05.10
6	OTHER	10.00	05.10
7	TOTAL	190.00	
8			
9			
10			
11	TRAINING	100.00	05.10
12	B. FUEL	40.00	05.10
13	10 SHAW	200.00	05.10
14	10 SHAW	200.00	05.10
15	TRAINING	100.00	05.10
16			
17			
18			
19			

## JUST LOOK WHAT THIS PACKAGE CAN DO!

**WORD PROCESSOR** – Ideal for writing letters or reports! *Features:* Constant time display ● Constant word count (even shows words per minute) ● Normal or double-height text on screen or printout.

**SPREADSHEET** – Use your micro to manage your money! *Features:* Number display in rows and columns ● Continuous updating ● Update instantly reflected throughout spreadsheet ● Save results for future amendments.

**GRAPHICS** – Turn those numbers into an exciting visual display! *Features:* 3D bar chart ● Pie chart ● Graph.

**DATABASE** – Use it like an office filing cabinet! *Features:* Retrieve files at a keystroke ● Sort ● Replace ● Save ● Print ● Search.

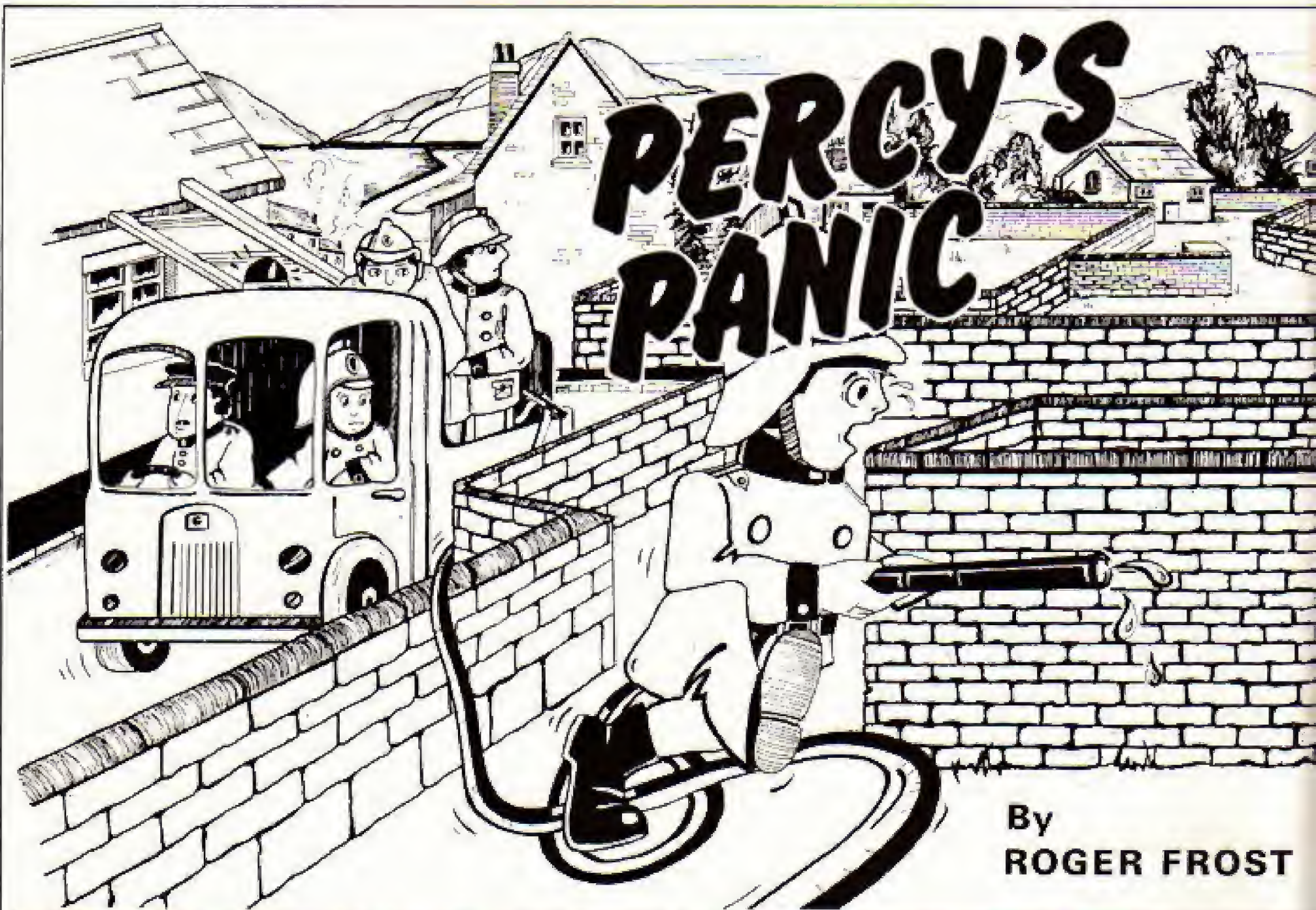
**DATABASE**

RECORD No. 1	RECORD No. 1
SURNAME: JONES FIRST NAME: SIMON ADDRESS1: 5 MADON LANE ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 47	SURNAME: ANDREWS FIRST NAME: JAMES ADDRESS1: 10 ELF ROAD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 17
RECORD No. 2	RECORD No. 7
SURNAME: ANDREWS FIRST NAME: PETER ADDRESS1: 10 ELF ROAD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 35	SURNAME: ANDREWS FIRST NAME: PETER ADDRESS1: 10 ELF ROAD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 19
RECORD No. 3	RECORD No. 8
SURNAME: SMITH FIRST NAME: JANE ADDRESS1: 42 HIGH STREET ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 27	SURNAME: JONES FIRST NAME: PETER ADDRESS1: 10 ELF ROAD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 20
RECORD No. 4	RECORD No. 9
SURNAME: JONES FIRST NAME: JIM ADDRESS1: 177 FORD ROAD ADDRESS2: GULLHAM TELEPHONE: 450-986 76540 AGE: 35	SURNAME: BROWN FIRST NAME: JIM ADDRESS1: 17 LEAVERD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 21
RECORD No. 5	RECORD No. 10
SURNAME: ANDREWS FIRST NAME: JAMES ADDRESS1: 10 ELF ROAD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 17	SURNAME: BROWN FIRST NAME: JIM ADDRESS1: 17 LEAVERD ADDRESS2: WARRINGTON TELEPHONE: 051-627401 AGE: 17

## ...and it's all at price of just







# PERCY'S PANIC

By  
ROGER FROST

```

10 REM***PERCY'S PANIC**
*
20 REM***BY ROG & SUE FR
OST***
30 REM (C) ELECTRON USE
R
40 REM
50 MODE6:PROCintro
60 vol%=-10:hiscore%:=0:h
iscore%=""
70 PROCchrs
80 REPEAT
90 MODE5
100 flag%=0
110 PROCcaze
120 PROChouse
130 PROCengine
140 TIME=0
150 PROCcan
160 IF flag%=0PROCmove:IF
life%(>0)FORdelay%=0TO2000
0:NEXT:burn%=burn%+1
170 UNTILlife%=0
180 #FX15,0
190 REPEATUNTILGET=32
200 GOTO70
210 END
220 DEFPROCcaze
230 fuse%=0
240 VDU5
250 GCOL0,3
260 FORY%=700TO140STEP-40
270 MOVE0,Y%:DRAW1279,Y%
280 GCOL0,0
290 p%:=RND(1100):q%:=RND(1
100):r%:=RND(1100)
300 gap%:=RND(3)
310 IF gap%>0 MOVEp%,Y%:D
RANp%+50,Y%
320 IF gap%>1 MOVEq%,Y%:D
RANq%+50,Y%
330 IF gap%>2 MOVEr%,Y%:D
RANr%+50,Y%
340 GCOL0,3
350 NEXT
360 GCOL0,3:MOVE0,100:DR
A#370,100:MOVE440,100:DRAW12
79,100
370 MOVE0,0:DRAW0,800:MOV
E1279,100:DRAW1279,800
380 MOVE0,0:DRAW0,800:MOV
E1271,100:DRAW1271,800
390 ENDPROC
400 DEFPROChouse
410 GCOL0,2
420 MOVE800,900:MOVE1100,
800:PLOT85,800,900:PLOT85,1
100,900
430 MOVE1050,900:PLOT85,1
050,1000:PLOT85,850,900:PLO
T85,850,1000:PLOT85,800,900
440 GCOL0,1:MOVE915,885:M
OVE985,885:PLOT85,915,900:P
LOT85,985,900:GCOL0,0:MOVE0
30,850:MOVE880,850:PLOT85,0
30,900:PLOT85,880,900:MOVE1
070,850:MOVE1020,850:PLOT85
,1070,900:PLOT85,1020,900
450 GCOL0,3
460 MOVE0,850:DRAW000,850
470 MOVE900,780:VDU224
480 ENDPROC
490 DEFPROCmove
500 X%:=max%:Y%:=50
510 GCOL0,1
520 REPEAT
530 IF INKEY(-66)AND POIN
T(X%+12,Y%)(<>3) THEN Y%=Y%+1
0
540 IF INKEY(-104)AND X%
1220 X%=X%+20
550 IF INKEY(-103)AND POI
NT(X%-20,Y%+30)<>3 X%=X%-20
560 IF INKEY(-98) AND Y%
800 Y%=Y%-10
570 MOVEX%,Y%:VDU225
580 IF TIME MOD 9>6 GCOL0
,1:MOVE fuse%,850:DRAWfuse%
+burn%,850:fuse%=fuse%+burn
%
590 SOUND1,vol%,fuse% DIV
4,1
600 UNTILPOINT(X%,Y%)=2 0
R fuse%>820
610 IFfuse%>820PROCexplod
e:ENDPROC
620 TX%=TIME DIV 100
630 PROCsuccess
640 ENDPROC
650 DEFPROCexplode
660 life%=0
670 SOUND0,vol%,70,20
680 FOR line%:=0TO100
690 GCOL0,0:MOVE600,1000-(
line%+2):DRAW1279,1000-(li
ne%+2)
700 GCOL0,RND(4)
710 MOVE900,900:DRAW600+R
ND(800),900+RND(200)
720 NEXT
730 GCOL0,3
740 score%=score%+Y%+700-
max%
750 #FX15,0
760 CLG:PRINT""Your sc
ore is ";score%
770 IF score%>hiscore% IN
PUT""Your score is the""
best. Please enter""your
name ",hiscore%:hiscore%:=LE
FT$(hiscore%,10):hiscore%:=s
core%:PRINT""SPACE TO PLA
Y":ENDPROC
780 PRINT""Best score i

```



PERCY gets home from work and finds a burning fuse leading to his house. Luckily he hears a fire engine in the distance, but it's the other side of a maze. Can you guide Percy to the fire engine and roll a hose back to the house in time to save an explosion? If you get the hose to the house in time you get another go, but with less time allowed.

```

s ;hiscoreZ""Scored by ";
hiscore$"" "SPACE TO PLAY"
790 ENDPROC
800 DEFPROCsuccess
810 CLG
820 VDU4
830 PRINTTAB(0,10)"Well d
one! You saved""the house
. You took""TZ;" seconds."
840 scoreZ=scoreZ+YZ+(400
00 DIV TZ)+700-manyZ
850 PRINT""Your score is
";scoreZ
860 *FX15,0
870 ENDPROC
880 DEFPROCman
890 manxZ=900;manyZ=700
900 REPEAT
910 SCOL0,0;MOVEmanxZ,man
yZ:VDU224;SCOL0,3
920 IF INKEY(-104)AND man
xZ<1220 manxZ=manxZ+20
930 IF INKEY(-103)AND man
xZ>20 manxZ=manxZ-20
940 IF INKEY(-98) AND POI
NT(manxZ+30,manyZ-40)<>3 ma
nyZ=manyZ-40
950 IF INKEY(-66) AND man
yZ<100manyZ=manyZ+20
960 MOVEmanxZ,manyZ:VDU22

```

```

970 IF TIME MOD 9>5 SCOL0
,1;MOVE fuseZ,850;DRAWfuseZ
+burnZ,850;fuseZ=fuseZ+burn
Z
980 SOUND1,volZ,fuseZ DIV
4,1
990 UNTIL (POINT(manxZ+12
,manyZ-50)=1 OR fuseZ>820)
1000 IF fuseZ>820 flagZ=1:
PROCexplodes:ENDPROC
1010 SCOL0,0;MOVEmanxZ,man
yZ:VDU224
1020 ENDPROC
1030 DEFPROCengine
1040 SOUND1,1,60,60
1050 VDU5
1060 EXZ=1140
1070 REPEAT
1080 MOVEEXZ,70;SCOL0,0;PR
INT;ENGINE$:EXZ=EXZ-40;SCOL
0,1;MOVEEXZ,70;PRINT;ENGINE
$
1090 UNTIL EXZ<300
1100 SOUND1,0,0,0,0
1110 ENDPROC
1120 DEFPROCchrs
1130 ENVELOPE1,20,20,-20,0
,1,1,1,0,0,-126,126,126,126
1140 burnZ=4;scoreZ=0;life
Z=1
1150 VDU23,224,0,56,56,16,

```

```

56,16,40,100
1160 VDU23,225,0,0,60,60,6
0,60,0,0
1170 VDU23,226,15,13,16,40
,40,112,112,127
1180 VDU23,227,192,192,255
,135,135,135,135,255
1190 VDU23,228,0,0,255,255
,255,255,255,255
1200 VDU23,229,127,255,255
,255,255,255,7,3
1210 VDU23,230,255,240,240
,240,255,255,120,0
1220 VDU23,231,255,63,63,6
3,255,255,60,24
1230 ENGINE$=CHR$226+CHR$2
27+CHR$228+CHR$0+CHR$0+CHR$
8+CHR$10+CHR$229+CHR$230+CH
R$231
1240 ENDPROC
1250 DEF PROCintro
1260 VDU19,0,4,0,0,0
1270 PRINTTAB(13,4)"PERCY'
S PANIC"TAB(13,5)*****
*****
1280 PRINT"" Percy finds a
burning fuse leading to""
his house. Luckily he hea
rs a fire "" engine in the
distance; but it's the""
other side of a maze."" C

```

```

an you guide Percy to the f
ire engine"
1290 PRINT" and roll a hos
e back to the house in"" t
ime to save an explosion?
You say"" unroll the hose
as soon as you reach"" the
fire engine."" If you ge
t the hose to the house in"
" time, you get another go
, but with"
1300 PRINT" less time allo
wed."TAB(6,22)"PRESS SPACE
BAR TO CONTINUE"
1310 *FX15,0
1320 REPEATUNTILGET=32;CLS
1330 PRINTTAB(4,5)"To move
both Percy and hose use"TA
B(14,9)"A ... UP"TAB(14,11)
"Z ... DOWN"TAB(14,13)"> ..
. RIGHT"TAB(14,15)"< ... LE
FT"TAB(0,20)"PRESS SPACE BA
R TO BEGIN"
1340 *FX15,0
1350 REPEATUNTILGET=32
1360 ENDPROC

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*

# What a dump!

... but what a useful routine too! ROLAND WADDILOVE shows how to get your screens down on paper

APART from listing programs and printing text, many printers are capable of producing an exact copy of whatever is on the screen, be it text or graphics.

The ability to draw patterns and pictures and dump the screen to the printer is great fun, and also very useful if you use your micro for displaying data in graphical form.

Daisywheel printers, while giving excellent quality print, are not much use for screen dumps. You need a dot matrix printer with a directly addressable print head. The printer must have a bit image mode allowing the printing of raw data.

Assuming that you have a suitable printer, in normal operation any number sent to it will be interpreted as the code for a character.

By sending a series of control codes the bit image mode can be set, and now any

number sent to be printed is interpreted as literal data and is sent directly to the print head.

The print head consists of nine pins. The ninth is not used in the bit image mode, but the other eight each print a single pixel if the corresponding bit is set in the byte of data received by the head.

Pin 1 corresponds to bit 7, pin 2 to bit 6, pin 3 to bit 5 and so on down to pin 8 and bit 0.

To produce an exact replica of the screen all that is necessary is to scan it line by line, converting whatever is drawn or printed on it to a series of bytes with an identical bit pattern and sending it to the printer.

It probably sounds an extremely complicated process, but in actual fact is relatively simple.

How would you work out the eight parameters to define a character using VDU 23?

First you would draw the

A Histogram To Show The Frequency Of Occurance OF The Numbers 1-10 When RND(10) Is Used 500 Times



## BASIC listing

```

9000 DEF PROCscreen_dump(
1)
9010 VDU 2,1,27,1,65,1,8
9020 FOR YX=1023 TO 31 STEP
P -32
9030 VDU 1,9,1,27,1,75,1,6
4,1,1
9040 FOR X1=0 TO 1276 STEP
4
9050 AX=0:BX=128
9060 FOR CX=0 TO 28 STEP 4
9070 IF POINT(X1,YX-CX)<>I
1 AX=AX+BX
9080 BX=BX DIV 2
9090 NEXT
9100 VDU 1,AX
9110 NEXT
9120 VDU 1,10
9130 NEXT
9140 VDU 3
9150 ENDPROC

```

## Machine Code listing

```

10 REM *Mode 1/2/4/5 Scr
een Dump+
20 REM **By R.A.Waddilo
ve**
30 REM **IC) Electron Us
er**
40 REM ** CALL D%C,backg
round) **
50 x%=%70:y%=%72
60 a%=%74:b%=%75:c%=%76
70 background=%77
80 block=%78
90 parameter=%600
100 oswrch=%FFEE
110 osword=%FFF1
120 FOR pass=0 TO 3 STEP
3
130 PX=%900
140 I OPT pass
150 .D%
160 LDA parameter \back
ground ?
170 BEQ none
180 LDA parameter+1 \set
it
190 STA background
200 LDA parameter+2
210 STA background+1
220 LDY #0
230 LDA (background),Y
240 .none
250 STA background
260 LDA #2 \enable
printer
270 JSR oswrch
280 LDA #27 \set pap
er feed
290 JSR print
300 LDA #65
310 JSR print
320 LDA #8
330 JSR print
340 LDA #1023 MOD 256 \y%
=1023
350 STA y%
360 LDA #1023 DIV 256
370 STA y%+1
380 .loop1
390 LDA #9
400 JSR print \margin
410 LDA #27 \set numbe
r of
420 JSR print \data itee
s
430 LDA #75
440 JSR print
450 LDA #64
460 JSR print
470 LDA #1
480 JSR print
490 LDA #0 \x%=0
500 STA x%
510 STA x%+1

```

character, then convert it to a binary bit pattern then finally convert it to decimal or hexadecimal. It's just the same with the printer.

VDU 23 requires eight items of data, no more and no less. The printer, however, can accept a variable number of data items up to a maximum.

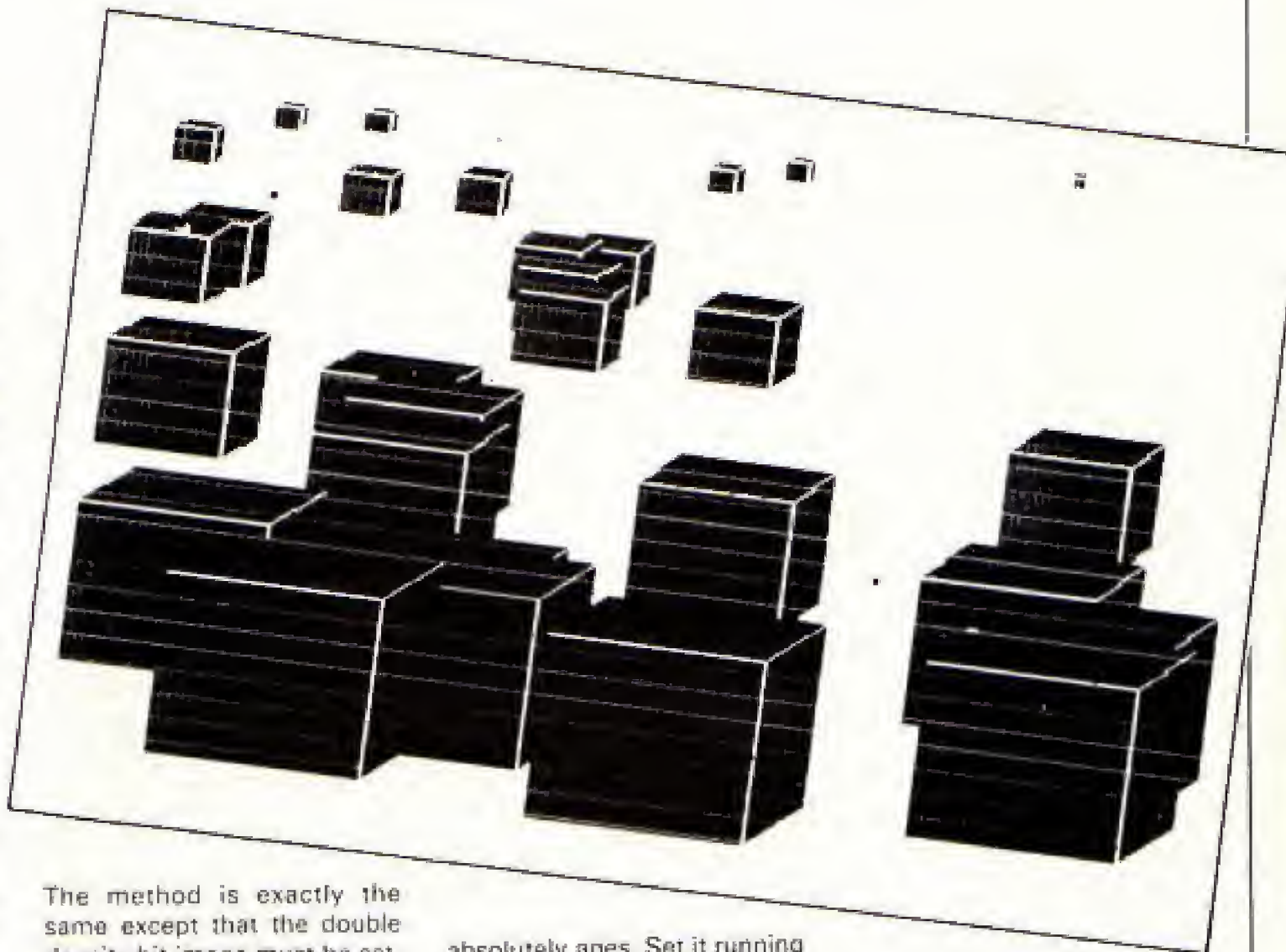
Before the data is sent it must be told how many items it is to expect, any further data being interpreted as the codes for characters as normal.

When printing text, a gap is left in between lines for clarity, and as I mentioned before, pin 9 is not used in the bit image mode, adding a further gap.

This would obviously ruin our picture or graph, so the paper feed must be set so that the next line of eight pixels is printed directly beneath the previous one without any space. A few control codes will do this.

The screen dumps listed here operate in Modes 1, 2, 4 and 5. Modes 3 and 6 are text only so a dump is unnecessary.

Mode 0 requires the printer to be set up slightly differently as the resolution is so high.



The method is exactly the same except that the double density bit image must be set.

The machine code dump is identical in structure to the Basic dump.

Even though the assembly listing is 10 times as long and 10 times as complex it has a couple of advantages over the Basic dump.

The main difference is the speed. The Basic dump takes

absolutely ages. Set it running then go and have a cup of tea. By the time you get back it might have finished, if you are lucky.

The machine code version knocks about 10 minutes off the time.

The second advantage is that the code can be tucked away in some odd corner of

the memory that is not being used, leaving more room for your program.

The dumps should work with any Epson-compatible printer with a bit image mode.

The Basic listing can be

```

520 .loop2
530 LDA #128 \b%=128
540 STA b%
550 LDA #0 \a%=0
560 STA a%
570 STA c% \c%=0
580 .loop3
590 LDA x% \POINT(x%,
y%-c%)
600 STA block
610 LDA x%+1
620 STA block+1
630 SEC
640 LDA y%
650 SBC c%
660 STA block+2
670 LDA y%+1
680 SBC #0
690 STA block+3
700 LDA #9
710 LDX #block MOD 256

720 LDY #block DIV 256
730 JSR oswrd
740 LDA block+4
750 CMP background \is it
background?
760 BEQ next
770 CLC \a%=a%+b%
780 LDA a%
790 ADC b%
800 STA a%
810 .next
820 CLC \b%=b% DIV 2
830 ROR b%
840 LDX c% \c%=c%+4
850 INX:INX
860 INX:INX
870 STX c%
880 CPY #32 \c%=32 ?
890 BNE loop3
900 LDA a% \send data
910 JSR print

920 CLC \x%=x%+4
930 LDA x%
940 ADC #4
950 STA x%
960 LDA x%+1
970 ADC #0
980 STA x%+1
990 CMP #65 \x%=128 ?
1000 BNE loop2
1010 LDA #10 \paper feed
1020 JSR print
1030 SEC \y%=y%-32
1040 LDA y%
1050 SBC #32
1060 STA y%
1070 LDA y%+1
1080 SBC #0
1090 STA y%+1
1100 BCC end \y%<0 ?
1110 JMP loop1
1120 .end

1130 LDA #3 \disable pr
inter
1140 JSR oswrch
1150 RTS \finished
1160 .print
1170 PHA \save char
acter
1180 LDA #1 \printer o
nly
1190 JSR oswrch
1200 PLA \get chr
1210 JSR oswrch \print it
1220 RTS
1230 )
1240 NEXT
1250 END

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*

## From Page 21

added to the end of an existing program, PROCscreen\_dump (1%) where 1% is the background colour will produce the dump.

The assembly listing could be added to an existing program, or loaded and run when you switch on. It's then ready whenever you need it.

Simply CALL the start to dump the screen. CALL D% if the background is colour 0, or CALL D%,b where D% is the start and b is the background colour.

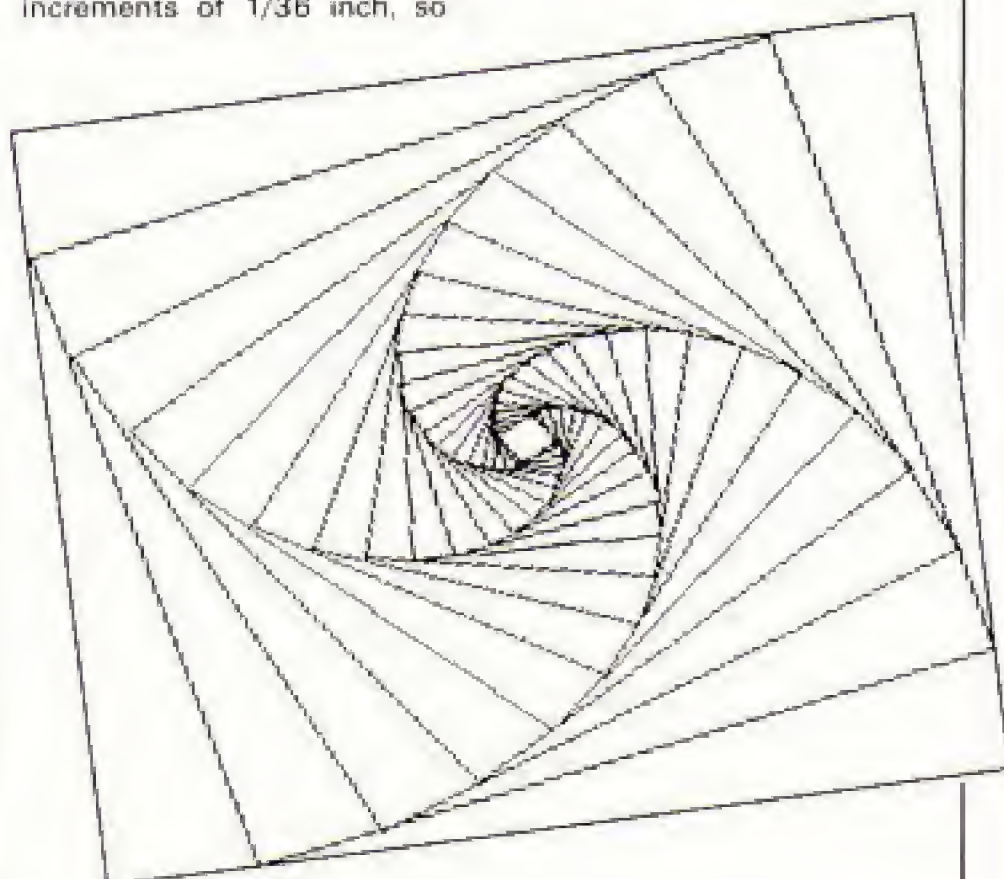
The pin spacing of the print head is 1/72 inch so the paper feed must be set to 8/72 inch. Line 9010 of the Basic listing and lines 280-320 of the assembly listing do this. Check out your printer manual for the

correct code.

The Epson in my office has a minimum paper feed of 1/72 inch so the feed is 8\*minimum, my own Brother HR-5 moves the paper up in increments of 1/36 inch, so

the feed is 4\*minimum, 4/36 = 8/72.

I simply change the 8 in line 9010 (Basic), or line 320 (Machine code), to 4.



```

9010 Enables printer, sets paper feed.
9020 From top line to bottom line.
9030 Print margin, set number of data items to 320.
9040 From left to right.
9050 Sets data to 0, bit to 7.
9060 For pixel 0 to 7.
9070 Tests pixel, adds bit to data.
9080 Next bit.
9090 Next pixel.
9100 Prints data.
9110 Next-x coordinate.
9120 Paper feed.
9130 Next line.
9140 Disables printer.
    
```

How the Basic works

# FOR CONNOISSEURS OF ACTION, LEISURE OR LEARNING

BBC Electron

**ANSWER BACK Sports Game £9.95**  
(ages 14 and over)  
Challenge the KOSMOS team at football or tennis. But be warned, we don't lose easily...  
- NEW RELEASE  
Whatever your sport, this program probably knows a lot more than you do!  
**ANSWER BACK Junior £9.95**  
(ages 6 to 11)  
Defeat the dragon and save the princess at KOSMOS Castle...  
- a superb program with vast potential... (A & B Computing, November 1984)



**ANSWER BACK Senior £9.95**  
(ages 12 and over)  
Outwit the KOSMOS robot and destroy the alien invaders...  
- superb program... (Personal Computing Today, October 1984)  
**IDENTIFY EUROPE £7.95**  
(for all ages)  
Save the European jigsaw puzzle by land or sea...  
- NEW RELEASE Discover Europe without leaving your home!

The IDENTIFY EUROPE program provides a fascinating way of discovering and learning the geography of Europe. The program will provide hours of amusement for all the family and everyone is sure to benefit from it.

The ANSWER BACK series needs little introduction. Each program combines a massive wealth of information with a fascinating and compelling game. The Senior and Junior quizzes each contain 750 questions and 3000 optional answers on General Knowledge. The Sports program is even larger containing a mind-bending 800 questions on Sport and two high-speed machine code games - Football and Tennis. But that's not all. For quiz fans, full facilities are provided for creating and saving new quizzes or modifying those supplied - and you don't need to be a programmer!

KOSMOS SOFTWARE LTD  
1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX  
Tel. (05255) 3942



Dealers, contact Lightning, Microdeal, Proteus or Tiger

Please supply the following programs for the BBC/Elctron computer

ANSWER BACK Sport @ £9.95  ANSWER BACK Junior @ £9.95   
IDENTIFY EUROPE @ £7.95  ANSWER BACK Senior @ £9.95

Mr/Mrs/Miss .....

Address .....

Postcode .....

Orders are normally despatched within 48 hours

KOSMOS SOFTWARE LTD, 1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX Tel. (05255) 3942

# Notebook Part 14

## Get the Point?

CAN you get the Electron to display decimal numbers underneath each other with all the decimal points aligned?

Until I was asked how to achieve this at a recent Electron & BBC Micro User Show I'd never realised that there was no straightforward way of doing it.

Hence this month's Notebook program, Centre Point, which does the job.

FOR...NEXT loop cycles 5 times READs in DATA

```

10 REM CENTRE POINT
20 MODE 6
30 FOR loop=1 TO 5
40 tab=20
50 READ decimals
60 PROCcentre(decimals,tab)
70 NEXT loop
80 END
90 DEF PROCcentre(number,pointposition)
100 number$=STR$(number)
110 length=LEN(number$)
120 slice=0
130 REPEAT
140 slice=slice+1
150 point$=MID$(number$,slice,1)
160 UNTIL point$="." OR slice=length
170 PRINT TAB(pointposition-slice);number$
180 ENDPROC
190 DATA 1.2,12.3,123.45,123.456,123456.78
    
```

calls centring procedure

makes number a string and finds its length

searches string for decimal point

centring procedure

output from program

```

1.2
12.3
123.45
123.456
123456.78
    
```

- 30-70 Form a FOR...NEXT loop which cycles five times. Each time round the loop a number is READ from the DATA list (line 50) and PROCcentre (line 60) is called to deal with it.
- 40 Sets the variable *tab* to 20. If you want the decimal points to occur at a different location then change this line.
- 60 Calls PROCcentre, passing to it the values contained in *decimals* and *tab*.
- 80 The END stops the program crashing into the procedure definition.
- 90-180 Define PROCcentre using the variables *number* and *pointposition*.
- 100 Uses STR\$ to convert the numeric variable *number* into the string variable *number\$*.
- 110 Finds the length of *number\$* and stores it in *length*.
- 120 Sets the variable *slice* to zero, before entering the following loop.
- 140 Adds one to *slice* each time round the loop.
- 150 Uses MID\$ to take one character at a time (depending on *slice*) from *number\$* and store it in *point\$*.
- 160 Sends the program back round the loop until *point\$* is a decimal point or every character of *number\$* has been examined.
- 170 Displays *number\$* on the screen using *slice* as an offset to ensure that the decimal point is always at *pointposition*.

Trevor Roberts



**There's really no need to go to such elaborate lengths to fill shapes on the Electron. DERRICK GARNER demonstrates a useful machine code routine**

# FILL IT UP—FAST!

**THIS machine code program fills a contrived shape in two and a half seconds and changes the colour of the fill.**

It consists of nine sub-routines, six of which move the graphics cursor around the screen inside the shape to be filled while the ninth does the actual filling.

The other two set and change the colour.

At the start of the six subroutines which move the cursor its starting position is stored in zero page locations &70 to &73.

The subroutine moves the cursor four pixels and then jumps to *LOOP* which fills the shape.

The subroutine then checks

to see if the current cursor position, on the Y axis, has reached the end of the part of the shape being filled.

If it has, the program moves to the next subroutine. If it hasn't it returns and repeats the routine.

The subroutines move the cursor four pixels at a time because, although the screen is theoretically divided into 1280 by 1024 pixels, the definition of the screen can only cope with four pixels at a time.

Moving one pixel at a time would still work but three quarters of the time the program would be doing nothing.

The six subroutines that move the cursor are all self

contained so they need not all be used to fill a desired shape, just select the ones that are of some use. The *LOOP* routine must be included.

I used the shape in the program because I considered that the routines used to fill it could be used to fill almost any shape such as the one in Figure 1.

To fill the shape in Figure 1 only the subroutines *VERTUP* and *VERTDOWN* need be used.

To use the program to fill this shape in red, delete lines 190 to 440, 660 to 1220 and 1430 to 1670.

Lines 110, 130, 140 and lines 1680 to 1790 *must* be retained whatever shape is being filled.

The first step is to move the graphics cursor at least eight pixels inside the shape to be filled, such as the bottom to position 136,64.

This start position is then put into the program at the start of *VERTUP* in lines 470 to 500.

```

470 LDA#136:STA#70\X Low
Byte (136 MOD 256)
480 LDA#64:STA#71\X High B
yte (136 DIV 256)
490 LDA#64:STA#72\Y Low B
yte (64 MOD 256)
500 LDA#0:STA#73\Y High B
yte (64 DIV 256)

```

Lines 600 and 630, which check the current cursor position on the Y axis, are then



set to the top of the shape being filled - 960 in this case.

```
600 CMP#3\Y High Byte (96
# DIV 256)
630 CMP#192\Y Low Byte (9
60 MOD 256)
```

These alterations will fill the top and bottom horizontal blocks and the left hand vertical block of the shape.

The same procedure is used for the VERTDOWN routine. The start position of the cursor at the top of the remaining unfilled block, position 1064.832 is put into the program in lines 1250 and 1280.

```
1250 LDA#40:STA#70\X Low B
yte (1064 MOD 256)
1260 LDA#4:STA#71\X High B
yte (1064 DIV 256)
1270 LDA#64:STA#72\Y Low B
yte (832 MOD 256)
1280 LDA#3:STA#73\Y High B
yte (832 DIV 256)
```

Lines 1375 and 1400, which check the current

position of the cursor on the Y axis, are then set to the bottom of the shape being filled - 188 in this case.

```
1375 CMP#0\Y High Byte (18
8 DIV 256)
1400 CMP#188\Y Low Byte (1
88 MOD 256)
```

The alterations needed are now complete for the Figure 1 shape. The same procedure must be carried out when using other subroutines to fill a different shape.

You may have noticed that in the original program: all the numbers inside the assembly routine were hexadecimal, whereas the numbers in the example program are decimal.

There is no reason for this other than I prefer to use hex. As some people might find it easier to use decimal I have used both types.

The program runs in Mode 5 but it will run in any of the graphics modes. It's full of REM statements to explain what is happening.

It also has a few lines which



Output produced by unaltered programs

are actually not needed in the original program but make it easier to use part of the program for other purposes.

I used part of the program to fill the upper case M in the October 1984 *Electron User* and knocked over 48 seconds off the time taken to fill.

If it is being used as part of another program utilising user defined characters or the function keys then line 130 should be changed to another address as the machine code is stored in the memory pages usually used for these purposes. **Derrick Garner**

### SUBROUTINES

150 .colour RED	Sets colour.
200 .DIAGDOWNLEFT	Moves the cursor.
460 .VERTUP	Moves the cursor.
660 .DIAGDOWNRIGHT	Moves the cursor.
920 .colour YELLOW	Changes colour.
970 .DIAGUPRIGHT	Moves the cursor.
1240 .VERTDOWN	Moves the cursor.
1430 .DIAGUPLEFT	Moves the cursor.
1700 .LOOP	Fills the line at current cursor position.

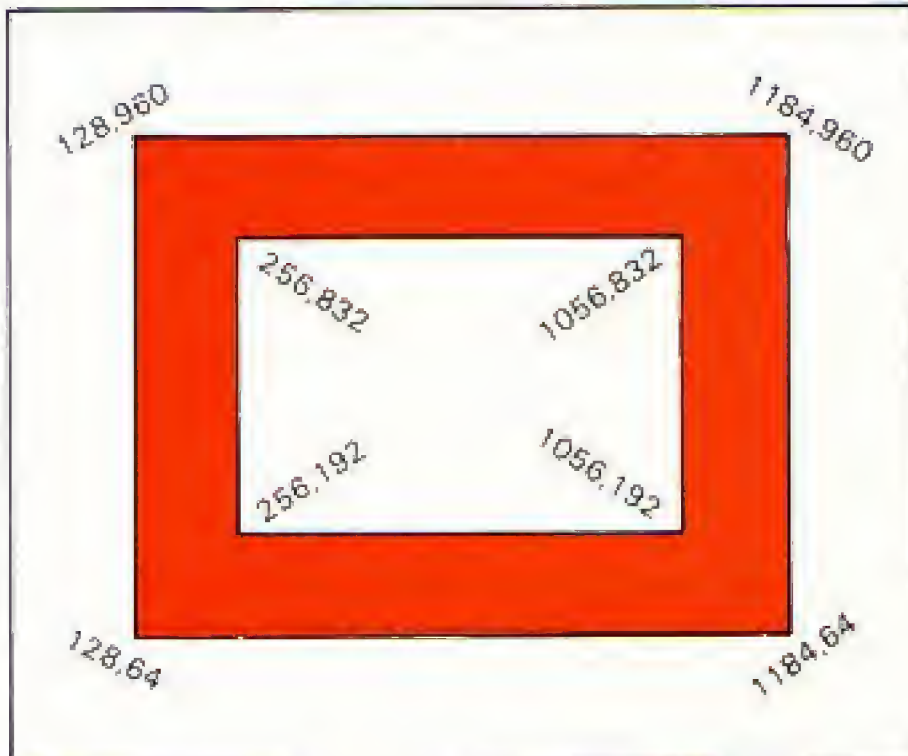


Figure 1: Example shape to be filled

```
10 REM ***** 240 LDA#&02:STA#73 \Y H
***** igh Byte
20 REM ** 250 .Sub1
** 260 JSR LOOP
30 REM ** DERRICK GAR 270 LDA#70 \Dec
NER ** rease X by 4
40 REM ** 280 SEC
** 290 SBC#&04
50 REM ** (C) Electron 300 STA#70
User ** 310 LDA#71
60 REM ***** 320 SBC#&00
***** 330 STA#71
70 REM****TURN OFF PLUS 340 LDA#72 \Dec
ONE**** rease Y by 4
80 *FX163,128,1 350 SEC
90 MODE5 360 SBC#&04
100 VDU23,1,0;0;0;0; 370 STA#72
110 FORI=0TO2STEP2 380 LDA#73
120 REM**ALL NUMBERS ARE 390 SBC#&00
HEXADECIMAL** 400 STA#73
130 PX=#B00 405 CMP#&00 \Che
140 (OPTI ck value of Y
150 .colourRED 410 BNE Sub1 \
160 LDA#&12:JSR#FFEE Hi Byte
170 LDA#&00:JSR#FFEE 420 LDA#72
180 LDA#&01:JSR#FFEE 430 CMP#&20 \Che
190 \ ck value of Y
200 .DIAGDOWNLEFT 440 BNE Sub1 \
210 LDA#&8B:STA#70 \X L Lo Byte
ow Byte 450 \
220 LDA#&02:STA#71 \X H 460 .VERTUP
igh Byte
230 LDA#&00:STA#72 \Y L
ow Byte
```

Turn to Page 28



# National Micro

## Electron price DOWN £70!

- and lots more cuts that make  
*Electron* computing the biggest  
bargain there's ever been!

Never before has there been such an opportunity to move into serious computing at such a low, low price. With your new *Electron* you'll find all you need to get you started - an introductory cassette of 15 programs, a fully-detailed User Guide, an easy-to-understand book on programming - **PLUS** a free dust cover that comes with the compliments of National Micro Centres

**£129**

*Carriage £7*

## BIG SAVINGS ON PRINTERS

	Old price	New price
Brother HR5	£170.95	£165.00
RX80	£272.00	£260.00
RX80FT	£315.00	£295.00
<i>Silver Read Typewriter/Printer</i>		
EX43 only	£286.90	£279.00
EX43+I/F	£394.25	£383.00

*Carriage £7*

## DATA RECORDER

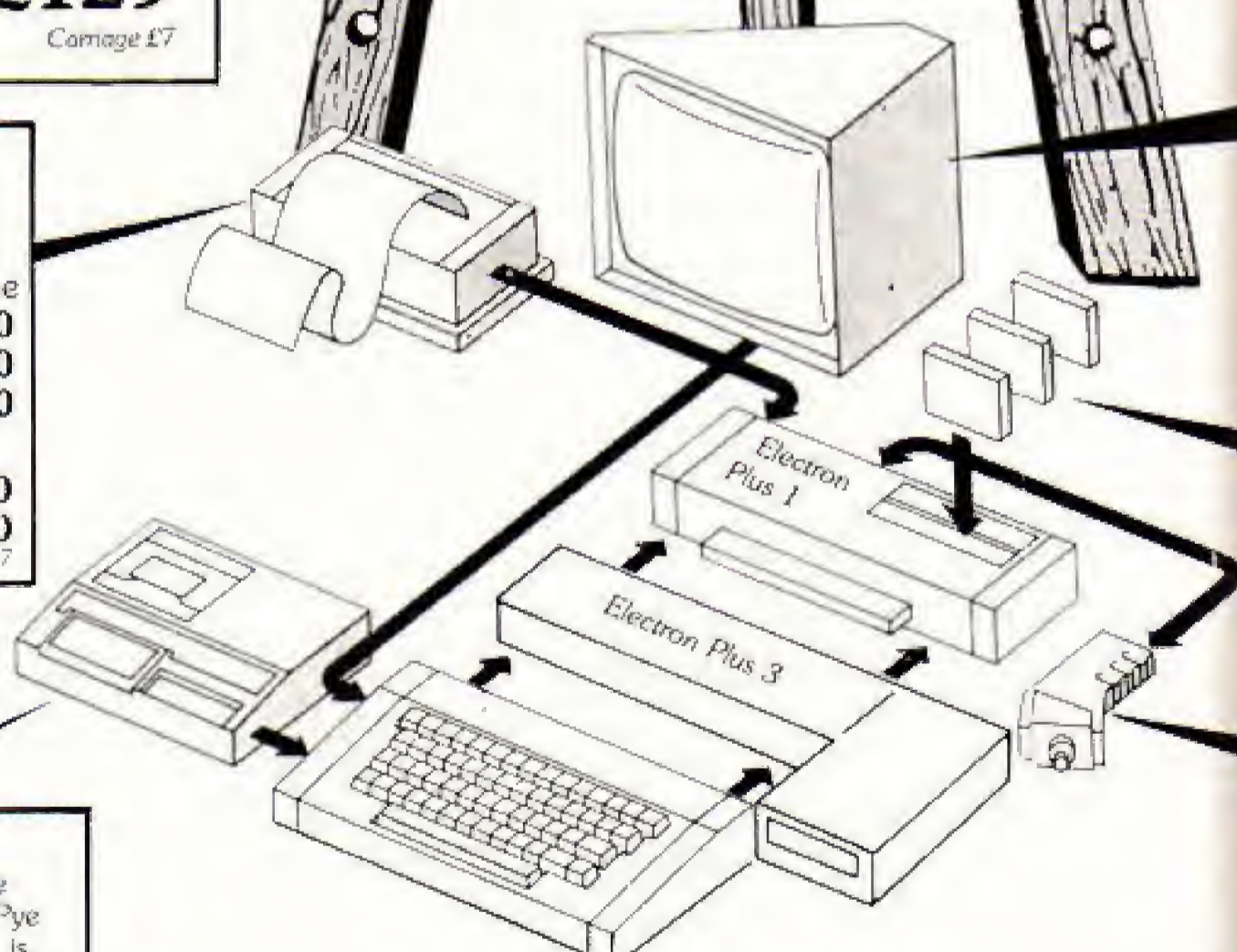
From a wide selection of cassette recorders we recommend the Pye Data Cassette Recorder, which is a perfect match for the *Electron*. With it comes a FREE power pack and *Electron* lead.

**£38.00**

*Carriage £7*

Plus 1  
+ Plus 3

— A new dimension in  
— *Electron* computing!



Call in for a demonstration from our friendly sales staff at:

**National Micro Centres**

**36-38 St. Petersgate**

**Stockport SK1 1HL**

or phone on:

**061-429 8080/8074**

## Shape listing

### From Page 25

```

470 LDA#188:STA#70 \X L
ow Byte
480 LDA#180:STA#71 \X H
igh Byte
490 LDA#180:STA#72 \Y L
ow Byte
500 LDA#180:STA#73 \Y H
igh Byte
510 .Sub2
520 JSR LOOP
530 LDA#72 \Inc
ease Y by 4
540 CLC
550 ADC#104
560 STA#72
570 LDA#73
580 ADC#100
590 STA#73
600 CMP#103 \Che
ck value of Y
610 BNE Sub2 \
Hi Byte
620 LDA#72
630 CMP#100 \Che
ck value of Y
640 BNE Sub2 \
Lo Byte
650 \
660 .DIAGDOWNRIGHT
670 LDA#140:STA#70 \X L
ow Byte
680 LDA#101:STA#71 \X H
igh Byte
690 LDA#140:STA#72 \Y L
ow Byte
700 LDA#103:STA#73 \Y H
igh Byte
710 .Sub3
720 JSR LOOP
730 LDA#70 \Inc
ease X by 4
740 CLC
750 ADC#104
760 STA#70
770 LDA#71
780 ADC#100
790 STA#71
800 LDA#72 \Dec
ease Y by 4
810 SEC
820 SBC#104
830 STA#72
840 LDA#73
850 SBC#100
860 STA#73
870 CMP#102 \Che
ck value of Y
880 BNE Sub3 \

```

```

Hi Byte
890 LDA#72
895 CMP#100 \Che
ck value of Y
900 BNE Sub3 \
Lo Byte
910 \
920 .colourYELLOW
930 LDA#112:JSR&FFEE
940 LDA#100:JSR&FFEE
950 LDA#102:JSR&FFEE
960 \
970 .DIAGUPRIGHT
980 LDA#100:STA#70 \X L
ow Byte
990 LDA#102:STA#71 \X H
igh Byte
1000 LDA#100:STA#72 \Y L
ow Byte
1010 LDA#102:STA#73 \Y H
igh Byte
1020 .Sub4
1030 JSR LOOP
1040 LDA#70 \Inc
ease X by 4
1050 CLC
1060 ADC#104
1070 STA#70
1080 LDA#71
1090 ADC#100
1100 STA#71
1110 LDA#72 \Inc
ease Y by 4
1120 CLC
1130 ADC#104
1140 STA#72
1150 LDA#73
1160 ADC#100
1170 STA#73
1180 CMP#103 \Che
ck value of Y
1190 BNE Sub4 \
Hi Byte
1200 LDA#72
1210 CMP#100 \Che
ck value of Y
1220 BNE Sub4 \

```

```

Lo Byte
1230 \
1240 .VERTDOWN
1250 LDA#120:STA#70 \X L
ow Byte
1260 LDA#104:STA#71 \X H
igh Byte
1270 LDA#140:STA#72 \Y L
ow Byte
1280 LDA#103:STA#73 \Y H
igh Byte
1290 .Sub5
1300 JSR LOOP
1310 LDA#72 \Dec
ease Y by 4
1320 SEC
1330 SBC#104
1340 STA#72
1350 LDA#73
1360 SBC#100
1370 STA#73
1375 CMP#100 \Che
ck value of Y
1380 BNE Sub5 \
Hi Byte
1390 LDA#72
1400 CMP#120 \Che
ck value of Y
1410 BNE Sub5 \
Lo Byte
1420 \
1430 .DIAGUPLEFT
1440 LDA#100:STA#70 \X L
ow Byte
1450 LDA#104:STA#71 \X H
igh Byte
1460 LDA#100:STA#72 \Y L
ow Byte
1470 LDA#100:STA#73 \Y H
igh Byte
1480 .Sub6
1490 JSR LOOP
1500 LDA#70 \Dec
ease X by 4
1510 SEC
1520 SBC#104
1530 STA#70

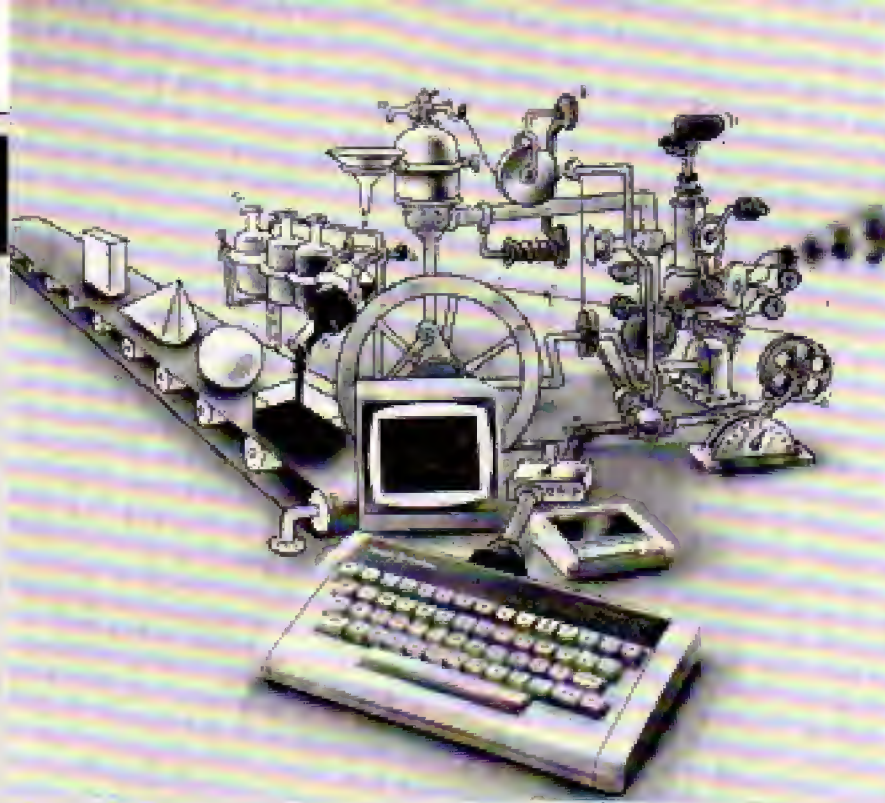
```

```

1540 LDA#71
1550 SBC#100
1560 STA#71
1570 LDA#72 \Incr
ease Y by 4
1580 CLC
1590 ADC#104
1600 STA#72
1610 LDA#73
1620 ADC#100
1630 STA#73
1640 CMP#102 \Che
ck value of Y
1650 BNE Sub6 \
Hi Byte
1660 LDA#72
1665 CMP#100 \Che
ck value of Y
1670 BNE Sub6 \
Lo Byte
1680 BED DONE \Fini
shed
1690 \
1700 .LOOP \PLOT
77,X,Y
1710 LDA#119:JSR&FFEE
1720 LDA#140:JSR&FFEE
1730 LDA#70:JSR&FFEE
1740 LDA#71:JSR&FFEE
1750 LDA#72:JSR&FFEE
1760 LDA#73:JSR&FFEE
1770 .DONE RTS \Back
to BASIC
1780 J
1790 NEXT
1800 REM****BASIC DRAW PR
OG*****
1810 REM
1820 MOVE128,32:DRAW128,99
2:DRAW256,992:DRAW648,592:D
RAW1056,992:DRAW1184,992:DR
AW1184,32:DRAW1056,32:DRAW6
48,432:DRAW256,32:DRAW128,3
2
1830 MOVE256,192:DRAW256,8
32:DRAW576,512:DRAW256,192
1840 MOVE736,512:DRAW1056,
832:DRAW1056,192:DRAW736,51
2
1850 MOVE648,432:DRAW648,5
92
1860 CALL#800
1870 END

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*



# SLOWLY- BUT SURELY

*Slomo*  
Nidd Valley Micro Products

I MUST admit that when I first got the Slomo I was more than a little puzzled. Why should Nidd Valley Micro Products go to the trouble of producing a piece of hardware that slows down the Electron? I couldn't see the point.

However after a couple of hours messing around with it I soon got the idea, and now I wouldn't be without one.

The Slomo is a small, neat box that attaches by cable to the Electron's expansion port. On top of the box are a small knob and two buttons, one marked Freeze frame, the other Slow motion.

Using these, the Slomo can slow down or even stop any program running in the Elec-

tron. As a consequence the screen display also slows down or stops.

As I said, at first I couldn't see the point, but after trying it out I was soon convinced of its usefulness.

My first idea was, as usual, entirely selfish. Working for *Electron User* I tend to get a lot of games to review and the truth is that as I get older the games are getting faster.

Not any more though, I can use the Slomo to cheat. By pressing the slow motion button and turning the knob I can adjust the speed of the game to a rate I can manage.

I can even freeze the frame, either to take off-screen photos, spy out where the hazards are, or just to take a breather.

And, of course, it's not just slow-witted adults who'll benefit. The Slomo can be used to tailor the speed of games to



growing children's abilities. And more importantly it will be a boon to the handicapped and to teachers in special schools.

It's one of those pieces of equipment that you keep finding uses for. I've used its slow motion facility to try to figure out how a program works and also in debugging my own.

From initially wondering why anyone had brought it out, I was soon wondering why no one had done it before!

It's an excellent piece of equipment, full of potential.

The instructions are more than adequate - and a lesson to other hardware manufacturers.

It fits straight onto the back of the Electron and works perfectly with no problems. I can recommend it thoroughly and I'm sure that I've not tapped all of its uses.

There is one drawback. From now on when someone tells us that they got 23,000 on Cylon Attack, will they be honest enough to admit it was Slomo assisted?

**Trevor Roberts**

ELECTRON, BBC Model B  
(any OS, BASIC I/II)

## QUAL-SOFT

£9.95  
(inc. VAT and p.p.)

"There is one fault though. I am going to lose a lot of sleep over it, it is so addictive". Steven Wiseman of Liverpool.

"Many thanks for the fantastic game. As soon as I received it, there was no stopping until the end of the season". J. Hooley of Twickenham.

"I am writing to say what a wonderful football program SOCCER SUPREMO is. It really is the best football game on the market at the moment". Anthony Hayes of Redcar.

# "SOCCER SUPREMO"

NOT SO MUCH A GAME, MORE A WAY OF LIFE!

You have just been appointed Manager of a newly promoted 1st Division Club, and it is up to you to transform this very ordinary side into one that can realistically challenge for the 1st Division Championship within the next 5 seasons. You must assess your side's capabilities and then, through your youth policy and the transfer market, reinforce the strengths and eliminate the weaknesses. It's all so easy . . . or is it?

\*\*\* "3-D", 22 MAN, FULL PITCH, FULL MATCH GRAPHICS SIMULATION

- 42 match season, 21 home games, 21 away games.
- Transfer market (Rush, Robson, Hoddle etc).
- 4-4-2, 4-3-3 and 4-2-4 team formation.
- In match tactical adjustments.
- Opposition skills related to League record.
- Opposition: 21 of the current 22 DIV 1 sides
- Match injuries: Your physio reports
- Team selection by names. (enter initials)
- Home/away bias, opposition tactical play
- Tactical substitutions

And many more features, but will take a full page advert if we are to continue. (That'll be O.K. Ad. Man).

The game will be posted on the same day as the receipt of order. ACCESS telephone authorisations should take no more than two days to arrive.

**QUAL-SOFT**  
Dept. EU.  
18, Hazlemere Rd.,  
Stevenage,  
Herts. SG2 8RX  
Tel: (0438) 721936

Please supply a copy of SOCCER SUPREMO. I enclose a cheque, postal order, ACCESS card authorisation for £9.95

(Please state Electron or BBC)

Name: .....

Address: .....

CARD NO: .....

# SCRAPBOOK

SCRAPBOOK is the feature that contains a selection of all the short, simple programs sent in by our readers.

It's where we keep a record - our scrapbook - of all the interesting little routines that don't end up in the Notebook or in Program Probe but are too good for us not to share.

This month it's very much a graphics show. Next month - who knows? It's up to you.

So if you enjoy messing about with your Electron and want to share your discoveries with other Electron users, send them in to us.



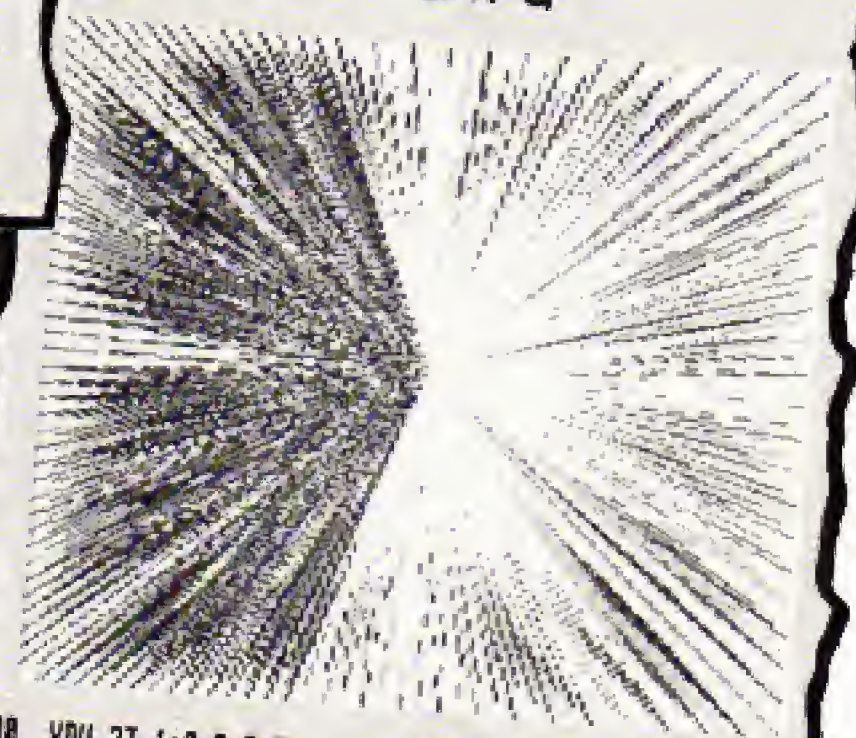
## Simple Circles by Tony Remmer

```

10 REM RANDOM CIRCLES
20 REM BY TONY REMMER
30 REM BLANDFORD, DORSET
40 MODE 2
50 VDU 23;B202;0;0;0;
60 X=RND(1000);Y=RND(10
80)
70 GCOL 0,RND(Y)
80 FOR F=-150 TO 150 ST
EP 4
90 L=INT((SQR(ABS(150+1
50)-(F*F))))+.5)
100 MOVE X-F,Y+L:DRAW X-
F,Y-L
110 NEXT F
120 GOTO 60
    
```

Send your programs to  
Scrapbook, *Electron*  
User, 68 Chester Road,  
Hazel Grove, Stockport  
SK7 5NY.

## Paintshop Pyrotechnics in full colour By Trevor Bird



```

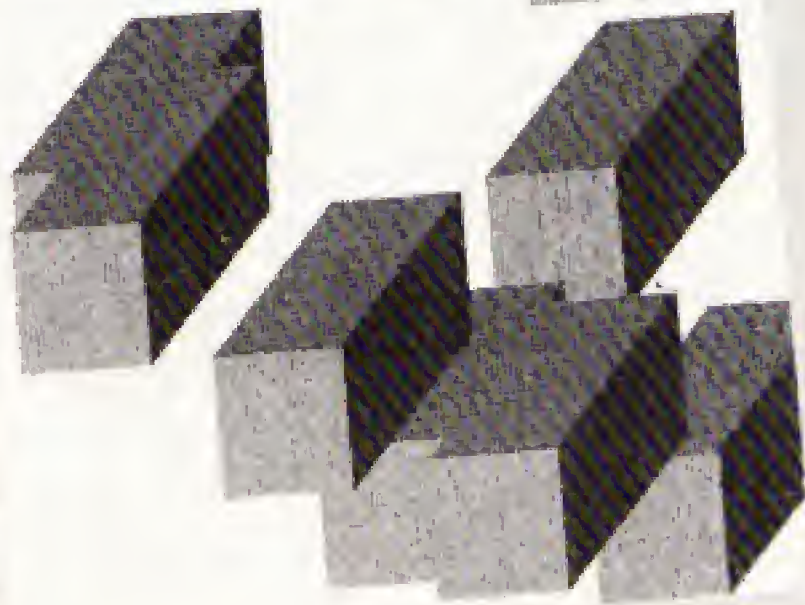
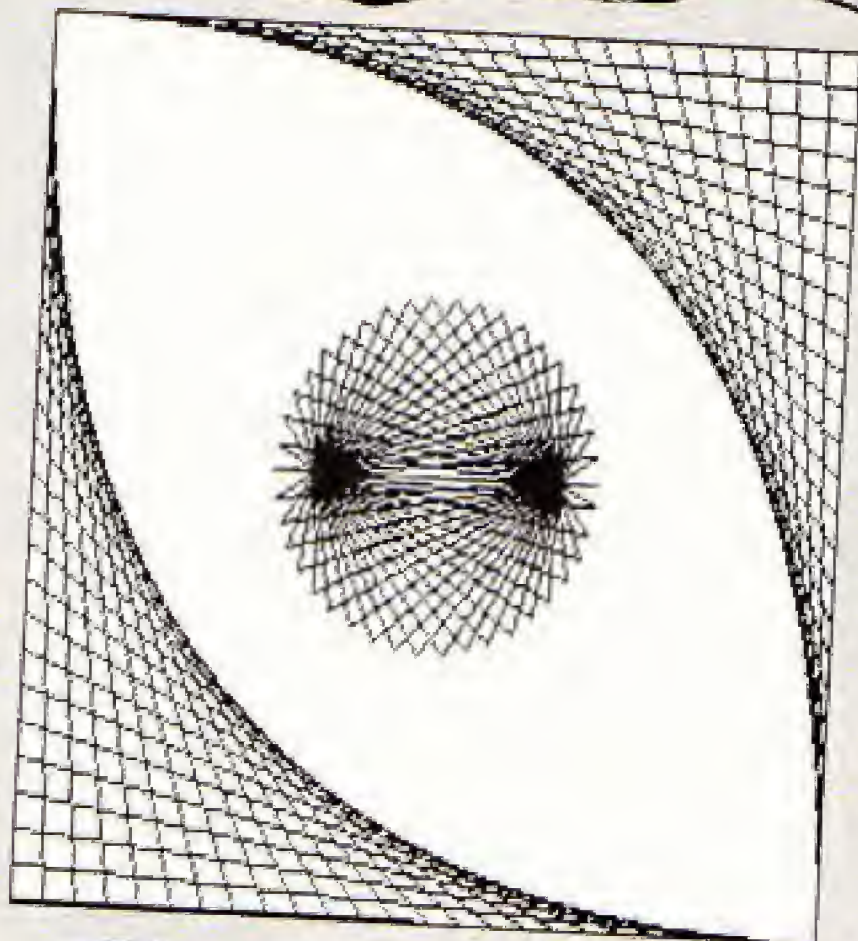
10 REM PAINTSHOP EXPLOSI
ON
20 REM BY TREVOR BIRD
30 REM STEVENAGE HERTS
40 MODE 2
50 PROCpaintshop
60 END
70 DEF PROCpaintshop
80 VDU 23,1;0;0;0;0
90 VDU 29,640;512;
100 MOVE 0,0
110 NZ=0:BX=0
120 FOR X=-640 TO 640 STE
P 32
130 FOR Y=-512 TO 512 STE
P 25
140 DRAW X,Y
150 MOVE 0,0
160 GCOL B1,N1
170 NZ=(NZ+1) MOD 8
180 NEXT: NEXT
190 B1=B1+1
200 IF B1>4 THEN ENDPROC
210 GOTO 120
    
```

*P. Charlesworth uses  
MOVE and DRAW to  
keep an eye on you*

```

10 REM EYE
20 REM BY P.Charlesworth
30 REM WEST YORKSHIRE
40 MODE4
50 VDU 19,1,0;0;
60 VDU 19,0,6;0;
70 VDU 23,1,0;0;0;0;
80 VDU 29,139;11;
90 MOVE 0,0
100 DRAW 0,1000
110 DRAW 1000,1000
120 DRAW 1000,0
130 DRAW 0,0
140 FOR A=1 TO 1000 STEP
40
150 MOVE 0,A
160 DRAW 1000-A,0
170 NEXT A
180 FOR A=1000 TO 1 STEP
-40
190 MOVE 1000,A
200 DRAW 1000-A,1000
210 NEXT A
220 FOR C=0 TO 2*PI STEP
0.15
230 MOVE 650,500
240 DRAW 500+COS C*200,500
+SIN C*200
250 MOVE 350,500
260 DRAW 500+COS C*200,500
+SIN C*200
270 NEXT C
280 VDU 7

```

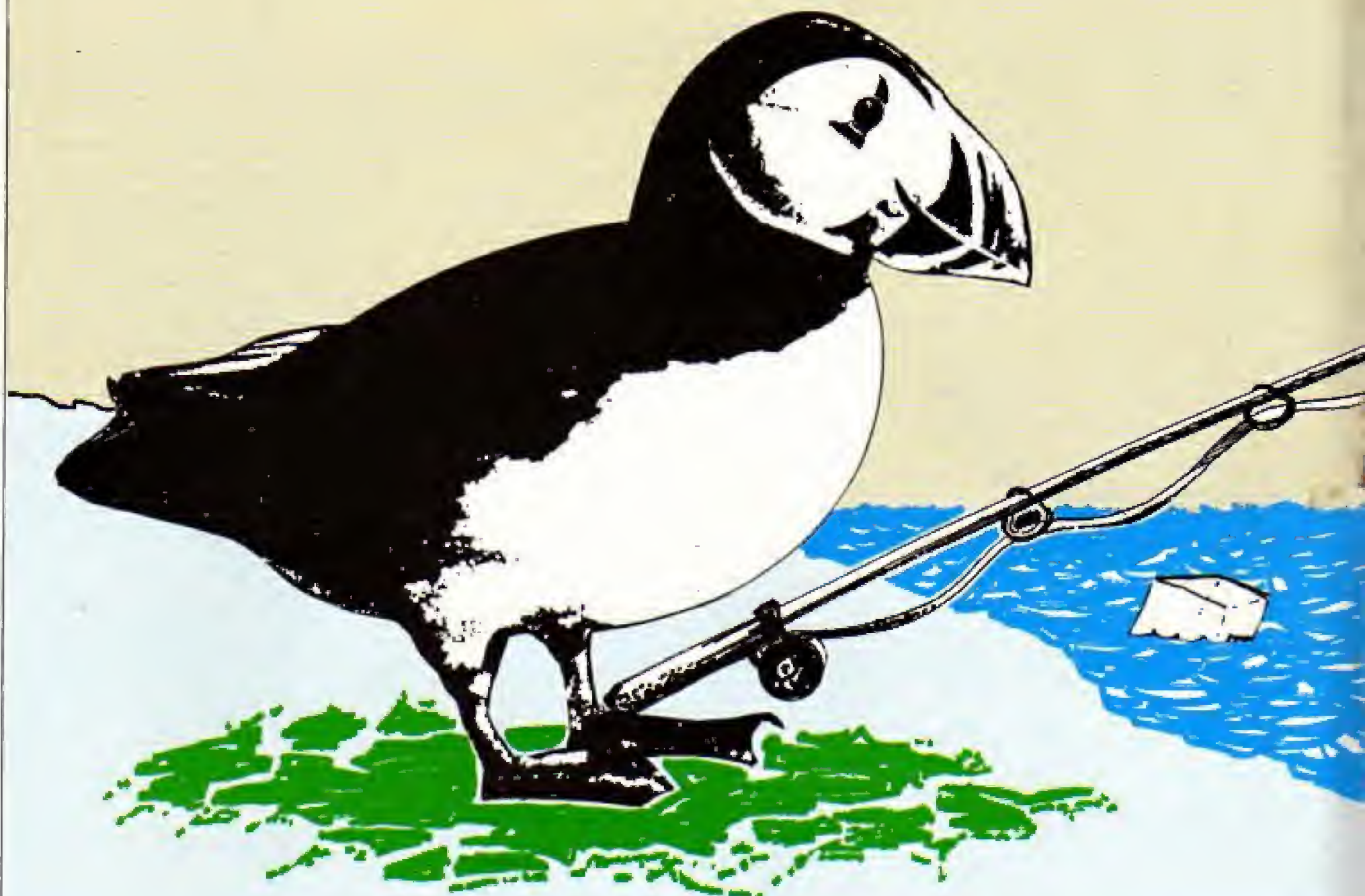


*Douglas Hodgson  
uses his  
Electron's blocks!*

```

10 REM 3D-CUBES
20 REM BY DOUGLAS HODGSON
30 REM KESWICK,CUMBRIA
40 MODE 1
50 VDU 23,1,0;0;0;0;
60 delay% = 2000
70 FOR B% = 0 TO 25
80 AX = RND(1000); CI = RND(1
000)
90 GCOL 0,1
100 MOVE AX,CI; MOVE AX+20
0,CI
110 PLOT 85,AX,CI+200
120 MOVE AX,CI+200; MOVE A
X+200,CI+200
130 PLOT 85,AX+200,CI
140 GCOL 0,2
150 MOVE AX,CI+200; MOVE A
X+200,CI+200
160 PLOT 85,AX+400,CI+400
170 MOVE AX+400,CI+400; MO
VE AX+200,CI+400
180 PLOT 85,AX,CI+200
190 GCOL 0,3
200 MOVE AX+200,CI; MOVE A
X+200,CI+200
210 PLOT 85,AX+400,CI+200
220 MOVE AX+400,CI+200; MO
VE AX+400,CI+400
230 PLOT 85,AX+200,CI+200
240 VDU 19,1,RND(7),0,0,0
250 VDU 19,2,RND(7),0,0,0
260 VDU 19,3,RND(7),0,0,0
270 SOUND 1,-15,RND(255),
1
280 NEXT B%
290 VDU 19,1,RND(7),0,0,0
300 VDU 19,2,RND(7),0,0,0
310 VDU 19,3,RND(7),0,0,0
320 FOR delay% = 0 TO delay%
NEXT delay%
330 GOTO 290

```

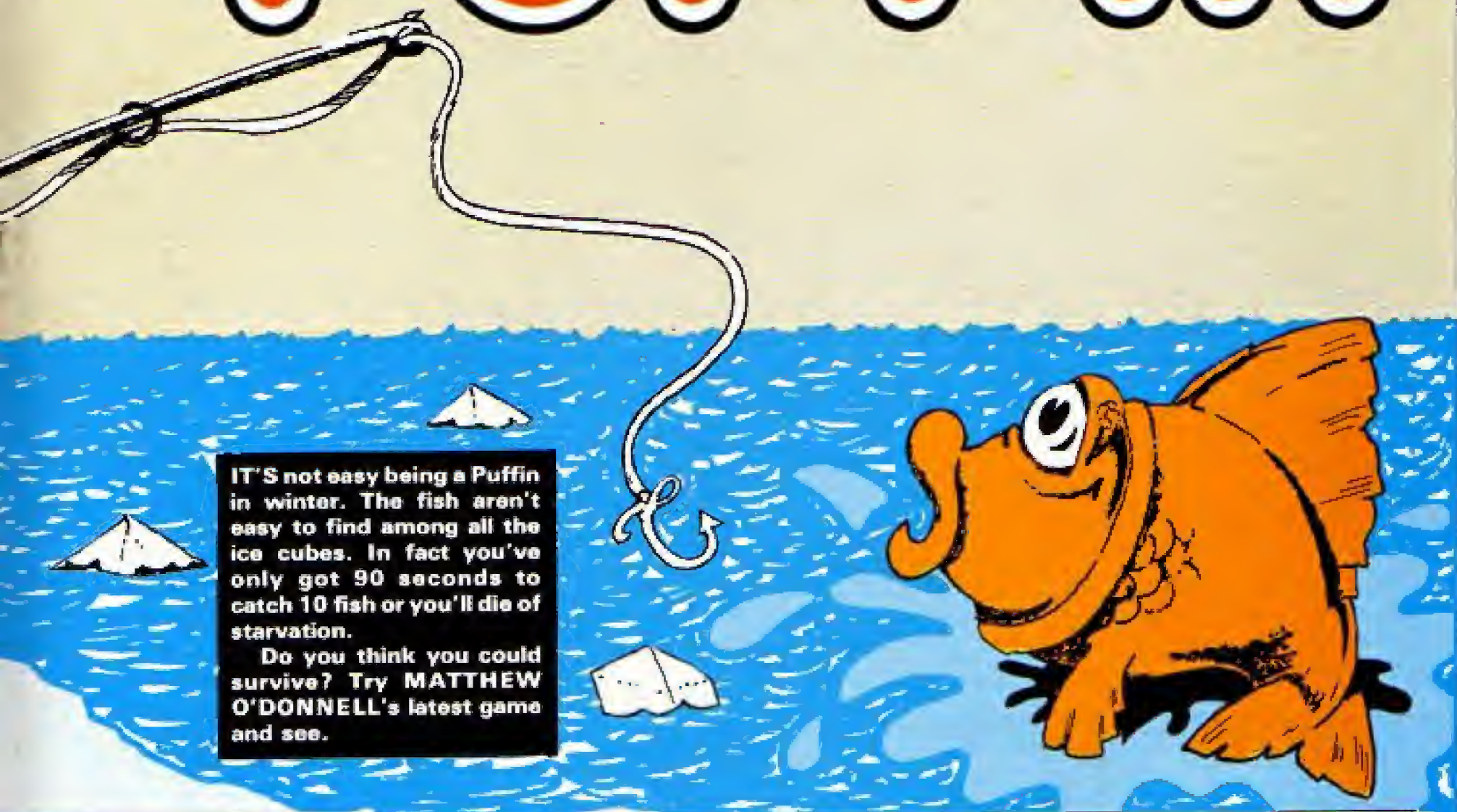


```

10 REM ***** PUFFIN ***
***
20 REM ** By Matthew O'Donnell **
30 REM (C) ELECTRON USE
R
40 #FX229,1
50 MODE1:VDU23;B202;0;0;
0;:PROCinit
60 PROCtune
70 MODE1:VDU23;B202;0;0;
0;
80 REPEAT
90 TIME=0
100 VDU19,3,6,0,0,0
110 COLOUR2:PROCdraw
120 COLOUR3:PROCBLOCKS
130 REPEAT
140 IF NOT BFX PROCfish
150 PROCmove
160 PROCab
170 UNTILfish=10
180 PRDctune2:CLS:PROCend
190 DEFPROCmove
200 IF INKEY(-98) X1=X1-1
:F=1:SOUND1,2,50,1:GOTO240
210 IF INKEY(-67) X1=X1+1
:F=2:SOUND1,2,50,1:GOTO240
220 IF INKEY(-73) Y1=Y1-1
:SOUND0,-15,4,1
230 IF INKEY(-105) Y1=Y1+1
:SOUND0,-15,4,1
240 IF X1<1 X1=1
250 IF X1>30 X1=30
260 IF Y1<1 Y1=1
270 IF Y1>30 Y1=30
280 IF POINT((X1*32)+4,(3
1-Y1)*32)<>0X1=XX;Y1=YY
290 PRINTTAB(XX,YY)S#
300 COLOUR1:IF F=2 PRINTT
AB(X1,Y1)PUFFIN# ELSE PRINT
TAB(X1,Y1)PUFFIN2#
310 IF TIME/100 >=90 THEN
PROCFaint:CLS:PROCend:RUN
320 IF X1=6X1 AND Y1=6Y1
THEN fish=fish+1:FORP=1T025
5 STEP 52:SOUND1,2,P,1:NEXT
:BF1=FALSE
330 IF TIME/100>=75 AND W
=TRUE THEN SOUND1,-15,200,1
0:W=FALSE
340 XX=X1:YY=Y1
350 ENDPROC
360 DEFPROCdraw
370 MOVE30,30:DRAW30,994:
DRAW1250,994:DRAW1250,30:DR
AW30,30
380 ENDPROC
390 DEFPROCBLOCKS
400 FORP=1TOSK1+70:PRINTT
AB(RND(30),RND(30))BLOCK#
410 NEXT
420 ENDPROC
430 DEFPROCinit
440 VDU23,224,100,56,16,5
6,124,124,56,40
450 VDU23,225,20,60,196,6
8,26,26,26,122
460 VDU23,226,56,34,35,34
,88,88,88,94
470 VDU23,227,0,32,113,1,
175,143,240,15
480 VDU23,228,126,195,153
,165,165,153,195,126
490 ENVELOPE1,1,1,-2,1,7,
5,12,126,0,0,-126,126,126
500 ENVELOPE2,1,0,-0,0,4,
4,4,126,0,0,-126,126,126
510 TIME=0
520 F=0
530 W=TRUE
540 SK1=0
550 LSZ=90
560 BFX=FALSE
570 fish=0
580 X1=9:Y1=15
590 XX=X1:YY=Y1
600 6X1=0:6Y1=0
610 FISH#=CHR#224
620 PUFFIN#=CHR#226
630 PUFFIN2#=CHR#225
640 S#=CHR#32
650 BLOCK#=CHR#228
660 VDU19,3,6,0,0,0
670 COLOUR1:PRINTTAB(17,1
)"PUFFIN"
680 COLOUR2:PRINTTAB(17,2
)"*****"
690 COLOUR3
700 PRINT"Guide your puff
in around the ever ""chan
ging maze of ice cubes coll
ecting""fish. You have 90
seconds to collect ""all
ten fish before your puffi
n starves"
710 PRINT""to death. You
will here a warning beep""

```

# PUFFIN



IT'S not easy being a Puffin in winter. The fish aren't easy to find among all the ice cubes. In fact you've only got 90 seconds to catch 10 fish or you'll die of starvation.

Do you think you could survive? Try MATTHEW O'DONNELL's latest game and see.

"when you have only 15 seconds left."

```

720 COLOUR1
730 PRINT "The keys are:
";SPC(9);"Z .... LEFT";
SPC(9)"X .... RIGHT";SPC(
9);" .... UP";SPC(9);"/
.... DOWN"
740 PRINTTAB(28,19)PUFFIN
";SPC(3);:COLOUR2:PRINT"Puf
fin"
750 COLOUR3:PRINTTAB(28,2
2)BLOCK#;SPC(3);:COLOUR2:PR
INT"Ice Cube"
760 COLOUR2:PRINTTAB(28,2
5)FISH#;SPC(3);"Fish"
770 COLOUR2:REPEAT PRINTT
AB(11,30)"SKILL LEVEL (1-9)
";:INPUTSKX:UNTILSKX<=9 AND
SKX>=1
780 PRINTTAB(11,30)"PRESS
SPACE TO START ";REPEATUNT
ILGET=32
790 ENDPROC
800 DEFPROCmb
810 COLOUR3
820 REPEAT

```

```

830 BXX=RND(36)+1:BYX=RND
(30)
840 UNTILBXX<>YX AND BYX<
>YX
850 IF BXX<>GX AND BYX<>
GY THEN PRINTTAB(BXX,BYX);
:ELSE ENDPROC
860 IF RND(4)=1 PRINTBLOC
K#:ELSE VDU32
870 ENDPROC
880 DEFPROCfish
890 BFX=TRUE
900 BXX=RND(36)+1:BYX=RND
(30)
910 COLOUR2:PRINTTAB(GXX,
GYX)FISH#
920 ENDPROC
930 DEFPROCend
940 IF fish=10 COLOUR1:PR
INTTAB(4,3)"WELL DONE!!!";
:COLOUR2:PRINT"You collecte
d all ten fish." "It took
you ";TIME/100;" seconds."
";:COLOUR1:PRINT SPC(8)
";"PRESS SPACE TO PLAY AGAIN
";:REPEATUNTILGET=32:RUN
950 COLOUR1:PRINT "Your p

```

```

uffin has starved to death!
!!!""You collected ";fish
";" fish."";:COLOUR2:PRINT
SPC(5);"PRESS SPACE TO PLAY
AGAIN";:REPEATUNTILGET=32
960 ENDPROC
970 DEFPROCfaint
980 COLOUR1:PRINTTAB(X,Y
X)CHR#227
990 FORP=@T0100:NEXT
1000 RESTORE1060
1010 FORP=1T011
1020 READ PZ,DX
1030 SOUND1,1,PZ,DX
1040 SOUND1,0,0,1
1050 NEXT
1060 DATA68,8,68,6,68,2,68
,8,88,6,76,2,76,4,68,4,68,6
,64,2,68,8
1070 ENDPROC
1080 DEFPROCtune
1090 RESTORE1160
1100 FORP=1T017
1110 READ AX,BX
1120 IF AX=0 VX=0 ELSEVX=1
1130 SOUND1,VX,AX,BX
1140 SOUND1,0,0,1

```

```

1150 NEXT
1160 DATA136,2,128,2,136,1
2,128,2,128,2,116,2,108,2,1
04,8,108,16,0,2,136,2,128,2
,136,12,116,6,128,6,104,6,1
08,8
1170 ENDPROC
1180 DEFPROCtune2
1190 RESTORE1260
1200 FORP=1T025
1210 READ QZ,MZ
1220 IF QZ=0 CZ=0 ELSECZ=1
1230 SOUND1,CZ,QZ,MZ
1240 SOUND1,0,0,1
1250 NEXT
1260 DATA108,1,116,1,108,1
,108,1,96,1,88,1,88,2,88,1,
96,1,88,1,88,1,76,1,68,1,68
,2,108,1,116,1,108,1,108,1,
96,1,88,1,88,1,96,1,68,2,76
,4,88,4
1270 ENDPROC

```

This listing is included in this month's cassette tape offer. See order form on Page 61.



**WELCOME** to the first in a more-or-less regular series of columns for adventurers – especially frustrated adventurers.

Why frustrated? Because one of the purposes of this column is to try to help you if you are stuck. And we all are at one time or another.

However don't you experts lose interest, I expect you to do your bit by sending in maps for adventures you have completed or solutions to difficult problems you have solved.

I also hope to be a sounding board for your ideas and opinions on adventures. So if you have something to say, write in.

Sooner or later I hope to compile a Top Ten for adventures, so I'm relying on you to award marks for each one you try. I suggest you award marks out of 100 (it makes it easier for me).

To give you some idea of what I mean, my nomination for the adventure of 1984 is Epic's Wheel of Fortune. I



would award marks for it as follows:

Presentation .....	6/10
Contents .....	28/30
Value for money ....	27/30
Frustration factor!	29/30
<b>Total .....</b>	<b>90/100</b>

The categories I have used are only suggestions. What counts is the total mark. All the marks sent in will be averaged out and a Top Ten published in a future issue.

Try to be honest with your marks – don't give it 100 just because you managed to finish it!

Our Top Ten, combined

with our reviews, will then be the best possible way for you to judge whether a particular adventure is the one you want or not.

Incidentally I'm also prepared to help BBC owners who get stuck.

A warning. I won't be using any codes or letter-transposition routines, so if you don't want to see the answers don't read the last part of this article. This month we'll be giving hints on Adventure and Eye of Zoltan.

Now having said how we're going to help you, I'd like some help myself with Kingdom of

Klein. How do I get off the pile of rocks and how do I get into the mountain? And in Quest for the Holy Grail, how do I open the castle doors?

Finally, if you want an immediate answer to a problem, enclose an sae – I'll reply, if only to say I don't know either!

*And now to some hints. Mrs Bull wants to know how to get into the castle in Program Power's Adventure. Rub the lamp in the right place. Mrs E. Young is having problems with Sottek's Eye of Zoltan. To get the keys, take a book then look. As for the stone doors, this is the Temple of Zoltan. You don't need to go there yet! The keys are the key (sorry!).*

**If you want Merlin's help write to:**  
**Merlin, Electron User,**  
**Europa House, 68 Chester**  
**Road, Hazel Grove,**  
**Stockport SK7 5NY.**

# FOUR GAMES ON ONE! 'BUMPER BUNDLE'

**ELECTRON  
 BUMPER  
 BUNDLE**

4 GAMES ON 1 TAPE

How would you fancy a journey in space, followed by a trip to the darkest depths or a close encounter in the insect world, to be finally rewarded with untold riches? This latest Alligata special,

Alligata Software Ltd 1 Orange Street, Sheffield S1 4DW  
 Tel: (0742) 755796

a compilation of Bugblaster, Lunar Rescue, Hell Hole and Crown Jewels, guarantees many hours of action packed fun and enjoyment.

Send for full colour product leaflet (enclose a stamp)

**£7.95**

**Alligata**  
 Software Limited

Available through good  
 computer stores everywhere

**GOOD** news for Electron users who are frustrated by the slow speed and unreliability of cassettes – the Plus 3 disc expansion unit from Acorn has arrived and brought with it new dimensions of speed and reliability in the storage of programs and data.

In appearance the Plus 3 is fairly unprepossessing. It consists of an L shaped box coloured the usual shade of Acorn cream.

The long arm of the L contains the disc interface that allows communication between the Electron and the disc drive.

This attaches to the expansion bus at the back of the Electron while the short arm (which is the drive itself) curves neatly round the right hand side of the machine.

The result is an Electron that is some three inches wider with a disc drive to the right of the keyboard. As the Electron with the Plus 3 needs so much more power than the unexpanded version it comes with its own separate power supply – the old one becomes redundant.

And Plus 1 owners needn't worry. There is an expansion bus for it on the back of the Plus 3.

Appearances can be deceptive. The Plus 3 may look fairly dull but once you get to grips with it you find a fascinating piece of equipment offering a wealth of possibilities.

The obvious attraction of using discs to store programs and data is their speed.

A program that might take minutes to save to or load from tape only takes seconds using disc storage. And the time you spent waiting for cassettes to load and save can be used for programming.

Another benefit of the increased speed is that you



## ... and then there was Plus 3

can take on jobs using discs that you would never dream of when you're confined to tape storage.

Loading and saving the contents of a screen display during a program is one example.

With discs it's a quick job, but with tape the program could literally take hours.

Along with the faster speed of discs comes increased reliability. With them you rarely encounter the loading

just treated as a sort of super cassette system which uses discs instead of tapes.

Of course there's more to it than that – and you'll be able to read lots more about the subject in future issues of *Electron User* – but the simplicity and reliability of the discs and the drives they go into means you don't have to know much about them to use them.

You're not limited to one drive, either. It's possible to

own to make use of the increased potential of discs.

Examples are \*MAP, which gives information about how the disc space is used and \*DELETE. There's even \*TAPE which tells the Electron to use the cassette for storage.

These extra commands and what they can be used for are explained in the thorough, well written user guide that comes with the Plus 3.

Despite a few hiccups – it talks about the CSD and CSL before it explains them – it covers a fairly complex subject well, giving all the information needed for users to exploit the Plus 3's potential to the full.

To augment these extra commands the Welcome disc that comes with the Plus 3 contains a library of utilities designed to ease the life of disc users.

Easy to use and well explained on screen, these utilities are a lesson in user friendly programming.

All in all the Plus 3 is an excellent piece of work. Well made, easy to fit and well documented. Even a complete newcomer to discs will have no trouble using it for loading and saving programs.

And the expert will be delighted at the range and power of the available ADFS commands.

In fact it's a far better disc system than the official one for the BBC. The only quibble is the price, £229, which seems rather high.

Apart from this, the Plus 3 is an excellent addition to the Electron. If you can afford it, buy it.

**Nigel Peters**

---

## ‘With discs you rarely encounter the loading and saving problems that crop up with cassettes’

---

and saving problems that crop up with cassettes.

And when you do it's more likely that you've mistreated the disc rather than that the system is at fault.

Not only is a disc more reliable, you can store more on it. Each of the single sided 3½ inch, 80 track discs used by the Plus 3 can theoretically hold up to 320k.

Some versions of the Plus 3 allow both sides of the disc to be used, allowing 640k of information to be stored.

When you consider that that's 10 times the memory capacity of the Electron itself you'll see how much a disc can hold.

The Plus 3 can be looked on as a combination disc drive and interface.

The disc drive itself is fairly straightforward and can be

attach another drive via a standard connection and, if wanted, this can be one of the larger 5¼ inch drives.

The disc interface or the Acorn Advanced Disc Filing System (ADFS) as it is more properly known is the link between the Electron and the drive.

Once the ADFS is fitted, the Electron automatically uses the disc drive for storage.

Normal Basic commands such as LOAD, SAVE and CHAIN can still be used, but now the program will be saved to or loaded from disc instead of cassette.

Similarly, \*CAT tells you the contents of the disc rather than of the cassette.

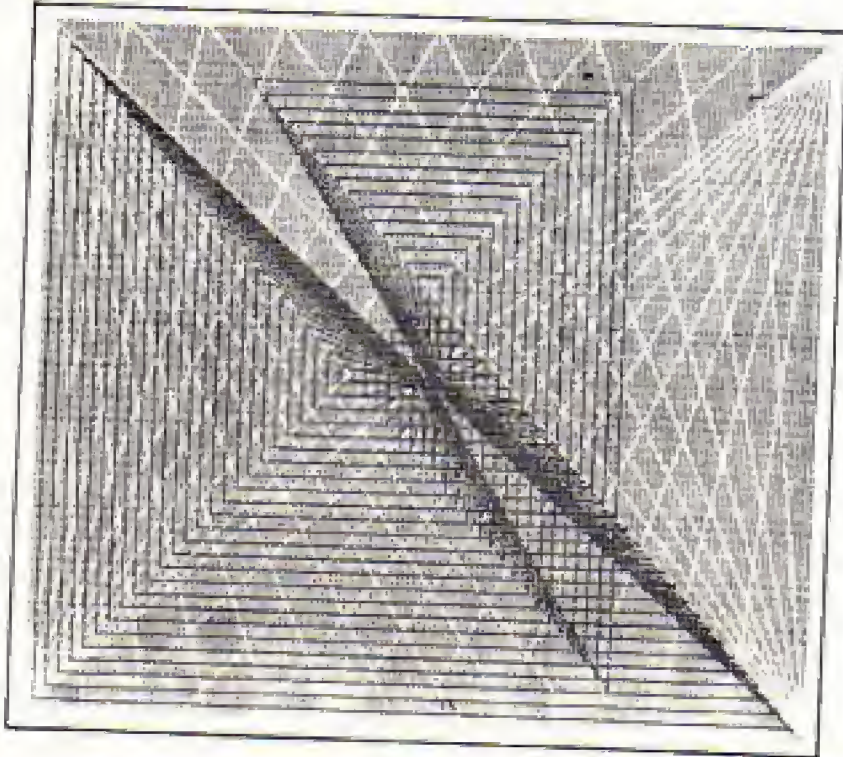
However discs are a lot more than just super cassettes, and the ADFS has a whole set of commands of its



# Triangle turnover

**CLEVER** use of the Electron's **MOVE** and **DRAW** commands combine to produce Triangle Turnover, a graphics listing from

**CHRIS WILD SMITH**. It's hard to believe that such a complicated pattern is made up of just straight lines.



```

10REM TRIANGLE TURNOVER
20REM BY CHRIS WILDSMITH
30REM (C) ELECTRON USER
40MODE 1
50VDU23,1,0;0;0;0;
60SCOLO,4
70COLOUR 129
80CLS
90FOR S=0 TO 1280 STEP 1
28
100MOVE 0,1024
110DRAW S,0
120NEXT S
130FOR G=1280 TO 0 STEP -
128
140MOVE 1279,1023
150DRAW G,0
160NEXT G
170FOR A=0 TO 1280 STEP 1
28
180MOVE 0,0
190DRAW A,1024
200NEXT A
210FOR H=0 TO 1280 STEP 1
28
220MOVE 1279,0
230DRAW H,1279
240NEXT H
250MOVE 0,0
260BCOLO,2
270Z=0
280X=1280
290C=1024
300G=0
310FOR Q=1 TO 66
320DRAW X,G
330DRAW X,C
340DRAW G,C
350X=X-20
360G=G+20
370C=C-20
380NEXT Q
390MOVE 1280,0
400W=0
410R=0
420P=1280
430Y=1024
440FOR V=1 TO 66
450DRAW W,R
460DRAW W,Y
470DRAW P,Y
480P=P-20
490Y=Y-20
500R=R+20
510W=W+20
520NEXT V
530FORC=1 TO 200:NEXT C
540GOTO 40

```

## FOR CONNOISSEURS OF MODERN LANGUAGE LEARNING

COMMODORE 64 ● BBC (32K) ● ELECTRON ● SPECTRUM (48K)



Dealers, contact  
Lightning, Proteus,  
Centresoft, Microdeal  
or Tiger

All files are  
immediately  
available from good  
computer stores or  
by 24-hour mail  
order. Price £8.95  
(overseas  
orders add £1)

For beginners, O-level and beyond, these best-selling programs are unique and highly successful aids to language learning. Each cassette provides a comprehensive series of vocabulary lessons and a variety of self-paced learning and test modes. All accents and special characters are clearly displayed and different colours denote masculine, feminine and neuter words to reinforce gender learning. The create command enables new lessons in vocabulary or grammar to be entered, edited as required, then saved on tape. By using this simple yet vital feature, homework lists and exam revision can be retained indefinitely and recalled on demand. Two cassettes are available for each language, covering thousands of words; Level A provides 16 lessons in general vocabulary; Level B provides a further 16 lessons including adjectives, adverbs and fully conjugated verb lists.



KOSMOS SOFTWARE LTD 1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX Tel: (05255) 3942

The French Mistress Level A @ £8.95 <input type="checkbox"/>	The French Mistress Level B @ £8.95 <input type="checkbox"/>	Computer type.....(Commodore 64/88C/Electron/Spectrum)
The German Master Level A @ £8.95 <input type="checkbox"/>	The German Master Level B @ £8.95 <input type="checkbox"/>	Mr/Mrs/Miss .....
The Spanish Tutor Level A @ £8.95 <input type="checkbox"/>	The Spanish Tutor Level B @ £8.95 <input type="checkbox"/>	Address .....

KOSMOS SOFTWARE LTD.  
1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX.

..... Postcode .....

# Software Surgery

THE COLUMN THAT TAKES A LOOK INSIDE THE LATEST RELEASES

**The Five Stones of Anadon**  
Softek

THE last Softek adventure I reviewed – Eye of Zoltan – was very good. So it was with some interest that I loaded in The Five Stones of Anadon.

I think that if anything this is a better adventure than the last one.

Your local wizard is dying and it is up to you to recover the ring of five stones that are scattered about the kingdom.

You start your quest in the wizard's house surrounded by a plethora of objects – a dust-pan, a broom, a fountain pen, keys, a crowbar and gloves to name but a few.

You discover the wizard, though why anyone should lock him in his own bedroom is a mystery to me. You soon come across a cemetery with grave advice and a cellar with a rather cross ghost.

Further explorations lead you to a dragon and a black

## Magical moments in Anadon

knight. The solution to the problem posed by the dragon requires knowledge of an old adage about making cakes. The result is invisibility.

An inconsistency here however is that while you are invisible you can get past the dragon, but the knight can still somehow see you.

One other thing that came as a surprise is that as the wizard weakens, the stones become invisible. Luckily though, you can still collect them, if you know where they are.

You are also limited to a set number of moves after the stones have become invisible

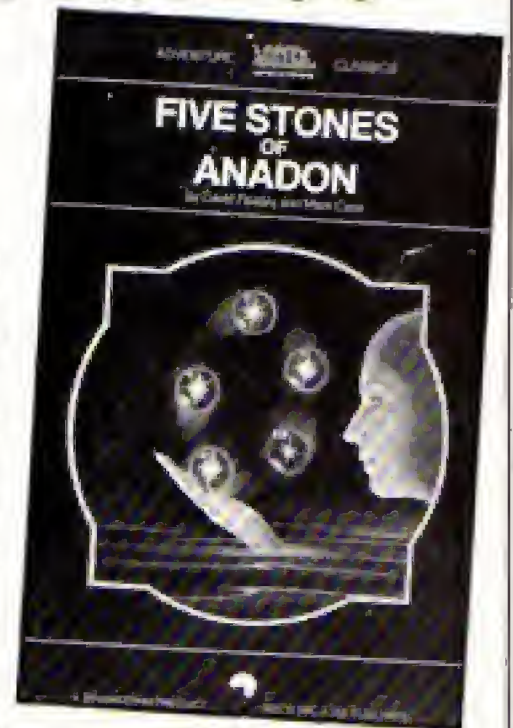
because eventually the wizard dies and then the game is over.

One slight niggle I have is that you have to QUIT in order to LOAD a saved position. It is all too easy to load in the wrong data file. It isn't difficult to implement this and I am surprised Softek have not done so.

Aside from that we have an adventure that is a joy to play. Most of the frequently used word parts are tokenised and thus, although written in Basic, the adventure is fast.

Overall, a very good attractively packaged adventure that is highly recommended.

Merlin



## Duo on data

**Data Structures Demonstrator**  
DP Publications

## LEARNING FROM DRAKE

THIS is a graphic adventure game simulating the voyages of Sir Francis Drake in the Pacific Ocean.

Having taken the cassette from its attractive library case, the first thing I noticed was that the program was almost impossible to load.

LCL have put the program on both sides of the cassette, but both proved difficult and required adjustments to tone and volume of my tape recorder for almost every block.

I took the only way out, struggled to load the program once and re-saved it onto my own cassette. It took over an hour to achieve.

The loading program presents a title sheet and sound that's meant to be the sea, and then draws a map of the Pacific Ocean. The second program loads and then takes about 30 seconds to initialise.

**Sir Francis Drake**  
LCL Software

Your boat starts at Lima and you must follow Drake's route via New Albion – California – across the Pacific to Java. On the way you commit acts of piracy so that you may bring riches back to your queen.

The boat is steered using the cursor keys and you have a permanent status record of cargo, supplies, crew number, cannon balls and damage.

As you sail you will encounter hazards such as rocks and reefs and may need to put into an unknown port for repair. The sea bed awaits anyone whose damage reaches 10.

If you see a Spanish ship you may attack or ignore. If the ship has a name, attack it. You always win. If it does not,

winning or losing is random.

Incidentally, a ship you beat has more cargo to steal. If you move away and come back, you can quickly gain your required cargo.

Winning is quite difficult and needs careful mapping and it is in this that the program has its value. A keen child would need to keep a chart – sample included with the program – and would thus practise record keeping, coordinates and map work.

A big snag is that if the player loses, the whole program has to be re-loaded.

It is not very well written in many ways and rather easy to cheat the system. Documentation is poor, keywords that are needed are not given but nonetheless at its lowish price – about £6 – it could be considered for primary schools.

Rog Frost

THIS cassette and book are totally interactive, and neither one would make any kind of sense without the other.

Well, perhaps the very keen student could work through the book alone but the 55k of programs on the tape are a considerable help.

I use the word student advisedly because the pack is really aimed at around 'A' level Computer Science exams. I confess I have never passed an exam in computing, having learned by doing and teaching myself but there is obviously a growing interest in formal qualifications in computing.

The back cover claims the book and cassette will also be of interest to anyone wishing to write data handling programs. I have my doubts whether the dry approach of this volume will encourage anyone not committed to this

## From Page 37

area of study by examination.

The programs set out to demonstrate on screen what is happening inside the computer during sorting and related activities.

This is done by the user making inputs which are manipulated into their correct places in the data structures while the appropriate Basic lines are highlighted.

In this way the use of loops is well demonstrated while conditions are met as usual and explained. A few terms new to my vocabulary appear in the book as, for instance, I had never used a hash table before.

I suppose it is a measure of the style of this book that I can now work reasonably well through examples using them.

The book is good value for those students meeting data structures in their syllabuses though I cannot imagine it becoming general reading matter among the average micro owners.

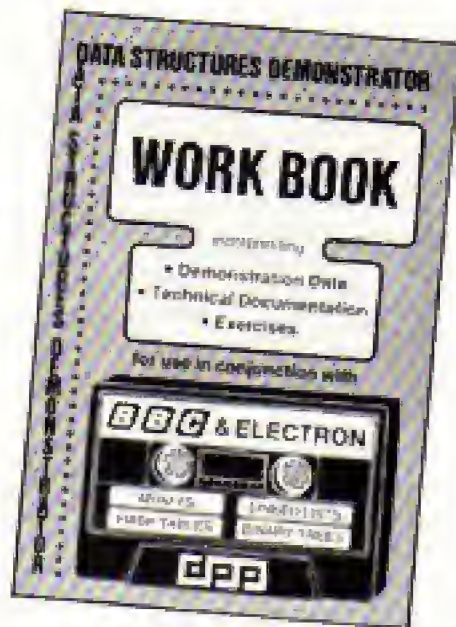
Most would be better advised to use a commercial program.

Phil Tayler

## Ticklish problem

**Here and There with the Mr. Men**  
Mirrorsoft by Primer Educational Software

Mr Tickle has a problem. He is in little bits all over the screen. Well, that's not too bad. We



can soon put him together.

Mr Grumpy has problems too. The regenerated Mr Tickle keeps tickling him.

Mr Lazy's problems are worst. A long red worm keeps eating through apples which then fall on his head.

In the end they all get their own back on Mr Tickle. The gang of four go after him!

That is the storyline behind this set of four programs aimed at 4 to 8-year-olds. It may sound trivial but it certainly is not.

The plot stimulates and involves the children in decision making and planning. The educational objectives are well defined and are met by the activities that the programs demand.

The theme of the package is left, right, up and down. In the first program the user has to move a gate either left and right or up and down to line up with parts of Mr Tickle.

The only keys used are the cursor controls and Return.

Graphics are good and produce comprehensive non-

verbal cues for the user.

The second program allows the user to control Mr Tickle's long tickling arms and attempt to tickle Mr Grumpy.

This time the child has to plan the actions and enter a short list of instructions such as "urd" (up-right-down) to guide Mr Tickle's arms to Mr Grumpy's nether regions.

Again the cursor keys can be used or the letters U D L R.

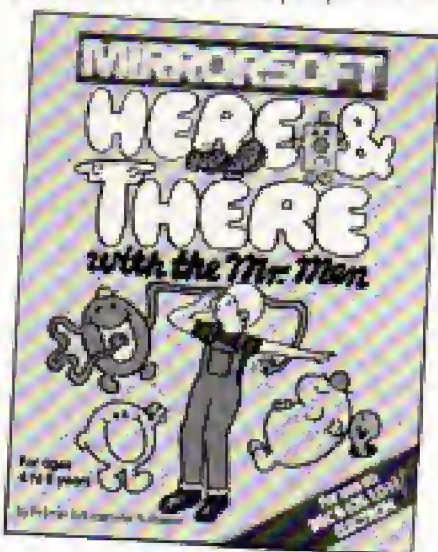
If your tickling ability gets too good then Mr Grumpy tries to place chairs in your way.

Mr Lazy appears in the third program. You control a worm that has to climb a tree and eat a particular apple. If successful the apple falls onto the head of an unsuspecting Mr Lazy.

Again, a set of instructions are entered and then carried out. I feel that this game is easier than the second, but that is a minor criticism as a parent or teacher can decide which order a child follows the programs.

I found this extremely difficult on a black and white monitor but fine in colour.

The final game, similar to Fox and Hounds, is played on a



chessboard.

The user controls four different Mr Men and attempts to trap Mr Tickle. Mr Tickle does not play the game too well and so it should be possible for most children to succeed in trapping him.

There is a comprehensive manual for an adult to read to the child. The sound cues can be switched off.

I really enjoyed going through this package and so did the children I tried it on!

This is the type of educational software I would like to see entering the home market.

John Woollard

## Jump to it...

**Eddie Kidd Jump Challenge**  
Martech

THIS program takes the story of Eddie Kidd and puts it neatly into a computer game. It is a 'jump challenge' for you because each copy of the program enables you to enter your highest total score into a national competition.

The game begins with you, the challenging stunt person, proving your ability on a BMX. Before you are let loose trying to jump cars you have first to try oil barrels.

To graduate from the BMX you have to make two successful jumps over the barrels. The first is easier than the second.

If at any time you crash you

# SIMPLICITY MAKES A WINNER

**Frenzy**  
Micropower

HERE Micropower has chosen a format which is simplicity itself, made it the simplest of games to use, and yet come up with what I think is one of the most amusing and compulsive games on the market today.

Combine this with a highly colourful display, high scoring and wide age range appeal and you have a winner.

What has happened is that a deadly Lepton has broken

free inside a laboratory and is bouncing around inside.

Luckily it cannot penetrate the walls or pass through the ion trail that you are about to lay inside.

To enter is certain death, so you employ a robot vehicle to do the work of laying the trail, and you sectionalise the laboratory, thus trapping the Lepton inside a small area.

If you cut off a small part of the laboratory but fail to trap the Lepton, you carry on until 95 per cent of the area has been covered, when the

Lepton is eventually caught. This, however, reduces your bonus, which decreases as time passes.

If you do trap the Lepton it is a points bonanza and a big bonus to boot. But if the Lepton hits the robot vehicle or the ion trail before contact is made with another wall or another part of the trail, then it's one life lost.

As the game progresses little refinements are added, like chasers which follow your tracks.

Needless to say, they are on

the Lepton's side and contact with one costs you a life.

At one stage the Lepton multiplies itself and all are equally deadly. Then again it can move at double speed and you need to be very clever to outsmart it.

Your robot vehicle has two speeds. The higher means the less time spent in the danger area but fewer points.

A cool head, a steady hand and nerves of steel should ensure you a top score in this excellent game.

Adam Young



are sent back to the BMX to start again.

Assuming a little competence you'll get to the motor bike level. The screen display now includes a speedo, separate rev counter and a gearbox indicator.

It is vital that you use the gearbox and throttle together to gain speed (gears are changed by pressing the corresponding number key).

Using the keyboard alone I found no difficulty, but with a joystick I doubt that I would manage to keep hitting the right key.

As you succeed with each jump the length increases and so does the difficulty.

The Eddie Kidd Jump Challenge Competition is open to each cassette owner. However you may only make one entry.

If you make a jump that qualifies the game stops and you are given two options.

One is to continue and take the risk that you may crash on the next round. The other is to enter the competition by inserting a blank cassette and following the screen instructions.

This is a neat games package with the bonus of a free-to-enter competition. It should be very popular.

**John Woollard**

## Be a villain

**Smash and Grab**  
Superior Software

WITH a swift kick you send the police traffic cone crashing through the bank window. Bags of money begin to fall from the broken window and drift gently down towards the river below...

In Smash and Grab your job

as the villain is to catch the falling loot before it reaches the river.

As is always the case in these ladders and levels games there is a snag. In this one it is in the shape of PC Plod, your local neighbourhood bobby.

This particular policeman would be more at home in the Sweeney than pounding the beat.

Should you be running along the level directly above him he is quite likely to leap upwards and thrust his truncheon through the floor into your nether regions.

When he is on the upper level his actions are even more dramatic as he falls flat on his face and batters you around the head.

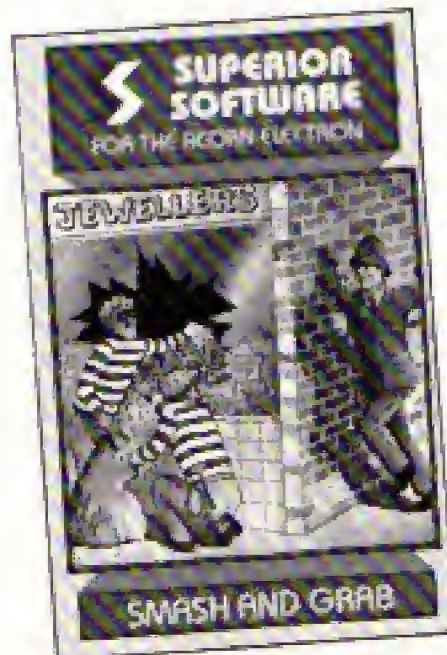
To complicate things further the occasional trio of flying traffic cones will wing their way along one of the levels in your direction. These can be dealt with by either getting out of the way, or by kicking each one in turn by pressing Return.

Although it may sound that the odds aren't exactly in your favour you do have one trick up your sleeve. Should you kick one of the four police boxes when the light on top is flashing the traffic light at the top of the screen changes to red.

Immediately the bags of money stop falling, and any physical contact with the policeman will send him plunging to the water below.

If you can collect eight bags of money you progress to the next screen. However as each missed bag of loot hits the water an alarm bell appears at the top of the screen. Five alarm bells and a life is lost.

**Jon Revis**



## Have gun, have fun

**Gunsmoke**  
Software Invasion

HAVE you ever had one of those days where you feel like picking up a gun and shooting a few dozen people?

Well, with Gunsmoke you can shoot as many as you like. But you have to be fast on the draw and quick on the trigger.

The game starts by setting the scene, a well drawn view of one side of a Western town with a store, saloon and sheriff's office.

You play the lawman, controlling an animated figure who walks up and down the street.

Suddenly you're under attack from up to 16 baddies who appear in the windows of the buildings and shoot at you.

You have to run to avoid the bullets and then fire back.

Even if you get hit you have three lives and, as a special offer, for every 16 gunmen you



kill you get an extra life.

It's a simple game that will appeal mostly to children. The animated man could be a little better but otherwise the graphics are excellent.

Not too difficult, it's the kind of game that has spectators looking over your shoulder yelling: "Saloon, top window. Quick!".

**Nigel Peters**

## Battles in the past

**Roman Empire**  
Lothlorien

HERE is a chance to turn the clock back 2,000 years and try your luck at conquering the world.

The nine armies of Rome, each commanded by a great general, consist of 18 legions, each with 5,000 legionnaires.

You must decide how they will be divided among the armies and which country each army will attack.

You can display at any time the state of each army, its manpower, fighting ability, morale and strength, and also the deployment of your legions and their current manpower.

Also available at any stage is a map of the world, showing the countries which you have conquered to date.

Another display gives the strength of your opposition.

Using all the information available, you decide who to attack and the optimum size of your army.

An attack is then launched and continued until either the opposition is subdued or your army is wiped out (if you are anything like me, it will be the latter).

The foregoing sounds simple, but tactics are every-

thing. Also the world is a big place, and the object of the game is to conquer the world, nothing less.

While you're doing this you also have to defend home against the barbarians who are just waiting for the defence to become weakened by excessive conquering and pillaging.

I had great fun switching my armies around, sending battle-weary troops home and replacing them with fresh men and launching raids to harass and weaken the defending armies.

I also found a way to conquer the world and lose hardly a man in the process, without cheating, but I'm keeping that to myself.

The game involves a lot of text and few graphics. I found it extremely intriguing, and it passes away the hours on a rainy afternoon.

A game, I think, for the more mature player, who likes to use his head rather than shoot from the hip.

**Adam Young**

Now No. 1 BBC  
game on Micro Deal  
Top Fifty chart

# YOU can go for gold ...with the



Fancy pitting yourself against the world's best at this summer's Olympics?

You can do so without going anywhere near Los Angeles – with the most challenging package of programs of 1984.

MICRO OLYMPICS is more than a game. It's a brilliantly written collection of ELEVEN track and field events.

And because we know we're going to sell many thousands of them we've brought the price right down – to just £5.95.

Ever imagined yourself as another Seb Coe? Then try to run against the world record holder at 1500 metres. And if that distance is too much for you then there's always the 100, 200, 400 and 800 metres to have a go at.

Not much good at running? Don't worry, MICRO OLYMPICS has many more challenges for you. Why not try your skill at the high jump or the long jump?

And if you can't beat the computer at running or jumping then you can always throw things around in frustration! The trouble is that it's just as hard to be a champion at the discus, the hammer or the javelin.

And the pole vault takes the event to new heights!

Yes, it's fast, furious fun, pitting yourself against the world's best times and distances on your micro.

You may not be another Steve Ovett or Alan Wells, but with practice you **COULD** become the Micro Olympics Champion!



Also available from WH Smith  
and all other leading stores



**Play Micro Olympics  
– and let your fingers  
do the running!**

**Send for it today**

Please send me \_\_\_\_\_ copy/copies of  
Micro Olympics

I enclose cheque made payable to  
Database Publications Ltd.  
for £ \_\_\_\_\_

<input type="checkbox"/> BBC '8" cassette	£5.95
<input type="checkbox"/> Electron cassette	£5.95
<input type="checkbox"/> BBC 40-track disc	£7.95
<input type="checkbox"/> BBC 80-track disc	£7.95

*Please tick box*

I wish to pay by

Access  Visa No. \_\_\_\_\_ Expiry date \_\_\_\_\_

Signed \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

Post to: Micro Olympics offer, Database Publications,  
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

EU/3



**Find the missing word in this fun-filled educational game by STEVE LUCAS**

IN this educational game aimed at 5 to 8-year-olds a series of simple sentences is displayed on the screen, each with a word missing.

To help the players fill the gap four words are printed in boxes on the screen and the child moves Fred until he is next to the word selected.

Return is pressed and each correct answer increases the score by one.

Keys to move Fred are Z for left and X for right.

After each question the child is asked to press the space bar for the next question or E to end the game.

**PROCEDURES**

**PROCmove**  
**PROCwin**  
**PROClose**  
**PROCinstructions**  
**PROCyn**  
**PROCdraw**

Moves Fred around the screen.  
 Correct answer.  
 Wrong answer.  
 Prints instructions.  
 Next question or end game.  
 Draws Fred.

**VARIABLES**

**S%** Score.  
**Z%** Colour for Fred.  
**X%** x coordinate of Fred.  
**A\$(X)** Words.  
**B\$** Sentence.  
**N%,B%** Flags.



## Fred's Word Game listing

### From Page 41

```

10 REM ** Fred's Word Ga
me **
15 REM (C) Electron User
20 VDU 23,1,0,0,0,0:
30 SX=0
40 *FX210,0
50 ON ERROR GOTO 1670
60 CLS:PRINT"Do you want
sound (Y)es or (N)o ?"
70 REPEAT:A$=GET$
80 UNTIL A$="Y" OR A$="N
.
90 IF A$="N" THEN *FX210
,1
100 ENVELOPE 1,1,43,0,0,1
00,0,0,126,0,0,-126,126,126
110 ENVELOPE 2,1,5,0,0,25
,0,0,126,0,0,-126,126,126
120 ENVELOPE 3,12,10,0,0,
4,0,0,126,0,0,-126,126,126
130 VDU23,255,255,255,255
,255,255,255,255,255
140 VDU23,254,7,13,30,13,
5,1,15,17
150 VDU23,253,224,176,120
,176,160,128,240,72
160 VDU23,252,17,17,57,1,
7,7,0,0
170 VDU23,251,72,72,92,64
,112,112,0,0
180 VDU23,250,24,24,24,24
,24,24,24,24
190 VDU23,249,126,126,126
,126,126,60,24,0
200 VDU23,248,0,0,0,0,20,
62,127,127
210 VDU23,247,127,62,20,0
,0,0,0,0
220 VDU23,246,24,60,126,1
26,126,60,20,54
230 VDU23,245,34,65,65,65
,0,0,0,0
240 VDU23,244,31,17,17,25
5,255,255,20,20
250 VDU23,243,254,254,254
,254,255,254,56,56
260 VDU23,242,0,0,15,0,12
7,127,24,24
270 VDU23,241,0,0,252,50,
254,254,40,40
280 VDU23,240,0,1,7,15,0,
127,31,15
290 VDU23,239,0,192,240,2
40,120,255,252,240
300 VDU23,238,0,20,42,127
,54,20,0,20
310 VDU23,237,62,54,54,54

```

```

,127,99,99,65
320 MODE 2:VDU19,0,7,0,0,
0,19,7,0,0,0,0,5
330 A$=" F r e d ' s   W
o r d   G a m e":BX=0:SDU
ND1,3,9,40
340 FOR X=100 TO 1200 STE
P 30:GCOL0,1:MOVE X,1000:VD
U254,253,10,0,0,252,251:BX=
BX+1
350 B$=MID$(A$,BX,1):MOVE
X,900:PRINTB$:MOVE X,1000:
TIME=0:REPEAT UNTIL TIME>0
360 GCOL0,0:MOVE X,1000:VD
U254,253,10,0,0,252,251:NE
XT:GCOL0,4
370A$=" Steve Lucas 198
4":BX=0:FOR X=1 TO 1270 STE
P 60
380 GCOL0,4:MOVE X,700:VD
U254,253,10,0,0,252,251:BX=
BX+1:B$=MID$(A$,BX,1)
390 MOVE X,600:PRINTB$:MO
VE X,700:TIME=0:REPEAT UNTI
L TIME>2:GCOL0,0
400 VDU254,253,10,0,0,252
,251:NEXT:Y=1:FOR X=1 TO 10
:MOVE X*105,400:GCOL0,Y:VDU2
54,253,10,0,0,252,251:Y=Y+1
:IF Y>7 THEN Y=1
410 NEXT:GCOL0,7:MOVE0,20
0:PRINT" Press (Space Bar)
for instructions.":REPEA
T UNTIL GET=32
420 CLS:MODE6:VDU19,0,4,0
,0,0,23,1,0,0,0,0:SDUND1,1
,5,15:PROCInstructions
430 MODE 1
440 DIM A$(4):VDU19,0,6,0
,0,0,19,2,5,0,0,0,19,3,4,0,
0,0
450 XX=0:FOR Y=1 TO
XX:READ A$,N$,C$,D$,E$,AX:
NEXT
460 REPEAT
470 CLS
480 READ B$:FORIX=1TO4:RE
ADA$(IX):NEXT:READAX: IFA$(
1)="I" THEN RESTORE:GOTO400
490 COLOUR3:PRINT"TAB(14
0-LEN(B$))/2):B$: ".":COLOUR
2
500 GCOL0,2:MOVE40,400:MO
VE340,400:PLOT05,40,500:PLD
T05,340,500:GCOL0,1:MOVE340
,400:PLOT05,640,400:PLOT05,
640,500:PLOT05,340,500
510 MOVE640,400:GCOL0,2:M
OVE940,400:PLOT05,640,500:P

```

```

LOT05,940,500
520 MOVE940,400:GCOL0,1:M
OVE1240,400:PLOT05,940,500:
PLOT05,1240,500
530 VDU5
540 GCOL0,3:MOVE(320-LEN(
A$(1))+16)/4+50,450:PRINTA$(
1)
550 MOVE(320-LEN(A$(2))+1
6)/4+350,450:PRINTA$(2)
560 MOVE(320-LEN(A$(3))+1
6)/4+650,450:PRINTA$(3)
570 MOVE(320-LEN(A$(4))+1
6)/4+950,450:PRINTA$(4)
580 PROCmove
590 IF NX=AX THEN PROCwin
ELSE PROClose
600 UNTIL FALSE
610 END
620 DEFPROCInstructions
630 PRINT"In this game, y
ou will be shown a seriesof
sentences. Each sentence w
ill have a word missing and
you must try to find the
missing word."
640 PRINT:PRINT"To help y
ou, four words will be show
n inboxes on the screen. On
e of these words will fit i
nto the sentence."
650 PRINT""You should mo
ve FRED around the screen
until he is underneath the
box containing the
word you want to select a
nd then press (RETURN) to m
ake your choice."
660 PRINT""Use the follow
ing keys to move FRED a
round the screen :-"
670 PRINT"SPC10"Z = left
X = right"
680 PRINT"TAB(2)"Press (
Space Bar) to start the gam
e":*FX15,0
690 REPEAT UNTIL GET=32
700 ENDPROC
710 DATA The dog lives in
a ---,hutch,kennel,sty,nes
t,2
720 DATA We stayed at the
--- of the baths,hide,side
,tide,bide,2
730 DATA We went to the -
-- today,lark,park,dark,bar
k,2
740 DATA I put the rubbish
h in the ---,pin,din,bin,si

```

```

n,3
750 DATA The boy read a -
--,book,arc,chalk,paper,1
760 DATA The girl played
in the ---,mark,park,lark,b
ark,2
770 DATA The boy likes to
wear a ---,tap,cap,rap,sap
,2
780 DATA I like to ride i
n a ---,far,car,tar,bar,2
790 DATA We saw an elepha
nt at the ---,zoo,few,too,b
oo,1
800 DATA I wrote a letter
with my ---,ten,pen,den,he
n,2
810 DATA The fire was ver
y ---,tot,dot,hot,not,3
820 DATA My mother was ve
ry ---,dad,lad,sad,fad,3
830 DATA I like --- on my
toast,jax,pax,sax,dax,1
840 DATA I put the --- in
the socket,mug,plug,rug,tu
g,2
850 DATA The --- layed an
egg,pen,ten,den,hen,4
860 DATA I ate a --- of c
hocolate,far,bar,tar,car,2
870 DATA I --- my sister
to go home,sold,told,gold,m
old,2
880 DATA I sat on the ---
,tug,bug,dug,rug,4
890 DATA I had to --- the
bell,sing,ping,ring,zing,3
900 DATA The sea was very
,hold,told,sold,cold,4
910 DATA There was a high
---,tide,side,bide,hide,1
920 DATA I like to lie in
--- in the morning,red,ted
,bed,fed,3
930 DATA I like to --- my
dinner,meat,feat,eat,seat,
3
940 DATA The butcher sell
s ---,seat,neat,meat,heat,3
950 DATA I had to open th
e ---,gate,late,rate,fate,1
960 DATA The lady pushed
the baby in a ---,san,pran,
ram,pam,2
970 DATA We played in the
---,land,sand,hand,and,2
980 DATA The weather was
---,funny,money,runny,sunny
,4

```

```

990 DATA The barber --- a
y hair,hut,nut,cut,rut,3
1000 DATA We went for a --
-,side,ride,hide,tide,2
1010 DATA I had to --- a m
odel car,make,fake,bake,cak
e,1
1020 DATA The car was very
---,mast,fast,last,past,2
1030 DATA Dad likes to ---
in his chair,fit,pit,nit,s
it,4
1040 DATA It is not --- to
go to school,far,tar,mar,b
ar,1
1050 DATA The plane was a
---,met,pet,set,jet,4
1060 DATA I forgot to ---
the gate,shut,put,hut,nut,1
1070 DATA We played cricke
t with a --- and ball,fat,m
al,bat,rat,3
1080 DATA The colour of th
e ball was ---,bed,red,fed,
ted,2
1090 DATA There were five
peas in the ---,pod,rod,cod
,god,1
1100 DATA I put some money
in the ---,plank,rank,tank
,bank,4
1110 DATA I went for a rid
e on my ---,bike,bike,hike,
like,2
1120 DATA I bought a cake
in a ---,hop,pop,shop,top,3
1130 DATA The fish --- in
a pond,swims,fins,bins,tins
,1
1140 DATA Pam's pet is a -
--,mat,cat,fat,cat,4
1150 DATA Mary put the foo
d in a ---,wish,wash,dish,f
ish,3
1160 DATA Paul was --- for
school,hate,plate,late,cak
e,3
1170 DATA Jill took her --
- to the shops,big,beg,bug,
bag,4
1180 DATA Tom will --- in
the bus,sit,set,sun,sue,1
1190 DATA Mum poured the m
ilk from a ---,tug,lug,bug,
jug,4
1200 DATA The boy will go
to --- tonight,bed,bud,bug,
bun,1
1210 DATA Paul wants a ---

```

```

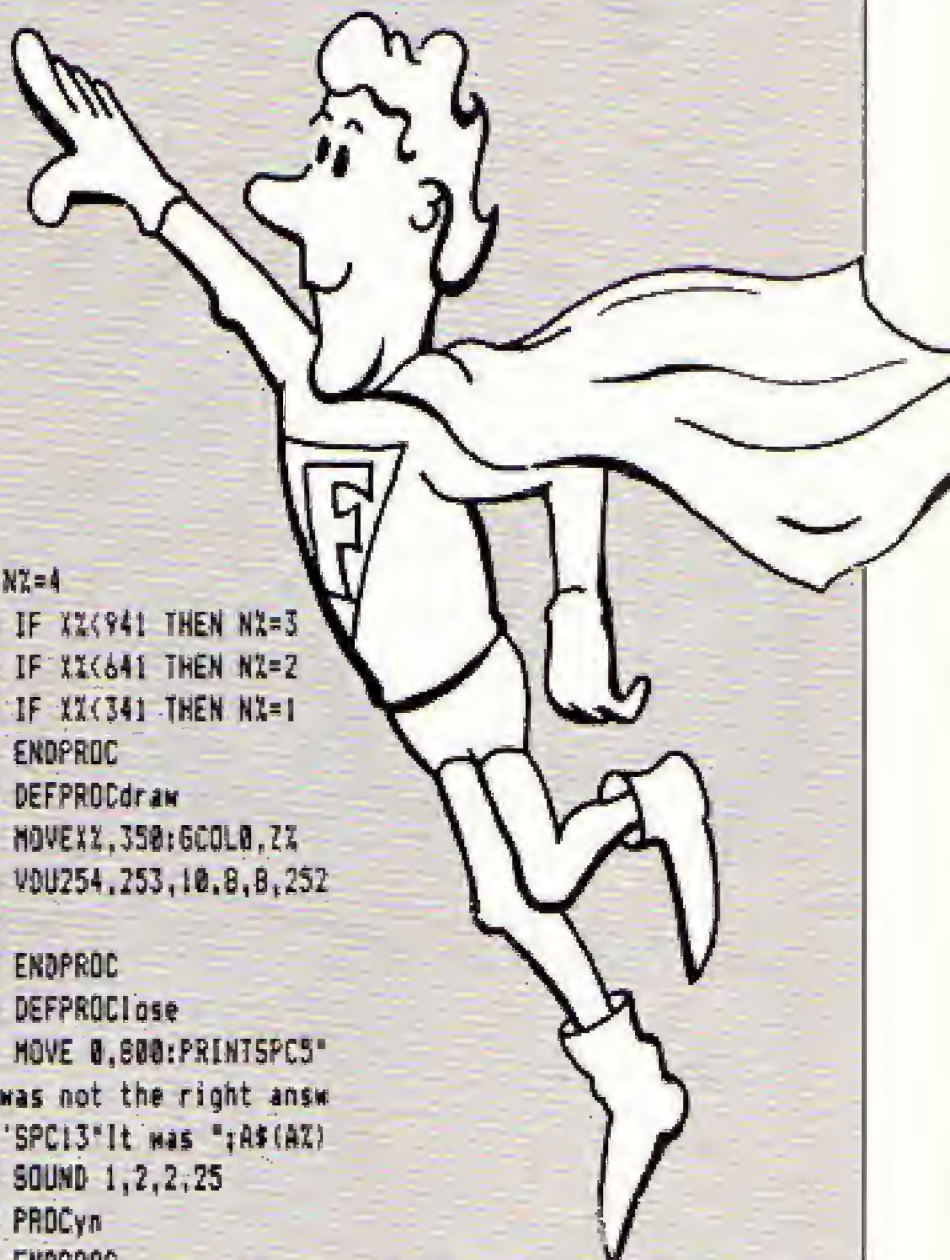
for a pet,log,dog,fog,jog,
2
1220 DATA The boy played w
ith a ---,kite,kite,site,bi
te,2
1230 DATA We put the plant
in a plant ---,lot,tot,dot
,pot,4
1240 DATA John played cric
ket with a ---,bull,bell,ba
ll,bill,3
1250 DATA The drink of tea
was very ---,hut,hit,hut,h
ot,4
1260 DATA The number after
nine is ---,tin,ten,tan,on
e,2
1270 DATA The boy wrote wi
th a ---,pen,pin,pan,pun,1
1280 DATA That man is very
---,tall,tell,till,toll,1
1290 DATA The baby lay in
a ---,cot,not,rot,dot,1
1300 DATA The --- was in a
field,hull,mull,full,bull,
4
1310 DATA Mum opened a ---
of beans,pin,bin,din,tin,4
1320 DATA The boy sat on a
---,hair,lair,chair,fair,3
1330 DATA We went for a sa
il on the ---,lake,make,cak
e,rake,1
1340 DATA Lynn climbed the
---,fill,mill,hill,till,3
1350 DATA The dog chases a
---,cut,cat,cot,sit,2
1360 DATA The dog likes to
---,mark,bark,park,lark,2
1370 DATA I missed the ---
bus and had to walk,last,m
ast,cast,vast,1
1380 DATA She sees Tom in
the ---,toad,load,road,sad,
3
1390 DATA I,I,I,I,I,1
1400 DEFPROCmove
1410 X1=0:Z1=1
1420 REPEAT
1430 IF INKEY(-98) THEN ZX
=0:PROCdraw:ZX=ZX-40:IFZX
<30 THEN ZX=0
1440 IF INKEY(-67) THEN ZX
=0:PROCdraw:ZX=ZX+40:IFZX
>1160 THEN ZX=1160
1450 ZX=1:PROCdraw
1460 UNTIL INKEY(-74):+FXI
S,0
1470 NX=0:IF ZX<1200

```

```

THEN NX=4
1480 IF ZX<941 THEN NX=3
1490 IF ZX<641 THEN NX=2
1500 IF ZX<341 THEN NX=1
1510 ENDPROC
1520 DEFPROCdraw
1530 MOVEZX,350:GCOLOR,ZX
1540 VDU254,253,10,0,0,252
,251
1550 ENDPROC
1560 DEFPROCclose
1570 MOVE 0,800:PRINTSPC5"
That was not the right answ
er""SPC13"it was ";A$(AZ)
1580 SOUND 1,2,2,25
1590 PROCyn
1600 ENDPROC
1610 DEFPROCwin
1620 SX=SX+1
1630 MOVE 0,800:PRINTTAB(2
)"Well Done. That's the ri
ght answer."
1640 SOUND1,3,4,50
1650 PROCyn
1660 ENDPROC
1670 MODE 8:PRINTTAB(5,15)
"Error ";ERR;" in line numb
er ";ERL:END
1680 DEFPROCyn
1690 MOVE 100,196:GCOLOR,2:
PRINT"Press:--"
1700 MOVE100,100:GCOLOR,3:P
RINTSPC(3)"or (Space Bar)
to continue."
1710 MOVE 100,140:GCOLOR,1
:PRINTSPC(8)"<E>
to end game."
1720 REPEAT
1730 FF=GET
1740 UNTIL FF=32 OR FF=69
OR FF=101
1750 IF FF=32 THEN ENDPROC
1760 CLS
1770 A$=" Goodbye. Thank
you for playing":BX=0
1780 FOR X=100 TO 1200 STE
P 30

```



```

1790 GCOLOR,1:MOVE X,1000:V
DU254,253,10,0,0,252,251:BX
=BX+1
1800 BX=MID$(A$,BX,1):MOVE
X,900:PRINTBX:MOVE X,1000:
TIME=0:REPEAT UNTIL TIME>0
1810 GCOLOR,0:MOVE X,1000:
VDU254,253,10,0,0,252,251:N
EXT:GCOLOR,4
1820 A$=" You score
d :- "+STR$(SX)
1830 BX=0:FOR X=100 TO 120
0 STEP 30
1840 GCOLOR,3:MOVE X,500:VD
U254,253,10,0,0,252,251:BX=
BX+1
1850 BX=MID$(A$,BX,1):MOVE
X,400:PRINTBX:MOVE X,1000:
TIME=0:REPEAT UNTIL TIME>0
1860 GCOLOR,0:MOVE X,500:V
DU254,253,10,0,0,252,251:NE
XT:GCOLOR,4
1870 VDU4:PRINTTAB(0,30);"
"
1880 END

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*

If you are getting bored with the same old type faces, think . . .

# BIG LETTERS



"The Abbot's playing space invaders again, so its back to the drawing board".

By MIKE BIBBY

THE inspiration for Font came when I was writing educational software for my children. I wanted to use Mode 5, but the character set was far too squat.

I decided that if I made them double height, the proportions would be more pleasing and it wouldn't be too difficult to do.

However I've a strong dislike of programs that, for the sake of a little more effort, could be made more general. So I wrote Font as a utility that allowed a wide variety of type faces to be used in the graphics modes.

The idea is quite simple really. In graphic modes the characters are composed of a set of pixels or dots, laid out in an eight by eight grid.

The micro lights different patterns of these dots for the various letters. Figure I

illustrates the pattern for the letter A.

It remembers these patterns in an area of ROM from &C000 onwards, storing each character in a group of eight bytes. The first byte corresponds to the first row of the character, the second byte to the second, and so on.

As the byte for each row has eight bits, each bit decides what goes on in a particular cell of that row. If there's a 1 in that bit the micro turns that pixel on - if there's a 0 it leaves it off.

Figure II shows how the numbers work out for the letter A.

By now you've probably noticed the resemblance to the way we create user

defined characters.

We can create another A (though heaven knows why we'd want to) with VDU 23, 224, 60, 102, 102, 126, 102, 102, 102, 0 where:

- 23 tells the micro what we are doing.
- 224 tells the micro the ASC number we're going to assign to the character we're defining.
- The next eight bytes teach it the pattern.

Font uses the patterns of the characters as a guide to placing rectangular building blocks on the screen.

The actual shape of the block is stored as a user defined character which I print on the screen wherever a 1 in the pattern dictates it - if it's a 0 I just move over, or down to the next row if it was the last bit.

When you think about it, you can vary exactly how much you move over or down between blocks, so you can "spread" the characters out over the screen. You can also vary the size of the blocks - allowing you to create a large

range of type faces.

The actual printing is done with VDU 5 on, so that the user defined character can overlap a previous one without blanking it out.

Font in fact consists of two programs. The first is an assembler language program (Listing I) which will create a piece of machine code.

When you run the program it will ask you for the location you wish the code to be stored in. I suggest &B00, where the function keys are normally stored. (This explains the funny letters you'll get on Break.)

Whatever it is, make a careful note of it. Incidentally, it expects hexadecimal input, so you don't need to enter the &.

A second program (Listing II) allows you to design your typeface and also contains procedures to handle the machine code you can incorporate in your own programs.

When you run Listing I it will prompt you to save the code on tape. Make sure that you do, as it's that you'll be using from now on, not Listing I. It's saved as FCODE.

Important point now. If you want to make use of the procedures contained in Listing II in your own programs

0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0

Figure I: How A is stored in ROM in binary

								Decimal	Hex
								60	3C
								102	66
								102	66
								126	7E
								102	66
								102	66
								102	66
								0	0

Figure II: The row numbers for A as a user defined character

you must have FCODE in your machine. To do this you must load it with:

#### \*LOAD FCODE

Notice that it's not an ordinary LOAD, because this is machine code, not a Basic program.

Notice that in Listing II line 60 is:

```
IX=4000
```

*I%* tells the micro where FCODE is stored. If you've stored it somewhere else you'll have to alter this.

As mentioned, Listing II not only shows you how to incorporate FCODE in your own programs, but also lets you create the type face of your choice.

When you run it you will be asked for the mode you require.

Try entering 2, and respond N to the special character option. You'll see a screen similar to Figure III.

The word Test is printed under conditions determined by *x*, *y*, *w* and *d*, giving you an idea of the characters you'll get with each set of options.

The *x* and *y* refer to the spacing, in pixels, between the rectangular blocks you are using to replace the pixels in the characters you want in the new type face.

Initially they are one pixel apart, as normal.

*w* and *d* refer to the width and depth of the blocks respectively – again in pixels.

The 2 simply reminds you that you are in Mode 2.

I stands for increasing. If you press Return the screen is redisplayed with D in its place – this stands for decreasing. Press Return once more and I is back.

The significance is that if you press the X key while I is displayed, the value of *x* increases and TEST is displayed accordingly.

If you press X while D is displayed the value decreases.

Return "toggles" between the two values.

The same holds for *y*, *w* and *d*. Escape will take you back to the beginning of the program.

F will let you finish, after giving



Figure III



Figure IV



Figure V

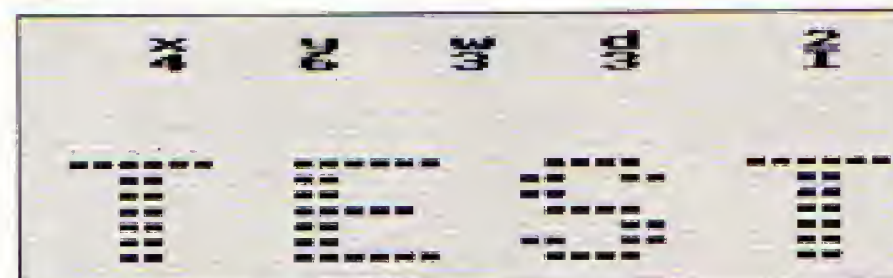


Figure VI

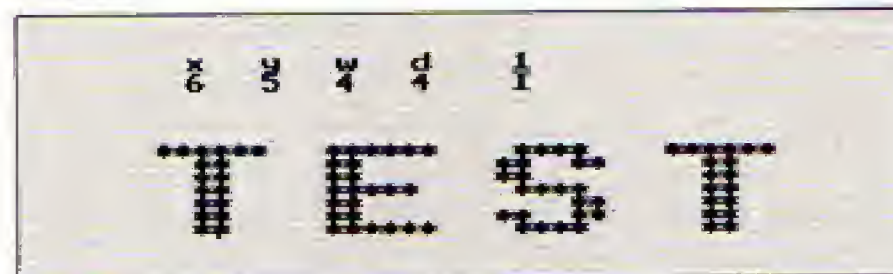


Figure VII

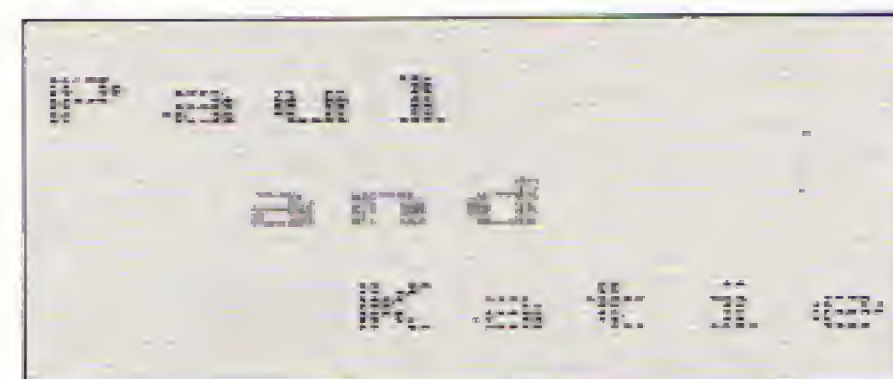


Figure VIII

you another chance.

It's easy to get used to and great fun to play with. Figures IV, V and VI show some possible combinations.

Note that you cannot increase *w* and *d* indefinitely. Each has a maximum size of eight, being limited by the size of the user defined character we're using to define the rectangle.

*x*, *y*, *w*, *d* are important – so once you've picked the way

you want your letters to look, note them down.

To use this character set in your own programs you then need to:

- At the program's start, set *I%* to the address at which FCODE will load.
- Include PROCw and PROCf in your program – it doesn't matter about line numbers.
- Have \*LOADed FCODE. You then call PROCf with

the figures for *x%*, *y%*, *w%*, *d%* you've noted as parameters. This sets up the font.

To print words in the type face thus chosen you simply pass them to PROCw. The first two parameters are the X and Y graphic coordinates of the start of the word on the screen and the third parameter is the string you pass to it.

So assuming you've actually got FCODE in,

```
PROCw(640,512,'A')
```

will print the letter A at the centre of the screen in whatever font you've chosen at the time (with PROCf).

Notice that:

- You can change fonts at any time.
- The font won't affect your normal printing.

The special character option allows you to actually print a large character made up of normal sized characters from the standard character set.

You could, for instance, make up a letter A entirely of As – or a user defined character of your choice.

If you want to do this, load FCODE and poke (*I%+&B3*) with the Ascii code of your character – also make sure that you've spaced out the *x%* and *y%* sufficiently in PROCf.

In Listing II I chose an initial width of four pixels between *x* and *y* to space the characters out – *w* and *d* are irrelevant. Having called PROCf, continue with PROCw as before.

Figure VII shows the sort of results you can achieve with this technique – I've used an asterisk as the special character.

Listing III demonstrates just a small amount of the potential of this technique.

Alter *location%* in line 30 if you've located FCODE elsewhere – I've used it rather than *I%*.

Notice that we use GCOL to change colour, not COLOUR. This is because we are writing under graphics control.

Figure VIII shows the output – my children's names! It's only fair, they started it all.

# HOW FONT WORKS

LISTING 1 consists of two main routines. One defines a rectangular character block, and the other prints out a single Ascii character using the pixel pattern of that character as a map.

We shall concentrate on the routine *definer* (line 910). Although it's towards the end of the listing, it does precede the others logically.

This routine defines the shape of character &EO (224) by using OSWRCH to do a machine code equivalent of VDU 23.

*width* holds the pixel width of the character you want. *define\_row* then repeatedly sets carry and RORs an initially empty accumulator *width* times, using X as index. This puts as many ones into the accumulator as necessary to define a row.

We need this row *depth* times, so using Y as an index we use OSWRCH to send these new values. Having done that, since VDU 23 needs eight values it might appear that we should then send (*8-depth*) zeros via OSWRCH.

To avoid doing the "sum"

we use *send\_blanks* to send eight consecutive zeros. The ones necessary to finish the VDU23 are "grabbed" by the OS, the rest ignored. This may be inelegant, but it's effective!

The rest of the code is straightforward. Lines 100-180 set up bytes for the variables to be placed into, 190 making space for an eight byte parameter block or table.

*start* loads X,Y with the address of this parameter block and, using OSWORD with A=10, reads the eight byte character definition of the letter we want to place on the screen into the table we have set up.

It will need to have the Ascii code for the letter at *pattern*, but we'll assume that the calling routine supplies this. The rest of the routine consists of two nested loops. These are *outer\_loop* and *inner\_loop*.

*outer\_loop* selects each row in turn from the table (indexed by Y) and *inner\_loop* examines that row bit by bit using ASL.

If there's a 1 carry is set and *write* called. If not, *noprint* simply skips past this.

*xpos* and *ypos* hold the

graphic coordinates at which the character is to be printed. These are transferred to *xstore* and *ystore*.

*ystore* is decremented at the end of each *inner\_loop* by *ystep*, to ensure that the next line is printed with the correct vertical spacing.

*xstore* is incremented by *xstep* within *inner\_loop* after *writing* or *noprint* to ensure that the next printing is correctly spaced across the screen.

After *outer\_loop*, *xpos* is updated to ensure that the next enlarged character is correctly positioned horizontally. If you think about it, you'll realise that *ypos* isn't updated since the next character should start on a level with the previous one.

*write* uses OSWRCH character repeatedly to:

*y%* holds the vertical separation.

*w%* holds the width of the block.

*d%* holds the depth of the block all in pixels.

In line 550 *m%* contains the current mode determined by calling Osbyte &87. It then uses *f%* to convert the variables from pixel size to graphic units, storing them in *f%*, *f%+3* for FCODE to use.

Finally it calls *f%+&BD* (*definer*) to define the rectangle as 224.

PROCw prints a string out in the style defined by PROCf. Its parameters are the coordinates to start printing at (*x%* and *y%* again - not the same as in PROCf) and the string (*w\$*).

The routine then pokes each character of the string in turn into *f%+&C* then calls

**VDU 5** joining text and graphic cursors so characters can overlap (line 720).  
**VDU 24,4, MOVE** *xstore, xstore+1, ystore, ystore+1*, the cursor to the absolute position for printing (lines 740-850).  
**VDU 224** printing the character (line 860).  
**VDU 4** separating cursors (880).

Listing II uses PROCf to set up the type face.

*x%* holds the horizontal separation.

*f%+&15* to "print" it (lines 480-490).

PROCkey simply looks for keyboard inputs.

### Listing 1

```

10 REM LISTING 1
20 REM (c) Electron User
1985
30 OSWORD=&FFFF1
40 OSWRCH=&FFEE
50 INPUT "Location - hex assumed", location$
60 FOR pass1=0 TO 3 STEP 3
70 PX = EVAL("&"+ locati
on$)
80 I
90 OPT pass1
100 .width NOP
110 .depth NOP
120 .xstep NOP
130 .ystep NOP
140 .xpos NOP:NOP
150 .ypos NOP:NOP
160 .xstore NOP:NOP
170 .ystore NOP:NOP
180 .pattern NOP
190 EQU$ "12345678"
200 .start LDX # pattern
MOD 256
210 LDY # pattern DIV 256
220 LDA # &8A
230 JSR OSWORD
240 LDA ypos
250 STA ystore
260 LDA ypos+1
270 STA ystore+1
280 LDY # 0
290 .outer_loop LDA xpos
300 STA xstore
310 LDA xpos+1
320 STA xstore+1
330 LDA pattern+1,Y
340 LDX # 0
350 .inner_loop ASL A
360 BCC noprint
370 PHA
380 TXA:PHA
390 TYA:PHA
400 JSR write
410 PLA:TYA
420 PLA:TXA
430 PLA
440 .noprint PHA
450 CLC
460 LDA xstore
470 ADC xstep
480 STA xstore
490 LDA xstore+1
500 ADC # 0
510 STA xstore+1
520 PLA
530 INX
540 CPX # 8
550 BNE inner_loop
560 SEC
570 LDA ystore
580 SBC ystep
590 STA ystore
600 LDA ystore+1
610 SBC # 0
620 STA ystore+1
630 INY
640 CPY # 8
650 BNE outer_loop
660 CLC
670 LDA xstore
680 STA xpos
690 LDA xstore + 1
700 STA xpos + 1
710 RTS
720 .write LDA # 5
730 JSR OSWRCH
740 LDA # &19
750 JSR OSWRCH
760 LDA # 4
770 JSR OSWRCH
780 LDA xstore
790 JSR OSWRCH
800 LDA xstore+1
810 JSR OSWRCH
820 LDA ystore
830 JSR OSWRCH
840 LDA ystore+1
850 JSR OSWRCH

```

```

860 LDA # &E0
870 JSR DSMRCH
880 LDA #4
890 JSR DSMRCH
900 RTS
910 .definer LDA # 23
920 JSR DSMRCH
930 LDA # &E0
940 JSR DSMRCH
950 LDA # 0
960 LDX width
970 .define_row SEC
980 ROR A
990 DEY
1000 BNE define_row
1010 LDY depth
1020 .send_rows JSR DSMRCH
1030 DEY
1040 BNE send_rows
1050 .send_blanks
1060 LDY # 0
1070 LDA # 0
1080 .blank JSR DSMRCH
1090 DEY
1100 BNE blank
1110 RTS
1120 ]
1130 NEXT pass%
1140 save%="SAVE FCODE "+location%+"EA"
1150 DSCL save%

```

### Listing 2

```

10 REM LISTING II
20 REM MIKE BIBBY
30 REM (c) 1985
40 ON ERROR GOTO 650
50 VDU 23;0202;0;0;0;
60 I%=&B00
70 CLS :toggle%=FALSE: finish%=FALSE: store%=0%: @I=3
80 x%=1: y%=1: w%=1: d%=1: hold%=?(@I+&B3)
90 INPUT TAB(0,10) "What Mode",mode%
100 PRINT TAB(0,14) "Special character? (Y/N)";
110 REPEAT: A%=GET%: UNTIL INSTR("YN",A%)
120 IF A%="Y" THEN PRINT TAB(0,18) "CHARACTER?";: A%=GET%: ?(@I+&B3)=A%x%=4: y%=4: w%=4: d%=4

```

```

130 MODE mode%: VDU 23;0202;0;0;0;
140 REPEAT
150 PROCf(x%,y%,w%,d%)
160 CLS: PROCw(0,024,"TEST")
170 *FX15,1
180 PRINT TAB(0,0) " x y w d "mode%
190 IF toggle% THEN t%="D" ELSE t%="I"
200 PRINT x%,y%,w%,d%,"t%"
210 PROCkey
220 UNTIL finish%
230 @%=store%: ?&B3=hold%
240 PRINT "Again? (Y/N)";
250 REPEAT: A%=GET%: UNTIL INSTR("YN",A%)
260 IF A%="Y" THEN RUN ELSE END
270 END
280 REM =====
290 DEF PROCkey
300 LOCAL key%,key$,increase%
310 increase%=2+SIGN(toggle%)+1
320 REPEAT
330 key%=GET%
340 UNTIL INSTR("WDXYF"+CHR$(13),key%)
350 key%=ASC(key%)
360 IF key% = 07 THEN w%=w%+increase%
370 IF key% = 08 THEN d%=d%+increase%
380 IF key% = 09 THEN x%=x%+increase%
390 IF key% = 09 THEN y%=y%+increase%
400 IF key% = 13 THEN toggle%=NOT toggle%
410 IF key% = 70 THEN finish%=TRUE
420 ENDPROC
430 REM =====
440 DEF PROCw(x%,y%,w%)
450 LOCAL w%
460 !(@I+4)=x%: !(@I+6)=y%
470 FOR w%=1 TO LEN(w%)
480 ?(@I+&B3)=ASC(MID$(w%,w%,1))
490 CALL (@I+&15)

```



```

500 NEXT
510 ENDPROC
520 REM =====
530 DEF PROCf(x%,y%,w%,d%)
540 LOCAL w%,f%
550 A%=&B7: w% = (USR(&FF4) AND &FF0000) DIV &10000
560 IF w%=1 OR w%=4 THEN f%=4
570 IF w%=2 OR w%=5 THEN f%=0
580 IF w%=0 THEN f%=2
590 x%=f%+x%: y%=y%+4
600 ?(@I+2)=x%: ?(@I+3)=y%
610 ?@I=w%: ?(@I+1)=d%
620 CALL (@I+&B0)
630 ENDPROC
640 REM =====
650 IF ERR=17 THEN ?(@I+&B3)=hold%: RUN
660 REPORT: PRINT " at "; ERL

```

### Listing 3

```

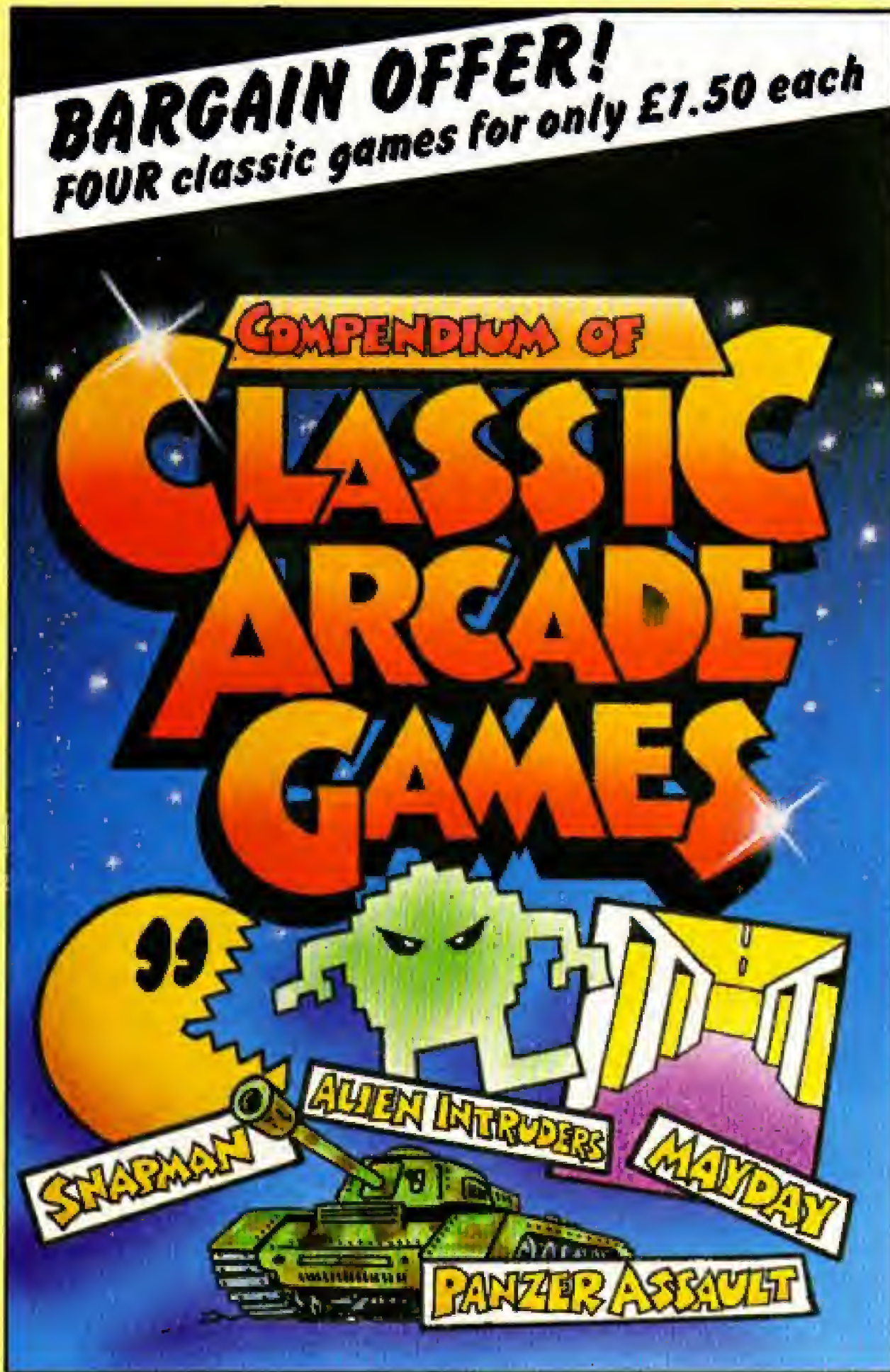
10 REM LISTING III
20 MODE 2
30 location%=&B00
40 PROCdefinefont(2,2,2,2)
50 GCOL0,5
60 PROCwriteword(100,900,"Paul")
70 PROCwriteword(350,600,"and")
80 PROCwriteword(500,300,"Katie")
90 GCOL0,6

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*

Here's something **SPECIAL** from

**BARGAIN OFFER!**  
FOUR classic games for only £1.50 each



electron  
user

We've commissioned four rip-roaring games for the Electron and BBC Micro

Three of this high-powered collection are top-rate machine-code versions of arcade classics and the fourth is a thrilling real-time adventure game. There's hours of enjoyment and something to suit everyone in this unique value for money collection

**SNAPMAN** – Guide your man through the maze as he munches energy pellets and avoids hostile aliens

**ALIEN INTRUDERS** – With only your laser for protection you must destroy the waves of aliens who threaten to engulf you

**PANZER ATTACK** – You are a tank commander engaged in vicious combat against encircling enemy forces

**MAYDAY** – A futuristic adventure! As captain of an interstellar cruiser you must guide the sole survivor of a stricken space freighter through the wreckage of his craft. If you fail to recover those vital medical supplies a whole planet is doomed!

Please send \_\_\_\_\_ copy/copies of  
Classic Arcade Games.  
I enclose a cheque/PO No. \_\_\_\_\_  
for £ \_\_\_\_\_  
made payable to: Database Publications Ltd.

- Electron tape £5.95
  - BBC Micro tape £5.95
  - BBC Micro disc £7.95
- (Please tick)

Name \_\_\_\_\_

Address \_\_\_\_\_

Post code \_\_\_\_\_ Tel: No. \_\_\_\_\_

Post to: Classic Arcade Games offer, Electron User, 68 Chester Road, Hazel Grove, Stockport SK7 5NY

EU/3

# Micro Messages

IT has come to our attention within the last few weeks that a certain number of recent Acornsoft games for the Acorn Electron will not operate with our Electron Joystick Interface.

This appears to be due to Acorn's adoption of a new loader program for their games.

The reasons for this are unclear, but it would appear that the only joystick interface available at present that will work with their most recent games is their own.

We would appreciate it if you would inform the public that there is no cause for alarm since the problem is isolated at present and that we are now in a position to offer a free upgrade service for end users only.

The interface must be sent to: John Smith, Powersoftware, Free E.J.I.F. Upgrade Service, Thorns Road Trading Estate, Thorns Road, Brierley Hill, West Midlands DY5 2JS.

We recommend that the customer packs the interface in a jiffy bag and uses first class recorded delivery.

This will cost them approximately 65 pence, since the unit and packaging weighs approximately 250 grammes.

The upgrade will then be despatched to the customer by

## Software that works

I RECENTLY saw a letter in your magazine asking about BBC software for the Electron. I have a large amount of software from a couple of friends and here is a list of the ones which work. Where possible I have given the publisher's name.

Logo2, Facemaker (Acornsoft?), Constellation, French Mistress (Kosmos), Escape from Orion, Danger UXB (Program Power), Invaders, Planes, \*Galaxy Wars (Bug Byte).

Chess (Program Power), \*\*Swoop (Program Power), Shootout (Program Power), The Wizard, Croaker (Program

Power), \*\*\*Snapper (Acornsoft), Monsters (Acornsoft), Painter, \*\*\*\*Super Invaders, \*\*\*\*\*Metecors (Acornsoft), Jumbo/BBC, Great Britain Limited.

\* Needs analogue input joystick.

\*\* Runs a bit slow.

\*\*\*Load first part and type GOTO 60.

\*\*\*\* Runs dead slow. Would not recommend.

\*\*\*\*\* Load first part and type GOTO 40.

I hope this list proves useful. - **Paul Renold, Liverpool.**

● Thanks Paul. Has anyone got anything to add to this?

return of recorded delivery letter post.

To facilitate identification of Mkl and Mkll products the latter will be marked with a coloured sticker on the packaging and on the unit. - **Ian Smart, Executive Director, Powersoftware.**

● It's nice to hear of such good service Mr Smart, especially since Acorn play their cards very close to their chest and it's no fault of your firm that the interface needs upgrading.

## Don't shoot the Electron

IT seems that a lot of readers are getting in a tizz trying to get their printers to print graphics, but the one that really got my goat was Ken Davies in your January issue.

Please Mr Davies, don't blame your Electron. It's obvious you thought you were going to be able to press a

button and your lovely Mode 2 screen would be printed out. But if, as you say, you have had 40 years in industry, you should have realised that few things are as easy as they are made to appear.

You say you saw your printer connected to a Plus 1. Really? If you saw a caravan hitched up to the same make of car that you own, would you go and buy it? You might do, of course, but not until you had made sure it would suit your needs.

Sorry, Mr Davies, but you should have made sure your printer would do what you expected of it before parting with your retirement cash. And I say again, please don't blame the Electron.

Now for the benefit of readers contemplating buying a printer, I can recommend the Mannesmann Tally MT80.

At around £200 I think it is the perfect printing partner to the Electron. No problems with graphics either!

An excellent graphics dump was printed in *The Micro User* in January 1984 - you might consider re-printing it Mr Editor. (Take a look at this issue - Ed.)

Finally, I think the word processor on your Mini Office tape is great. Coupled with the MT80 printer and an Electron, who could ask for more? - **Philip Lucas, Banbury.**

## Let's get technical

A LOT of your articles seem to be aimed at beginners. Not all of us are at this level, and are rapidly outgrowing the user manual.

Would it be possible to have

more articles on subjects such as advanced graphics or machine code etc?

I was delighted to see the start of Merlin's section. It's always useful to be given tips and methods for adventures since I spend most of my time in mazes or being killed!

In answer to recent letters, Katy King was fed up of having to re-load Twin-Kingdom Valley.

When the program first loads a message "North is a wooden hut..." (etc) appears. At this point (the very start) enter \*SAVE. This saves the start position.

Then when you die, it seems nothing can be done except BREAK but one command does work... \*LOAD.

If you type this a position saved on tape can be loaded in. If you load in your start position then you are back to the beginning.

So what, you might ask. Well the position save and load is about 21 blocks long compared with the 40 to 60 blocks of the actual program.

A copy of the initial position and/or your current position can be kept for future use and loaded in so long as BREAK isn't pressed.

Micro Messages asked for a list of BBC Micro games which work on the Electron.

The following work well, but sometimes the speed is slower or the Mode 7 characters are a bit odd:

Dallas: Everything works well since it is in (listable) Basic.

Chess (Micro Power): Slow computer response, but OK.

Commander (Acornsoft): Seems OK.

Monsters (Acornsoft):



## From Page 49

Mostly fine, but strange things happen past level 2.

Attack on Alpha Centuri: Works fine, but sometimes freezes unretrievably.

Cylon Attack: No problems.

Crazy Painter: Everything but sound is OK.

Danger U.X.B. (Micro Power): Mode 7 only wrong thing.

Vortex: Works well.

Gunsmoke: Very slow movement.

Positron Invaders: The slowness brings this game down to a playable speed.

Chuckie Egg: Rather slow. — **Justin Leese, Porthmadoc.**

## Bank Account hard copy

IN the July issue of your excellent magazine you published a program by Ken Smith entitled Bank Account.

I have been, for some

months, using this program with a paper back-up in case my wife or myself inadvertently lost the datafiles, and have found it very good indeed.

However a minor problem appeared over Christmas with the receipt of a printer. Even by attempting to insert VDU2 etc within the program I could not achieve a respectable hard copy of the data.

The result of these deliberations was a new print procedure added and certain adaptations to the areas where this PROC can be called from.

A page of data can be printed from either the current page on display if looking back over the account (from PROC-readpages) or the last 20 items if the current page is on the screen (from PROC-inputbox).

Listings are enclosed and I hope this will be of interest to yourselves and your readers. — **Gwynne Chivers, Barry, S. Glamorgan.**

It is really pleasing to find

that our readers have tailored Electron User programs to their needs and all the more so when they share them with us.

## Way into hyperspace

I HAVE heard that it is not possible to use the Galactic Hyperspace on the Electron Elite. After saving enough credits to buy Galactic Hyperspace I tried to get it to work. After many frustrating hours I gave up — it just didn't want to know!

Then three weeks later I had travelled to the top of Galaxy 1 and as I was trying to pick up a cargo cannister, one hit me and took away my forward shields.

I hit some keys all at the same time (I think they included Caps Lock and Ctrl) and a message "Galactic Hyperspace" appeared on the

screen. Next thing I knew I was in Galaxy 2.

If there is a method of getting the Galactic Hyperspace to work, please let me know (and many of my friends who also own Electrons). — **Michael Fuller (age 14), Epsom, Surrey.**

● Lots of letters asking this but, as yet, no answers. Anyone able to help?

## Modems coming

I AM thinking of buying a modem for my Electron.

Could you please advise me on the availability and cost of them? — **J. Brennan, Harrow, Middlesex.**

● Before you can use a modem on the Electron you'll need an RS232 interface. Acorn are working on one which will fit into one of the

## ELECTRON Joystick Breakthrough!!

Why waste money on separate interfaces, take advantage of the ELKAN 'Plug-in-and-go' Quickshot II Joysticks (with full auto-fire). And pocket the difference.

**Electron version £23.95**

BBC version £15.95 • Sinclair QL version £11.95  
Dragon/Tandy version £14.95 • Einstein version £15.95

If you really need a separate interface try these for value:

**Electron interface £13.95**

**BBC interface £9.95**

All prices include VAT and FREE postage and packing

Send Cash/Cheques/Visa/Access to:

**ELKAN ELECTRONICS**  
FREEPOST (No stamp required)  
11 Bury New Road, Prestwich  
Manchester M25 6LZ.



24 Hour Hotline  
061-798 7613

```
460 DEFPROCinputbox
470 VDU28,0,31,39,28
480 COLOUR 129:CLS:COLOUR
0
490 PRINT "NEXT ENTRY (SP
ACE)"TAB(20);"TO READ PAGES
(R)"
500 PRINT "TO SAVE DATA (
S)"TAB(20);"TO CHANGE ENTRY
(C)"
505 PRINT "TO PRINT THIS
PAGE (P)"
510 A$=GET$:SOUND 1,-15,8
7,2:CLS
520 IF A$=" " THEN ENDPRO
C
530 IF A$="C" THEN PROCch
ange:PROCinputbox:ENDPROC
C
540 IF A$="R" THEN PROCr
eadpages:PROCinputbox:ENDPRO
C
545 IF A$="P" THEN PROCpr
int:PROCinputbox:ENDPROC
550 IF A$="S" THEN PROCk
epdata ELSE 490
560 PROCinputbox:ENDPROC
1250 DEFPROCreadpages
1260 V=E:V=1:K=1
1270 REPEAT
1280 VDU28,0,31,39,28
1290 COLOUR129:CLS:COLOUR0
1300 PRINT"TO READ ACCOUNT
PAGES PRESS SPACE." "TO CH
ANGE AN ENTRY PRESS (C)."
1305 PRINT"TO PRINT THIS P
AGE PRESS (P)"
1310 A$=GET$:SOUND 1,-15,8
7,2:CLS
1320 IF A$="C" THEN PROCch
ange:GOTO1300
1325 IF A$="P" THEN PROCpr
int:GOTO1300
1330 IF A$=" " THEN 1340 E
LSE 1300
1850 DEFPROCprint
1860 #FX5,1
1870 #FX3,10
1880 LOCAL v,k
1890 k=1
1900 v=V-20:IF v<1 THEN v=
1
1910 VDU2
1920 REPEAT
1930 PRINTTAB(5);M$(v);TAB
(11);E$(v);TAB(22);Q$(v);
1940 IF A(v)=0 THEN PRINTT
AB(29);"-"; ELSE PRINTTAB(2
9);"+";
1950 PRINTTAB(30);C(v);TAB
(40);D(v)
1960 k=k+1:v=v+1
1970 UNTIL k>20 OR v=E
1980 #FX3,4
1990 VDU3
2000 ENDPROC
```

# Don't bust your bus

cartridge slots on the Plus 1. When this is generally available we'll be taking a look at modems.

## Games chart needed

I WOULD like to see in this magazine every month a chart of the best selling computer games for the Electron.

I bought your Micro Olympics when it first came out and I've managed to get 105m on the javelin.

I have only one thing against this computer game and that is having to keep pressing your keys up and down all the time. — **Stephen Brook, Rothwell, Leeds.**

● We take your point Stephen. It would be nice to have a games chart. The trouble is that we've yet to be convinced that a truly independent chart covering all Electron software exists. It's very much a case of people rubbishing the charts unless their game happens to be at the top!

## Deciding the draw

I TYPED in the Christmas Box program from the December issue of Electron User and my brother and I began to play it.

We are fairly good at games like this and found the game ended in a draw, but the micro was still waiting to carry on playing.

So I wrote an extra piece to add to the program.

Change line 170 to:

```
170 UNTIL win% OR go%=30
```

And add line 1111:

```
1111 IF go%=30 THEN Name$(person%)="NO-ONE":GOTO 1170
```

These lines determine whether there is a draw after there have been 30 goes. — **Stephen Manser, Tonbridge, Kent.**

## Spot for a toolkit

A TOOLKIT ROM by Beebugsoft was given me for Christmas, but I'm now won-

I HAVE recently bought a First Byte joystick interface, and find that although it is a good unit, it is a pity that it sticks out at the rear of the micro so much.

I am sure that it could have been designed to have made use of the two captive fixings that are provided for the Plus 1. This would have removed the dreaded thought of snapping off the expansion bus in a moment of carelessness.

To get over this threat, and to make fitting foolproof, I have designed a guide that allows you to plug in without having to up-end your micro.

I made my guide with brass as I was not sure if steel would have any effect in the micro's internals — brass cannot get magnetised.

All sharp edges were filed round and both halves polished so that the interface slides in easy and does not get damaged. It also looks very good.

In the January issue of Electron User, Miss D. Hillage asks for a list of BBC/Electron compatible software.

I have a Docsoft 747 Flight Simulator for the BBC 32k and it works perfectly.

I also have a selection of

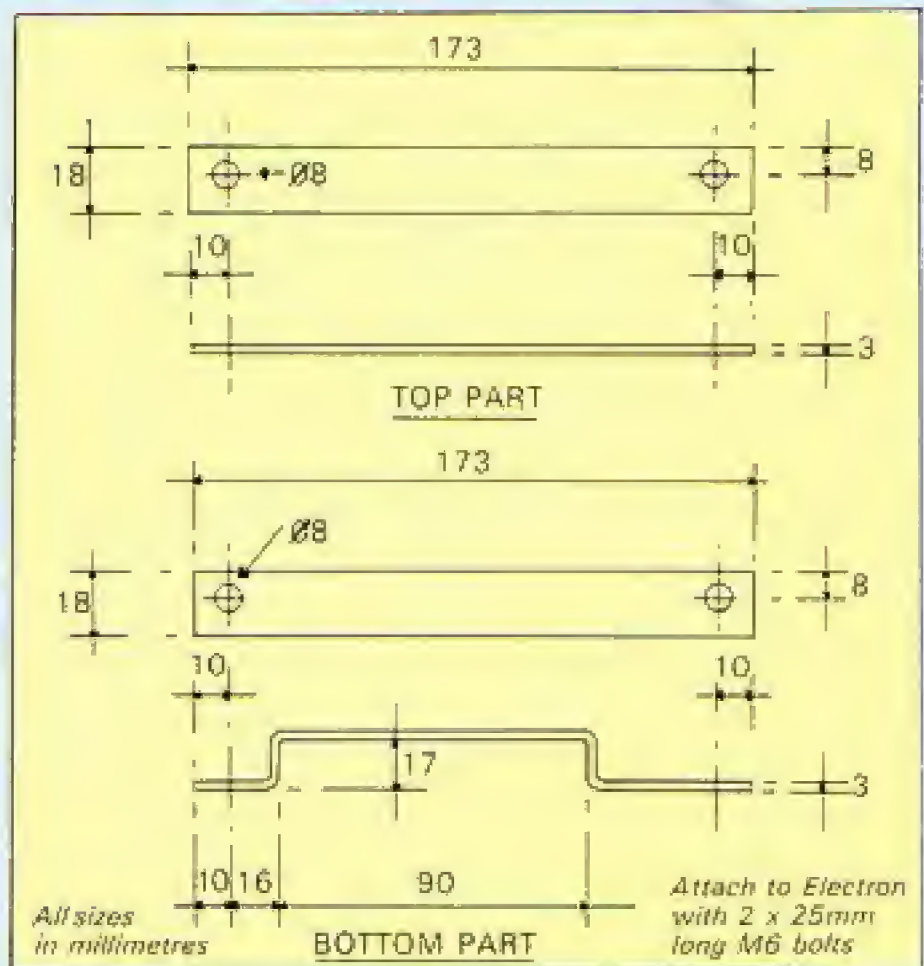
dering where to put it.

Can any extra ROMs be fitted internally to the Electron, as with the BBC, or is an expansion board required?

If the latter, are there any boards compatible with the Plus-1? The Beebugsoft instructions were unusually ambiguous on this point. — **Mark Long, Liverpool.**

● You've got two choices Mark. One is the Mushroom ROM card the other is the Slogger ROM box.

Both are excellent pieces of equipment that attach to the



programs which were listings for the BBC out of another magazine, which I used to get, so that I could decide which micro I was going to buy.

I have now found the perfect magazine — Electron User — and look no further. — **J.W. Boyle, Kilwinning, Ayrshire.**

● Thanks for the tip Mr Boyle. Having said that, we've never heard of the expansion bus being snapped off by accident, the First Byte being a fairly robust bit of kit. Still human nature being what it is, someone somewhere will try so your little device will come in useful.

back of your Electron and allow the use of ROMs.

## Shortage in Australia

BECAUSE the Electron isn't as popular here as it is overseas, owners have trouble purchasing English software.

The only software we can get is Acornsoft, which costs \$30 (£15).

If the English software houses sent their goods over

here it would not only please Electron owners but it would also boost the sales of Electrons.

Is there anything we can do? Please help. — **A. Meek, Belair, South Australia.**

● We contacted some of our leading software houses who seemed amazed that there was an Australian market.

They all said that they would investigate the possibilities, so maybe things will look up down under.

## Nice try!

I AM a proud owner of an Electron.

I am curious to know if Micro Power stockists would buy back any Micro Power software that I have become bored of playing.

If so what are their rates. — **Paul Tempest, Collingham, Wetherby.**

● We don't hold out much hope, Paul. But nice try!

**WHAT** would you like to see in future issues of Electron User?

What tips have you picked up that could help other readers?

Now's here is your opportunity to share your experiences.

Remember that these are the pages that you write yourselves. So

tear yourself away from your Electron keyboard and drop us a line.

The address is:

Micro Messages  
Electron User  
Europa House  
68 Chester Road  
Hazel Grove  
Stockport  
SK7 5NY.

# Mr Freeze

By ROLAND WADDILOVE

**MR FREEZE's frozen food factory has been sabotaged by one of his rivals. Several electric fires have been placed among the ice blocks to melt them and destroy the food.**

Each fire is on a timer and they are set to go off in sequence.

As each fire starts to count down to zero you must run to it

and switch it off.

If you are too slow the ice melts and you are electrocuted.

You can jump from one ice block to another or you can slide them left or right, but you can't jump into the spaces in between the ice blocks.

To make matters worse there is a strange springy, bouncy thing which gets in

your way - you must avoid this.

The program consists of about 3k of Basic, 1k of data, 1k of machine code and runs in Mode 2.

However 1k of machine code means 6 or 7k of assembler, making the program much too long.

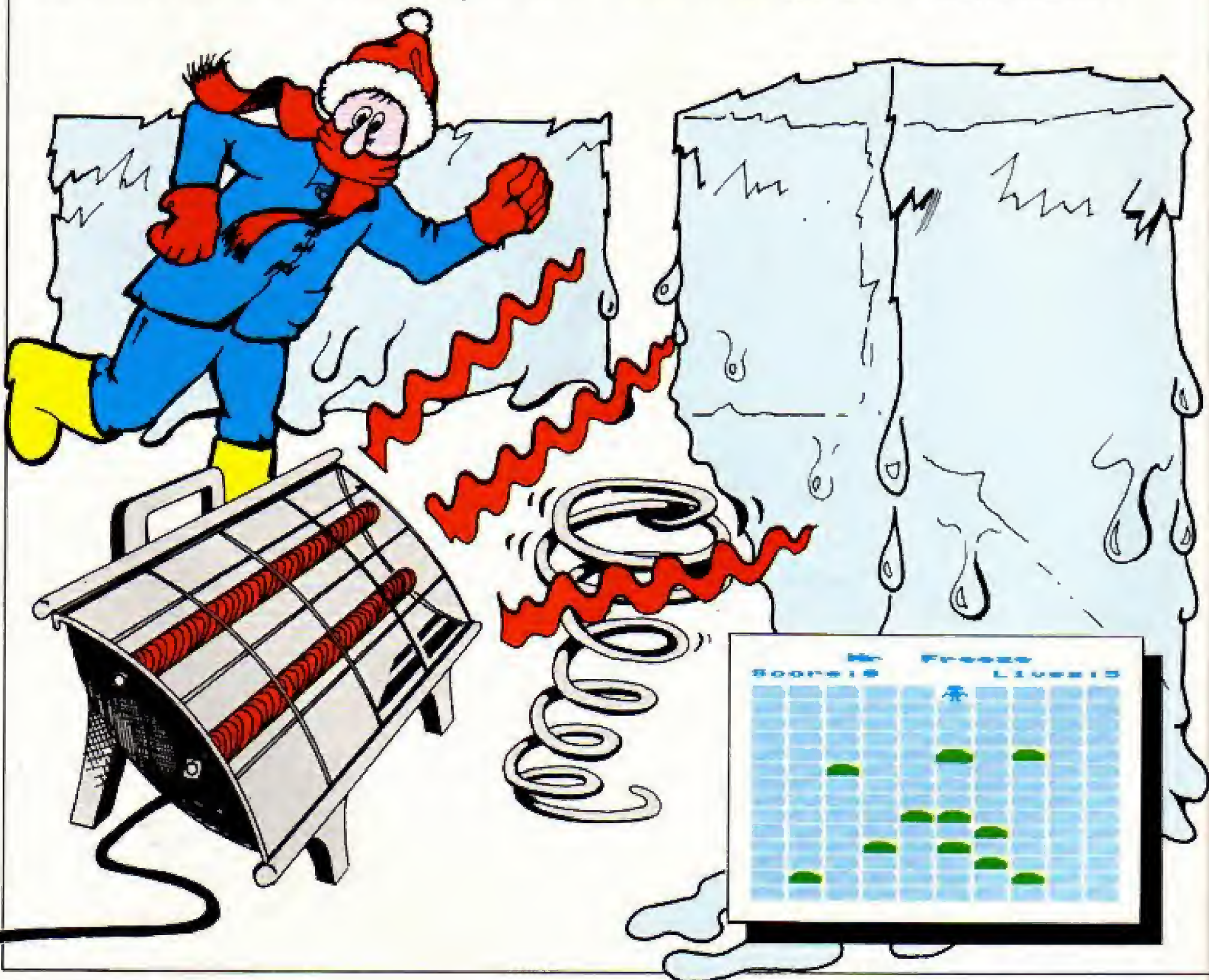
The simple answer is to delete all the data and

assembler when the machine code has been set up.

This is automatically done by the program, so do not run the program unless you have saved it at least twice as half of it will disappear!

If there is an error you will have to reload it so please check it through carefully first.

*Remember. Do not RUN it before you've SAVED it.*



## PROCEDURES

**PROCInstructions**

**PROChi\_score**

**PROCInitialise**

**PROCgame**

**PROCpause**

**PROCassemble**

Prints the instructions and large title.  
Prints the final score and lives. Inputs your name if it is a new hi score. Asks if you want to play again.  
Sets repeat for keys and flash rate for colours. Sets the variables and dims the array for the fires.  
Sets up a copy of the screen at &C00, sets the positions of the fires in the copy and the array. Calls the machine code to draw the screen. Sets the variables, selects a random fire and starts the game. Ends when dead or no fires left.  
Wait.  
Assembles the machine code. POKES the data (any labels referring to a

penguin actually move the bouncy thing). Defines the function key to delete the assembler.

## VARIABLES

**lives** Lives left.  
**name\$** Name of person with high score.  
**I%,J%,K%** General variables.  
**score** Score.  
**hi** Hi score.  
**screen** Screen.  
**fire%(25)** Positions of fires.  
**X%,Y%** Man coordinates.  
**x%,y%** Pointer to old coordinates of man.  
**dead%** Pointer to flag to show whether in fire.  
**counter%** Pointer to counter.  
**xpos%,ypos%** Pointer to coordinates of fire counting.

```

10 REM *****
****
20 REM * Mr Freeze
*
30 REM * By Mr Maddilov
e *
40 REM * (C) Electron Us
er *
50 REM *****
****
60 REM * DO NOT RENUMBE
R *
70 REM *SAVE BEFORE RUNN
ING*
80 REM *****
****
90 MODE 6:PROCassemble:É
ND
100 ON ERROR DSCLI "FX12,
0":VDU22,6:REPORT:PRINT" at
line ";ERL:END
110 HIMEM=42000
120 PROCInstructions
130 PROCInitialise
140 REPEAT
150 FOR lives=5 TO 1 STEP
-1
160 REPEAT PROCgame
170 UNTIL 2*IX<>?xpos% OR
YX<>fireX(IX)DIV10
180 NEXT
190 PROChi_score
200 UNTIL KX<>ASC"y"
210 +FX12,0
220 MODE 6
230 END
240
250 DEF PROCInstructions
260 VDU 22,1,23,1,0;0;0;0
270 name$="Mr Freeze":PR
INT TAB(15,1);
280 FOR IX=1 TO 10:?&70=A
5C(MID$(name$,IX,1)):AZ=10:
IX=&70:YX=0:CALL &FFF1:FOR
JX=0 TO 1:VDU 23,224:FOR KX
=2 TO 9:VDU ?(&70+JX*4+KXDI
V2):NEXT:VDU 224,10,0:NEXT:
VDU11,11,9:NEXT
290 COLOUR 2:PRINT""Mr
Freeze's frozen food factor
y has been""sabotaged by o
ne of his rivals. Electric"
"fires have been placed as
ongst the ice""blocks to
melt them."
300 COLOUR 1:PRINT""The
fires switch on one at a ti
me. You""must switch each
one off before the""time
r reaches zero."
310 COLOUR 2:PRINT""You
can jump from block to bloc
k and""slide left or righ
t."
320 COLOUR 3:PRINT"" S=u
p X=down (=left
)=right"" A=slide left":
SPC(13);?"=slide right"
330 COLOUR 129:COLOUR 2:P
RINT TAB(10,31);" Press spa
ce to start ";CHR$(17);+FX2
1,0
340 REPEAT UNTIL GET=32
350 ENDPROC
360
370 DEF PROChi_score
380 COLOUR 128:COLOUR 6:P
RINT TAB(6,2);score;SPC(3);
TAB(19,2);lives
390 COLOUR 5:+FX11,0
400 +FX21,0
410 IF score>hi hi=score:
PRINT TAB(0,10);"Best score
so far !":VDU 23,1,1;0;0;0
:PRINT""What is your name
?"""STRINGS(15,".")'CHR$(
11);:INPUT "name$:VDU 23,
1,0;0;0;0:;name$="By "+LEFT
$(name$,15)
420 COLOUR 2:VDU 20,0,31,
19,5,12,26
430 PRINT TAB(3,10);"High
score=";hi;TAB(120-LEN nam
e$)DIV2,15);name$
440 COLOUR 129:COLOUR 3:P
RINT TAB(2,25);" Another 6
ame ? "+FX21,0
450 REPEAT KX=GET OR 32:U
NTIL KX=ASC"y" OR KX=ASC"n"
460 score=0:screen=1
470 +FX11,1
480 ENDPROC
490
500 DEF PROCInitialise
510 +FX9,5
520 +FX10,5
530 +FX11,1
540 +FX12,10
550 +FX16,0
560 ENVELOPE 1,1,1,-1,0,4
,4,0,126,0,0,-126,126,126
570 DIM fireX(25)
580 hi=100:score=0:screen
=1
590 xX=&76:yX=&77:deadX=0
7A:counterX=&78:xposX=&7C:y
posX=&7D
600 VDU 22,2,5
610 GCOL 0,4:MOVE 330,102
3:PRINT name$:GCOL 0,3:MOVE
330,1019:PRINT name$
620 VDU 4,23,1,0;0;0;0;
630 FOR IX=0 TO 15:VDU 19
,IX,1;0;:NEXT
640 COLOUR 6:PRINT TAB(0,
2);"Score:";TAB(13,2);"Live
s:"
650 ENDPROC
660
670 DEF PROCgame
680 +FX202,0
690 FOR IX=0 TO 200:IX?&C
00=1:NEXT
700 fireX(0)=RND(128)+10:
?(&C00+fireX(0))=2
710 FOR IX=1 TO 9+screen
720 REPEAT fireX(IX)=RND(
129):ok=TRUE
730 FOR JX=0 TO IX-1
740 IF fireX(IX)=fireX(JX
) OR fireX(IX)=5 ok=FALSE
750 NEXT
760 UNTIL ok
770 ?(&C00+fireX(IX))=2
780 NEXT
790 CALL HIMEM
800 ?&72=&80:??&73=&8:IX=5
:YX=13:CALL MX
810 XZ=5:YX=0:??xZ=5:??yZ=0
:??&C05=0
820 ?&7E=5:??&7F=13
830 IX=-1
840 REPEAT IX=IX+1
850 COLOUR 128:COLOUR 6:P
RINT TAB(6,2);score;SPC(3);
TAB(19,2);lives
860 ?yposZ=2+(fireX(IX)DI

```

# Mr Freeze listing

## From Page 53

```

V10)+5:JX=-1:REPEAT JX=JX+1
:UNTIL ?(C00+10*(fireX(IX)
DIV10)+JX)=2:?xposX=JX*2:?c
ounterX=499:KX=0
 870 COLOUR 130:COLOUR 1:*
FX21,0
 880 REPEAT TIME=0
 890 YX=(YX-(KX=88)+(KX=83
))MOD14:IF YX=-1 YX=13
 900 XX=(XX-(KX=46)+(KX=44
))MOD10:IF XX=-1 XX=9
 910 CALL CX:CALL NX:REPEA
T UNTIL TIME>10
 920 UNTIL ?deadX=2 OR ?co
unterX=0
 930 IF 2+YX=?xposX AND YX
=fireX(IX)DIV10 SOUND 0,-15
,4,5:?72=L00:?73=L0A:CALL
LX ELSE VDU 19,3,11;0;19,4
,12;0;:SOUND 1,1,4,40:PROCP
ause(200):VDU 19,3,3;0;19,4
,4;0;
 940 score=score+10+(1?cou
nterX AND &F0)/16)+(1?counte
rX AND &F)
 950 UNTIL IZ=screen+9 OR
(2*XX<>?xposX OR YX<>fireX(
IX)DIV10)
 960 FOR JX=200 TO 8 STEP
-8:SOUND 1,1,JX,3:NEXT:VDU
24,0;0:1272;14*64;16,26
 970 IF IZ=screen+9 screen
=screen+1
 980 ENDPROC
 990
1000 DEF PROCpause(delay)
1010 TIME=0:REPEAT UNTIL T
IME>delay
1020 ENDPROC
1030
1040 REM **ice block**
1050 DATA 10101000,1010101
0,34303000,303E303D,3C30300
0,3F3F303F,3C303000,3C3F3E3
F,30303000,3F3E303E,3F30300
0,3F3F3E3D,3C303000,3D3D3F3
F,0,20202020
 1060 DATA 10101010,1010,3E
3F303D,3E3E3D,3D3D3F3E,3C3F
3E,3D3D3F3F,3E3D3E,3F3C3F3F
,3C3E3F,3F3F3D3D,3C3F3C,3E3
E3F3D,3C3F3F,202A2020,2A202
0
 1070 REM **electric fire**
1080 DATA 0,4040000,400000
0,9090C04,C0000,1030C0C,C00
00,3010C0C,C0000,1030C0C,40

```

```

00000,3010C0C,0,4060C0B,0,0
000000
 1090 DATA 4040404,404,9090
C0C,C0C,1030C0C,C0C,3010C0C
,C0C,1030C0C,C0C,3010C0C,C0
C,4060C0C,C0C,0000000,000
 1100 REM **man**
 1110 DATA 0,0,0,0,5111100,
30000000,F0A2733,3000050A,F
0A0F33,300A0F00,F183322,300
0000A,0,20000000,0,0
 1120 DATA 0,0,5101000,0,20
30,0,30303030,30101010,3030
3030,10101010,20202030,2000
0000,5103020,0,0,0
 1130 REM **blank**
 1140 DATA 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0
 1150 DATA 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0
 1160 REM **ball**
 1170 DATA 0,0,0,0,0,C0C0C0
40,C0000000,C0C0C0C0,C00000
00,C0C0C0C0,0,C0C0C000,0,0,
0,0
 1180 DATA 0,0,0,0,40C0,400
000,C0C0C0,C0C0C000,C0C0C0,
C0C0C000,00C0,000000,0,0,0,
0
 1190
1200 DEF PROCassemble
1210 FOR IX=L900 TO &AFC S
TEP 4:READ a$:!IX=EVAL("k"+
a$):NEXT
 1220 FOR IZ=L800 TO &BFC S
TEP 4:READ a$:!IZ=EVAL("k"+
a$):NEXT
 1230 px=L7E:py=L7F:screen
=L70:data=L72:temp=L74:xX=L
76:yX=L77:XX=L460:YX=L464:d
ead=L7A
 1240 counter=L78:xpos=L7C:
ypos=L7D:KX=L42C:osbyte=!&2
0A AND &FFFF:oswrch=!&20E A
ND &FFFF:osword=!&20C AND &
FFFF:sound=L80:!sound=LFFF6
0001:sound!4=L00010000
 1250 sounddata=L80:!soundd
ata=L9CA4A000:sounddata!4=L
00009094
 1260 FOR pass=4 TO 6 STEP
2
 1270 PX=L2C00:OX=L5000
 1280 [ OPT pass
 1290 \set up screen
 1300 LDA #L8B:STA screen:L
DA #L0C:STA screen+1 \scre
en copy
 1310 LDA #L9:STA data+1 \

```

```

address of data for char
1320 LDY #13:STY yX
1330 .loop1
1340 LDX #9:STX xX
1350 .loop2
1360 LDY #0:LDA (screen),Y
\ice or fire?
1370 CMP #1:BEQ here
1380 LDY #L00
1390 .here
1400 STY data
1410 LDX xX:LDY yX:JSR pri
nt_char
1420 DEC screen
1430 DEC xX:BPL loop2
1440 DEC yX:BPL loop1
1450 LDA #L20:STA data:LDA
#L0A:STA data+1:LDX #5:LDY
#0:JSR print_char \man
1460 RTS
1470
 1480 .print_char \data=ch
aracter data, X/Y=coordinat
es
 1490 LDA #L00:STA temp:LDA
#L34:STA temp+1
1500 CLC
1510 .loop1
 1520 LDA temp:ADC #L40:STA
temp:LDA temp+1:ADC #L0:ST
A temp+1
1530 DEY:BPL loop1
1540 .loop1
 1550 LDA temp+1:ADC #L05:S
TA temp+1
 1560 DEY:BPL loop1
 1570 LDA data+1:CMP #L8:BN
E pc:RTS
1580 .pc
 1590 LDY #63
1600 .loop1
 1610 LDA (data),Y:STA (tea
p),Y
 1620 DEY:BPL loop1
 1630 LDA temp:ADC #L40:STA
temp:LDA temp+1:ADC #L2:ST
A temp+1
 1640 LDY #127
1650 .loop1
 1660 LDA (data),Y:STA (tea
p),Y
 1670 DEY:CPY #64:BPL loop1
1680 RTS
1690
 1700 .above_man
 1710 LDA #31:JSR oswrch:LD
A xpos:JSR oswrch:LDA ypos:
JSR oswrch \PRINT TAB(1)

```

```

1720 SED:SEC:LDA counter:S
BC #1:STA counter:CLD \cou
nter=counter-1
 1730 AND #LFB:LSR A:LSR A:
LSR A:LSR A:CLC:ADC #L0:JSR
oswrch \1st digit
 1740 LDA counter:AND #L0F:
CLC:ADC #L0:JSR oswrch \2n
d digit
 1750 LDA #129:LDX #0:LDY #
0:JSR osbyte \INKEY(0)
 1760 INY:BNE key_pressed:S
TY KX:JMP not_escape
 1770 .key_pressed DEY:BEQ
not_escape
 1780 LDA #126:JSR osbyte:R
TS \escape pressed
 1790 .not_escape STX KX \
store key
 1800 LDA #21:LDX #0:LDY #0
:JSR osbyte \*FX21,0
 1810 JSR play_sound:LDA KX
\get key pressed
 1820 CMP #ASC"A":BNE m01:J
MP slide_left
 1830 .m01 CMP #ASC"/":BNE
m02:JMP slide_right
 1840 .m02
 1850 LDA YX:ASL A:ASL A:AS
L A:ADC YX:ADC YX:ADC XZ \
get screen address
 1860 TAY:LDA &C00,Y:STA de
ad:BNE ok \see if space
 1870 LDA xX:STA XZ:LDA yX:
STA YZ:RTS \restore old co
ordinates
 1880 .ok LDA #0:STA &C00,Y
\set position to zero
 1890 LDA #L00:STA data:LDA
#L0A:STA data+1 \blank squ
are
 1900 LDA #19:JSR osbyte \
*FX19
 1910 LDX xX:LDY yX:JSR pri
nt_char \erase old man
 1920 LDA #L0:STA data:LDX
XZ:STX xX:LDY yX:STY yX \s
tore present coordinates
 1930 JSR print_char \prin
t new man
 1940 LDA px:CMP xX:BNE m05
 1950 LDA py:CMP yX:BNE m05
 1960 LDA #2:STA dead
 1970 .m05 RTS
1980
 1990 .slide_left
2000 LDA yX:ASL A:ASL A:AS
L A:ADC yX:ADC yX \get scr

```

```

een address
2010 TAY:LDA &C00,Y:PHA:PH
A \save left character
2020 LDX #9
2030 .loop1
2040 LDA &C01,Y:STA &C00,Y
2050 INY:DEX:BNE loop1
2060 PLA:STA &C00,Y:STY sc
reen
2070 LDX x%:DEX:BPL sl
2080 LDX #9
2090 .sl
2100 STX x%:STX x%
2110 LDA #&00:STA data:LDA
#&00:STA data+1:LDX px:LDY
py:JSR pen_print \print pe
nquin
2120 LDA #&00:STA temp:LDA
#&35:STA temp+1
2130 LDY y%:CLC
2140 .loop1
2150 LDA temp+1:ADC #&05:S
TA temp+1
2160 DEY:BPL loop1
2170 LDA temp:ADC #&40:STA
data:LDA temp+1:ADC #0:STA
data+1
2180 LDX #19 \scroll left
2190 .loop1
2200 LDY #63
2210 .loop2
2220 LDA (data),Y:STA (tem
p),Y
2230 DEY:BPL loop2
2240 CLC
2250 LDA temp:ADC #&40:STA
temp:LDA temp+1:ADC #0:STA
temp+1
2260 LDA data:ADC #&40:STA
data:LDA data+1:ADC #0:STA
data+1
2270 DEX:BNE loop1
2280 PLA:BNE sla
2290 LDX #&00:LDY #&A:JMP
slb
2300 .sla TAX:DEX:BNE slb
2310 LDX #&0:LDY #&9:JMP sl
d
2320 .slb LDX #&00:LDY #&9
2330 .sls STX data:STY dat
a+1:LDX #9:LDY y%:JSR print
_char
2340 LDA #&00:STA data:LDA
#&0A:STA data+1:LDX x%:LDY
y%:JSR print_char
2350 LDA y%:ASL A:ADC #5:C
MP ypos:BNE slend
2360 DEC xpos:DEC xpos:BPL
slend
2370 LDA #10:STA xpos
2380 .slend
2390 LDA py:CMP y%:BNE sls
2400 DEC px:BPL sls
2410 LDA #9:STA px
2420 .sls
2430 LDA #&00:STA data:LDA
#&00:STA data+1:LDX px:LDY
py:JSR pen_print \print pe
nquin
2440 RTS
2450
2460 .slide_right
2470 LDA y%:ASL A:ASL A:AS
L A:ADC y%:ADC y%:ADC #9 \
get screen address
2480 TAY:LDA &C00,Y:PHA:PH
A \save right character
2490 LDX #9
2500 .loop1
2510 LDA &BFF,Y:STA &C00,Y
2520 DEY:DEX:BNE loop1
2530 PLA:STA &C00,Y:STY sc
reen
2540 LDX x%:INX:CPX #10:BN
E sr
2550 LDX #0
2560 .sr
2570 STX x%:STX x%
2580 LDA #&00:STA data:LDA
#&0B:STA data+1:LDX px:LDY
py:JSR pen_print \print pe
nquin
2590 LDA #&C0:STA temp:LDA
#&39:STA temp+1
2600 LDY y%:CLC
2610 .loop1
2620 LDA temp+1:ADC #&05:S
TA temp+1
2630 DEY:BPL loop1
2640 SEC:LDA temp:SBC #&40
:STA data:LDA temp+1:SBC #0
:STA data+1
2650 LDX #19 \scroll righ
t
2660 .loop1
2670 LDY #63
2680 .loop2
2690 LDA (data),Y:STA (tem
p),Y
2700 DEY:BPL loop2
2710 SEC
2720 LDA temp:SBC #&40:STA
temp:LDA temp+1:SBC #0:STA
temp+1
2730 LDA data:SBC #&40:STA
data:LDA data+1:SBC #0:STA
data+1
2740 DEX:BNE loop1
2750 PLA:BNE sra
2760 LDX #&00:LDY #&A:JMP
srd
2770 .sra TAX:DEX:BNE srb
2780 LDX #&0:LDY #&9:JMP s
rd
2790 .srb LDX #&00:LDY #&9
2800 .srd STX data:STY dat
a+1:LDX #0:LDY y%:JSR print
_char
2810 LDA #&00:STA data:LDA
#&0A:STA data+1:LDX x%:LDY
y%:JSR print_char
2820 LDA y%:ASL A:ADC #5:C
MP ypos:BNE srend
2830 INC xpos:INC xpos:LDA
xpos:CMP #20:BNE srend
2840 LDA #0:STA xpos
2850 .srend
2860 LDA pv:CMP y%:BNE sr5
2870 INC px:LDA px:CMP #10
:BNE sr5
2880 LDA #0:STA px
2890 .sr5
2900 LDA #&00:STA data:LDA
#&0B:STA data+1:LDX px:LDY
py:JSR pen_print \print pe
nquin
2910 RTS
2920
2930 .play_sound
2940 LDA K%:BNE note:RTS
2950 .note
2960 LDA counter:ASL A:ASL
A:ADC counter:AND #7 \RND
(7)-1
2970 TAY:LDA sounddata,Y:S
TA sound+4
2980 .beep
2990 LDX #sound MOD256:LDY
#sound DIV256:LDA #7:JMP o
sword
3000
3010 .penquin
3020 LDA counter:AND #1:BE
0 pg8:RTS
3030 .pg8
3040 LDA #0:STA sound+4:JS
R beep
3050 LDA px:PHA:LDA py:PHA
3060 LDA counter:ASL A:ASL
A:ADC counter:AND #3 \AND
(7)-1
3070 TAX:BNE pg1
3080 INC px:JMP pengi
3090 .pg1 DEX:BNE pg2
3100 DEC px:JMP pengi
3110 .pg2 DEX:BNE pg3
3120 INC py:JMP pengi
3130 .pg3 DEC py
3140 .pengi
3150 LDA px:BPL pg4:LDA #9
:STA px
3160 .pg4 CMP #10:BNE pg5:
LDA #0:STA px
3170 .pg5 LDA py:BPL pg6:L
DA #13:STA py
3180 .pg6 CMP #14:BNE pg7:
LDA #0:STA py
3190 .pg7 LDA #&00:STA dat
a:LDA #&00:STA data+1
3200 PLA:TAY:PLA:TAX:JSR p
en_print
3210 LDX px:LDY py:JSR pen
_print
3220 LDA px:CMP x%:BNE pge
nd
3230 LDA py:CMP y%:BNE pge
nd
3240 LDA #2:STA dead
3250 .pgend RTS
3260
3270 .pen_print \X,Y=coor
dinates
3280 TXA:PHA:TYA:PHA:JSR p
rint_char:PLA:TAY:PLA:TAX
3290 LDY #63:.loop1 LDA (d
ata),Y:EOR (temp),Y:STA (te
mp),Y:DEY:BPL loop1
3300 LDA temp:ADC #&40:STA
temp:LDA temp+1:ADC #&2:ST
A temp+1
3310 LDY #127:.loop1 LDA (
data),Y:EOR (temp),Y:STA (t
emp),Y:DEY:CPY #64:BPL loop
1
3320 RTS
3330 ]
3340 NEXT
3350 CX=move_man:L%:print_
char:M%:pen_print:N%:penqui
n
3360 *KEY0 "GDEL.60,90:MI
GDEL.1030,3300:M:GF.1%:0 TO
PX:1%?2C00=1%?5000:N:IMR
UN:M"
3370 PRINT "Thinking..."
3380 *FX130,0,128
3390 ENDPROC

```

*This listing is included in this month's cassette tape offer. See order form on Page 61.*

# Make light work of listings

ONLY  
£3.75  
each

To save your fingers most of the listings in *Electron User* have been put on tape.

**On the March 1985 tape:**  
**MR. FREEZE** Ice cube arcade action. **SCREENDUMP** Two procedures for printer dumps. **FILLER** The machine code fill routine. **FRED'S WORD GAME** Educational fun. **BIG LETTERS** Large text utility. **PERCY** Beat the burning fuse. **ANIMATION** Two example programs. **PIGS** Flying bacon. **NOTEBOOK** Display formatting.

**On the February 1985 tape:**  
**CRAAL** The mystifying maze adventure. **BOUNCY** Addictively annoying action. **PAIRS** Can you remember the cards? **BASE A** Binary/hexadecimal conversion utility. **CATCHER** Collect the eggs before they break. **CLOCK** Time-keeping utility. **RACER** Grand Prix action. **NOTEBOOK** Graphics windows. **TRIG** All the right angles.

**On the January 1985 tape:**  
**SPACE BATTLE** Destroy the deadly descending aliens! **NEW YEAR A** Sound and graphics greeting. **ESCAPE FROM SCARGOV** Minefield action. **PIE CHART** Statistics made simple. **CLAYPIGEON** An Electron birdshoot. **ORGAN** Music maestro please! **NOTEBOOK** An original program. **RANDOM NUMBERS** Or not so random! **SNAKES** Reptilian arcade action. **CHEESE RACE** Bean rival race.

**On the December 1984 tape:**  
**CHRISTMAS BOX** Align the presents logically. **SILLY SANTA** Sort out the muddle. **SNAP** Match the Xmas pictures. **RECOVERY** The Bad Program message tamer. **CAROL** Interrupt driven music. **AUTOBATA** A program that grows and grows. **NOTEBOOK** Simple string handling.

**On the November 1984 tape:**  
**STAR FIGHTER** Anti-alien missions. **SCROLLER** Wrap around machine code. **URBAN SPRAWL** Environmental action game. **SPELL** Alphabetic education. **JUMPER** Level headed action. **CAESAR** Code breaking broken. **KEYBOARD** Typing game.

**On the October 1984 tape:**  
**BREAKFREE** Classic arcade action. **ALPHASWAP** A logic game to strain your brain. **SOUND GENERATOR** Tune the Electron's sound channels. **MULTICHARACTER GENERATOR** Complex characters made simple. **RIGEL 5** Out of this world graphics. **MAYDAY** Help with your Morse code. **NOTEBOOK** Palindromes and string handling.

**On the September 1984 tape:**  
**HAUNTED HOUSE** Arcade action in the spilt world. **SPLASH** A logic game for non-swimmers. **SORT SHOWS** How sorting algorithms work. **SORT TIME** The time they take. **CLASSROOM INVADERS**

Multicoloured characters go to school. **SAILOR** Nautical antics. **MATHS TEST** Try out your mental powers.

**On the August 1984 tape:**  
**SANDCASTLE** The Electron seaside outing. **KNOCKOUT** Bouncing balls batter brick walls. **PARACHUTE** Keep the skydivers dry. **LETTERS** Large letters for your screen. **SUPER-SPELL** Test your spelling. **ON YOUR BIKE** Pedal power comes to your Electron. **SCROLLER** Sliced strings slide sideways.

**On the July 1984 tape:**  
**GOLF** A day on the links with your Electron. **SOLITAIRE** The classic solo logic game. **TALL LETTERS** Large characters made simple. **BANK ACCOUNT** Keep track of your money. **CHARTIST** 3D graphs. **FORMULAE** Areas, volumes and angles.

**On the June 1984 tape:**  
**MONEY MAZE** Avoid the ghosts to get the cash. **CODE BREAKER** A mastermind is needed to crack the code. **ALIEN** See little green men - the Electron way! **SETUP** Colour companions without tears. **CRYSTALS** Beautiful graphics. **LASER SHOOT OUT** An intergalactic shooting gallery. **SMILER** Have a nice day!

**On the May 1984 tape:**  
**RALLY DRIVER** High speed car control. **SPACE PODS** More aliens to annihilate. **CODER** Secret messages made simple. **FRUIT MACHINE** Spin the wheels to win. **CHASER** Avoid your opponent to survive. **TIC-TAC-TOE** Electron noughts and crosses. **ELECTRON DRAUGHTSMAN** Create and save Electron masterpieces.

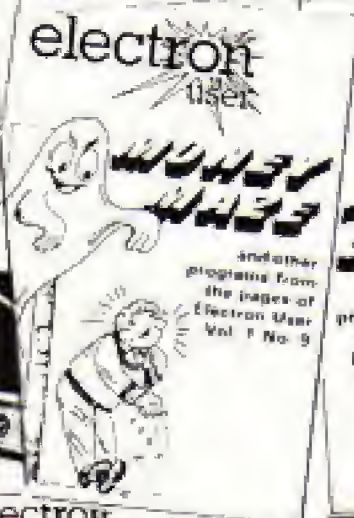
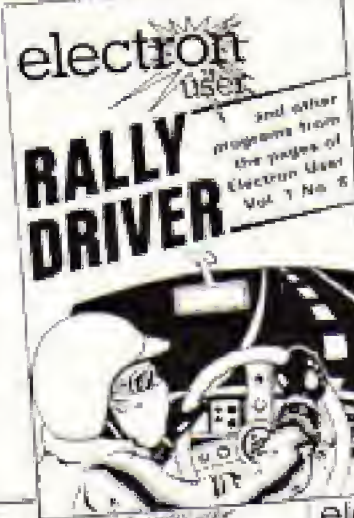
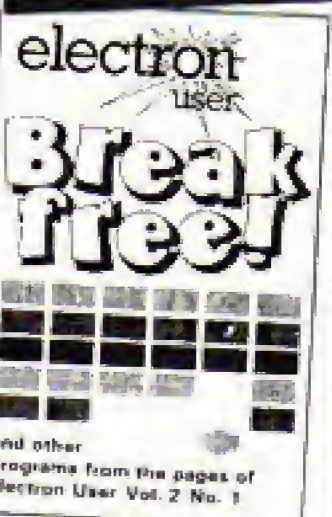
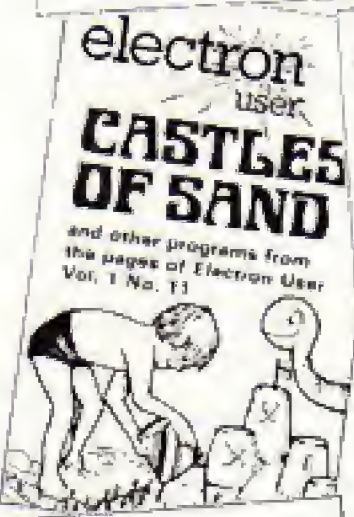
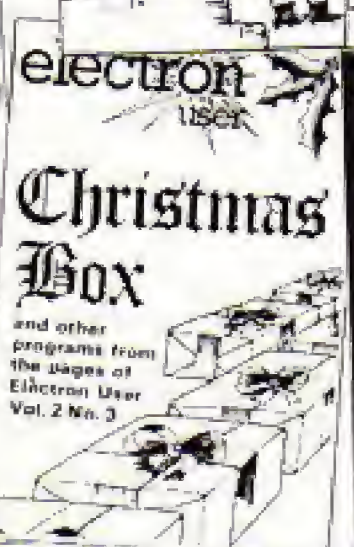
**On the April 1984 tape:**  
**SPACEHIKE** A hopping arcade classic. **FRIEZE** Electron wallpaper. **PELICAN** Cross roads safely. **CHESSTIMER** Clock your moves. **ASTEROID** Space is a minefield. **LIMERICK** Automatic rhymes. **ROMAN** Numbers in the ancient way. **BUNNYBLITZ** The Easter program. **DOGGUICK** The classic logic game.

**On the March 1984 tape:**  
**CHICKEN!** Let dangerous drivers test your nerve. **COFFEE** A fantasising word game from Down Under. **PARKY'S PERIL** Parky's lost in an invisible maze. **REACTION TIMER** How fast are you? **BRAINTEASER** A puzzling program. **COUNTER** Mental arithmetic can be fun! **PAPER, SCISSORS, STONE** Out-guess your Electron. **CHARACTER GENERATOR** Create shapes with this utility.

**On the February 1984 tape:**  
**NUMBER BALANCE** Test your powers of mental arithmetic. **CALCULATOR** Make your Electron

a calculator. **DOILIES** Multi-coloured patterns galore. **TOWERS OF HANOI** The age old puzzle. **LUNAR LANDER** Test your skill as an astronaut. **POSITRON INVADERS** A version of the old arcade favourite.

**On the introductory tape:**  
**ANAGRAM** Sort out the jumbled letters. **DOODLE** Multicoloured graphics. **EUROMAP** Test your geography. **KALEIDOSCOPE** Electron graphics run riot. **CAPITALS** New upper case letters. **ROCKET, WHEEL, CANDLE** Three fireworks programs. **BOMBER** Drop the bombs before you crash. **DUCK** Simple animation. **METEORS** Collisions in space.



## HOW TO ORDER

Please send me the following *Electron User* cassette tapes:

- Sixteen programs from the March 1985 issue ..... £
- Fourteen programs from the February 1985 issue ..... £
- Ten programs from our January 1985 issue ..... £
- Nine programs from the December 1984 issue ..... £
- Nine programs from the November 1984 issue ..... £
- Seven programs from the October 1984 issue ..... £
- Nine programs from the September 1984 issue ..... £
- Fourteen programs from the August 1984 issue ..... £
- Ten programs from the July 1984 issue ..... £
- Ten programs from the June 1984 issue ..... £
- Twelve programs from the May 1984 issue ..... £
- Eleven programs from the April 1984 issue ..... £
- Twelve programs from the March 1984 issue ..... £
- Nine programs from the February 1984 issue ..... £
- 26 programs from the introductory issues ..... £

I enclose the sum of £ \_\_\_\_\_

Name .....  
 Address .....

POST TO: Tape Offer, *Electron User*, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY

E  
EU/3

MIKE BIBBY and PETE BIBBY provide some helpful suggestions for would-be contributors to *Electron User*

# The 18 commandments

WHILE not wanting to put writers into a creative strait-jacket we've found that life can be made a lot easier for the magazine, our readers and the contributors themselves if our programmers stick to certain standards.

It has also occurred to us that it's no good our just knowing what we want. We have to tell you, our potential contributors.

So here are our 18 commandments.

Don't be too daunted by the list. It's mostly just common sense and good programming practice.

And we've not forgotten that the *Electron* doesn't have Mode 7! Some of our contributors spend at least part of their time on the BBC Micro, so we may as well cater for them.

Here's the list:

1. We don't use two part programs in the magazine. Games in two files may look professional but they're the kiss of death as far as the magazine is concerned. Too much can go wrong when people type them in!

2. Avoid variable names that lead to confusion such as *x* and *X*, *l* and *l* or *o*, *o* and *O*.

Also try to use lowercase variable and procedure names as it makes life easier for the reader who is trying to type it in and debug his errors. Meaningful variable names do help.

3. Tell us what the program is supposed to do and refer to it by name. After an exchange of letters, John Smith referring to "my program" can be a bit vague. If possible label everything with the program's name and your name and address, and keep your own copy of it.

If it's a game, let us know how to "cheat" so we can test out the higher levels.

4. Put more than one copy of the program on your tape, possibly recorded at different baud rates (if you can) and recording levels. And if you want the cassette back let us have an SAE with the name of the program on it.

5. Let us have a printed listing if possible. Also screen dumps or off-screen photos are much appreciated though not vital. Diagrams are always useful.

6. When you submit a listing, give us a description of the program - what it does, why you wrote it, and outline its variables and procedures. Maybe you could also give a few ideas for its improvement or expansion.

When you write out your list of procedures try to do it in the form:

```
100 PROCexample Shows how we want . . .
200 PROCdelay Holds things up . . .
```

where the line numbers refer to the lines where the procedure is defined with a DEF PROC. This helps make things clearer to our readers.

We don't expect English Literature but it does help if it makes sense.

Incidentally, it's good practice to renumber your program, starting at 10 in increments of 10 - the standard default.

7. Make sure the program actually works. Try it out on your friends for their criticism (painful though it may be).

Instructions should be complete and it helps if the spelling and grammar are correct.

8. Tell us what was the OS and Basic of the machine that the program was developed on. If possible try it out on other micros with different OSs and Basic. Always use a comma after INPUT in preference to a semi-colon as this works on both Basic I and Basic II.

9. Please do put lots of nice explanatory REMs in your programs but don't put a line of colons or asterisks or

whatever after them - they play havoc with our formatter.

Also a couple of REM statements with nothing after them at the beginning of the program gives us room to put in our messages without messing up all the line numbers.

10. Avoid having just a line number with a space and nothing else. It may make the program look neater but we get deluged with letters asking what the missing line was.

Remember, people will be

ways of doing things.

We talk about modes and Mode 7, we say that we press the Return key, not the RETURN key as you might expect.

Just look how we do it in the magazine. Our programs are Program I, Program II and so on, our diagrams Figure I, Figure II.

14. If you must use long multiple lines don't go over about 175 characters, as people always add spaces when they type them in, then complain the lines are too long.

Also if you use abbreviations such as P., they come out on the printed listing as PRINT - three more characters!

15. Always put in the right number of NEXTs. Don't just use NEXT followed by a comma as it causes a lot of confusion.

16. Please, when you send us your work, include a separate page telling us that:

- a) It is your own work.
- b) It has not been offered elsewhere.
- c) We have your permission to print it.

17. If you are sending us a Wordwise file, don't use formatting characters.

18. It's always nice if a program can have an alternative key or joystick option.

Thus endeth the 18 rules. If you follow these when you submit something to *Electron User* you will stand a much better chance of having it published. More importantly you'll be becoming a much more professional programmer.

And the better you become the more satisfying you will find it.

spending hours typing your programs into their micros. Make their life easier if you can.

11. Don't switch the error messages off or use things like ONERROR RUN - people can't debug their programs.

Always include the following error handling:

```
10 ON ERROR GOTO xxxx
.
.
.
xxxx MODE 6:PRINT " " ; ;
REPORT: PRINT " at line
"ERL
```

If you really want to trap the Escape use

```
xxxx If ERR=17 THEN
RUN ELSE . . .
```

followed by the above.

12. If you are using Mode 7 on a BBC Micro then please don't use control characters in your listings. Use CHR\$(151) or whatever instead. Control characters entered at the keyboard play havoc with our printers.

13. Double space all your written matter. Try to follow our style. We have our own



# ...and pigs might fly!

By MATTHEW  
O'DONNELL



REMEMBER those sheep jumping over the fence in the May issue of *Electron User*?

We wondered then

whether someone would send us a program that displayed low flying pigs. We might have known!

A 12-year-old reader, Matthew O'Donnell from Reading, rose to the occasion. This is the result.

Now, don't anyone mention kangaroos...

```

10 REM FLYING PIGS
20 REM (C) ELECTRON USER
30 MODE 2
40 VDU 23,244,0,0,32
    ,79,95,63,31,31
50 VDU 23,245,0,0,0,254
    ,255,255,255,255
60 VDU 23,246,0,0,0,28
    ,56,240,252,247
70 VDU 23,247,31,31,63
    ,63,115,97,64,64
80 VDU 23,248,255,255
    ,127,1,129,131,199
    ,68
90 VDU 23,249,255,254
    ,216,224,240,184,28
    ,4
100 VDU 23,250,31,31,31
    ,60,52,52,38,34
110 VDU 23,251,255,255
    ,255,1,1,1,3,2
120 VDU 23,252,255,254
    ,216,192,224,224,48
    ,16
130 VDU 23,255,255,255
    ,255,3,1,1,3,2
140 VDU 23,254,255,254
    ,216,192,224,96,32
    ,32
150 VDU 23,233,24,36,4
    ,24,16,16,0,16
160 PIG#=CHR# 244+
    CHR# 245+CHR# 246+
    CHR# 8+CHR# 8+CHR# 8+
    CHR# 10+CHR# 250+
    CHR# 255+CHR# 254
170 PIG#=CHR# 244+CHR# 245+
    CHR# 246+CHR# 8+
    CHR# 8+CHR# 8+CHR# 10+
    CHR# 247+CHR# 248+
    CHR# 249
180 SPACE#=CHR# 32+
    CHR# 32+CHR# 32+
    CHR# 8+CHR# 8+CHR# 8+
    CHR# 10+CHR# 32+
    CHR# 32+CHR# 32
190 REM DRAW SCREEN
200 COLOUR 5
210 VDU 19,0,6,0,0,0
    :GCOL 0,2
    :MOVE 0,0
    :MOVE 1280,0
    :PLOT 85,1280,410
    :MOVE 0,0
    :MOVE 0,410
    :PLOT 85,1280,410
    :VDU 23:8202:0:0:0:
220 REM START WALKING
230 FOR P=170 B
    :PRINT TAB(P-1,17)SPACE
    $;
    :PRINT TAB(P,17);
    :IF P/2=INT (P/2)
        PRINT PIG#
    :
    ELSE PRINT PIG2#
240 PROCwait(40)
250 NEXT
260 PROClookup
270 PROCflypig
280 PRINT TAB(0,5)SPACE$
290 PROCwalkon
    CHR# 228+CHR# 229
470 FOR P=17 TO 0 STEP -1
    :PRINT TAB(P+2,5)SPACE$
    :SOUND 0,-1,6,3
    :PRINT TAB(P,5)FLYPIG$
    :PROCwait(10)
    :NEXT P
480 ENDPROC
490 DEF PROCwalkon
500 PROCwait(20)
510 COLOUR 0
    :PRINT TAB(10,15)
    CHR# 233
520 COLOUR 5
530 VDU 23,246,0,0,0,28
    ,56,240,252,247
    :VDU 23,249,255,254
    ,216,224,240,184,28
    ,4
540 FOR P=8 TO 17
    :PRINT TAB(P-1,17)SPACE
    $;
    :PRINT TAB(P,17);
    :IF P/2=INT (P/2)
        PRINT PIG#
    :
    ELSE PRINT PIG2#
550 PROCwait(10)
560 NEXT
570 PRINT TAB(P-1,17)SPACE$
580 ENDPROC
300 PROCwait(100)
310 BOTO 230
320 REM DEFPROCS
330 DEF PROCwait(Wait)
340 end=TIME +Wait
350 REPEAT UNTIL TIME >=end
360 ENDPROC
370 DEF PROClookup
380 VDU 23,246,56,48,96
    ,252,246,254,252,240
390 VDU 23,249,224,192
    ,192,224,240,184,28
    ,4
400 PRINT TAB(P-1,17)PIG#
410 VDU 19,8,0,0,0,0
    :COLOUR 8
    :PRINT TAB(P+1,15)
    CHR# 233
420 ENDPROC
430 DEF PROCflypig
440 COLOUR 5
450 VDU 23,224,0,0,0,96
    ,56,27,63,239,23,225
    ,0,127,124,124,112
    ,255,255,255,23,226
    ,0,0,0,0,2,249,253
    ,254,23,227,255,127
    ,63,27,1,1,1,0,23
    ,228,255,255,255,255
    ,129,129,193,0,23
    ,229,252,252,248,240
    ,128,128,192,0
460 FLYPIG#=CHR# 224+
    CHR# 225+CHR# 226+
    CHR# 8+CHR# 8+CHR# 8+
    CHR# 10+CHR# 227+

```

This listing is included in this month's cassette tape offer. See order form on Page 61.

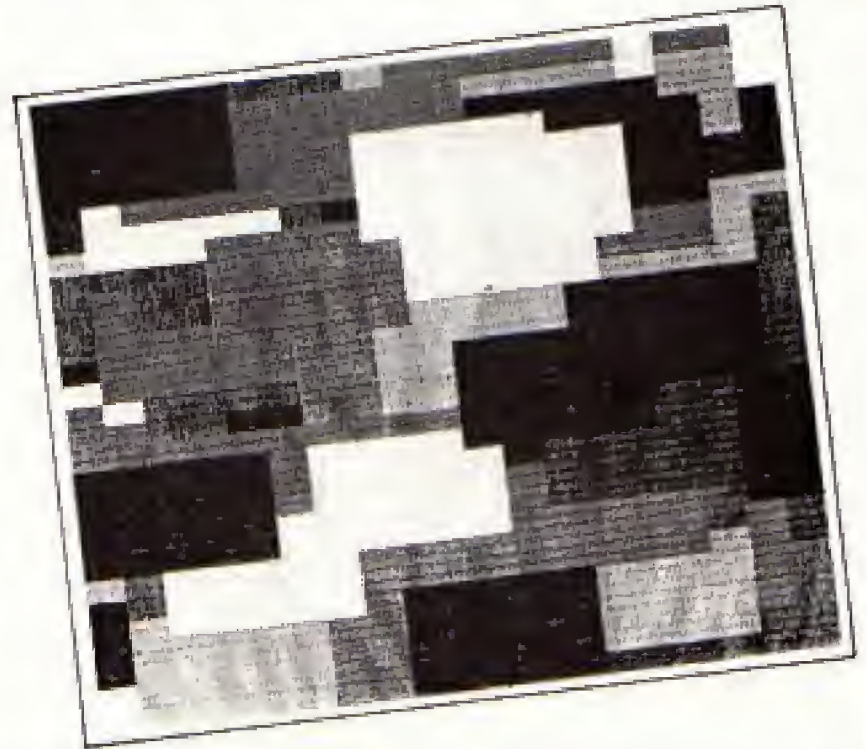
**THIS program by ROGER FROST turns your Electron into Picasso in his cubist phase.**

- 40 Sets mode.
- 50 Removes the flashing cursor, which would otherwise appear in all squares.
- 80 Stops the program if Return is pressed and restarts it when the spacebar is pressed.
- 70 Cancels any effect of long or double key strikes.
- 100 Draws a text window in a random position decided at line 90. It is randomly coloured by line 110.
- 60-120 Repeated ad infinitum.

```

10REM***PICASSO***
20REM***By R.Frost***
30REM (C) ELECTRON USER
40MODE2
50VDU23;B202;0;0;0;
60REPEAT
70*FX15,0
80IF INKEY(-74)REPEATUNT
ILGET=32
90XX=RND(15):YX=RND(25)
100VDU23,X2,Y2+4,X2+4,Y2
:100COLOUR(128+RND(7)):CLS
120UNTIL0
  
```

# Picasso, eat your heart out!



## ELECTRON OWNERS . . . . .

If you are thinking of expanding the capabilities of your Electron computer your first choice should be the ADDCOMM ROM.

ADDCOMM is now well established with BBC 'B' owners and the same chip is used with a ROM board to increase the Electron's BASIC language by forty new commands.

These new statements cover a wide range of utilities such as GRAPHICS, where eleven commands enable any shape to be drawn any size and filled with any colour combination (choice of 2 billion in Mode 2), more easily and faster than you thought possible. The TOOLKIT commands include 'find' and 'replace' statements, and a very efficient 'compact' command all of which put ADDCOMM into the top league of a recent Toolkit comparison review. The GENERAL PURPOSE statements include a sorting routine, and the ability to set up to seven windows on the screen — each with its own cursor. Split listing and jumping to a line via a label are also some of the other useful additions in this section. Eight LOGO GRAPHIC statements provide the necessary routines that when combined with BBC BASIC and ADDCOMM'S enhanced graphics give an exceptional Logo Graphics system.

ADDCOMM is available from Vine Micros, Marshborough, Nr. Sandwich, Kent, CT13 0PG. The price of £28.00 includes V.A.T. and first class post, or, if you would like more details, send a stamp for the sixteen page brochure which includes recent reviews.

## BBC/ELECTRON PROFESSIONAL SOFTWARE

*Our educational software is used in thousands of schools and homes throughout Great Britain.*

**EDUCATIONAL 1** BBC/ELECTRON Tape £8.00 Disc £10.00  
Hours of fun and learning for children aged five to nine years. Animated graphics will encourage children to enjoy counting, maths, spelling and telling the time. The tape includes six programs: MATH 1, MATH 2, CUBECOUNT, SHAPES, SPELL and CLOCK.  
*... An excellent mixture of games' ... Personal Software - Autumn 1983*

**EDUCATIONAL 2** BBC/ELECTRON Tape £8.00 Disc £10.00  
Although similar to Educational 1 this tape is more advanced and aimed at seven to twelve year olds. The tape includes MATH 1, MATH 2, AREA, MEMORY, CUBECOUNT and SPELL.

**FUN WITH NUMBERS** BBC/ELECTRON Tape £8.00 Disc £10.00  
These programs will teach and test basic counting, addition and subtraction skills for four to seven year olds. The tape includes COUNTING, ADDING, SUBTRACTION and an arcade type game called ROCKET MATHS which will exercise addition and subtraction. With sound and visual effects.  
*These are excellent programs which teachers on the project have no hesitation in recommending to other teachers' ... Computers in Classroom Project.*

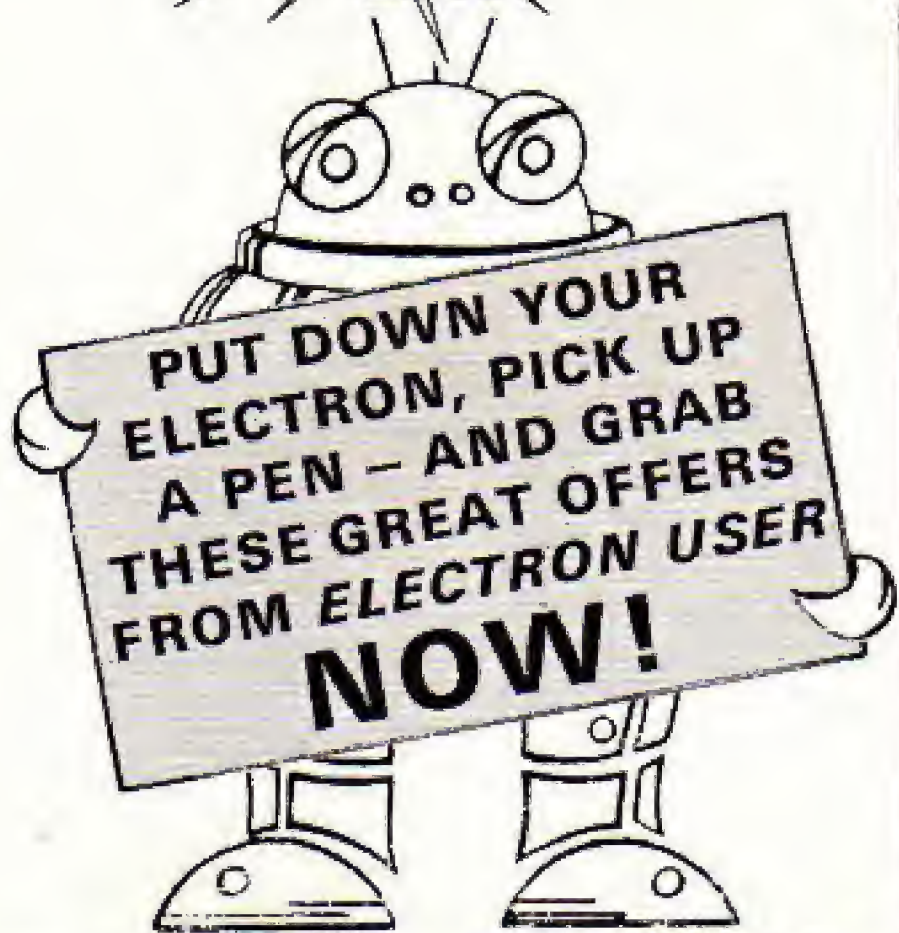
**FUN WITH WORDS** BBC/ELECTRON Tape £8.00 Disc £10.00  
Start your fun with alphabet puzzle, continue your play with VOWELS, learn the difference between THERE and THEIR, have games with SUFFIXES and reward yourself with a game of HANGMAN.  
*... 'Very good indeed' ... A&B Computing - Jan/Feb 1984*

**JIGSAW AND SLIDING PUZZLES** by P. Warner BBC/ELECTRON Tape £7.95 Disc £9.95  
There are two jigsaw and four sliding puzzles on a 3 x 3 and 4 x 4 grid. Each program starts off at an easy level to ensure initial success but gradually becomes harder. It helps children to develop spatial imagination and in solving problems. The tape includes: OBLONG, JIGSAW, HOUSE, NUMBERS, CLOWN and LETTERS.

**\*\*SPECIAL OFFER\*\***  
Buy three titles and deduct £4.00  
Add 50p p&p per order. Please state BBC or ELECTRON or  
40 or 80 track for discs

Golem Ltd, Dept E, 77 Qualitas, Bracknell, Berks RG12 4QG, Tel: 0344 50720

# electron user



## Be one of the first to get each issue

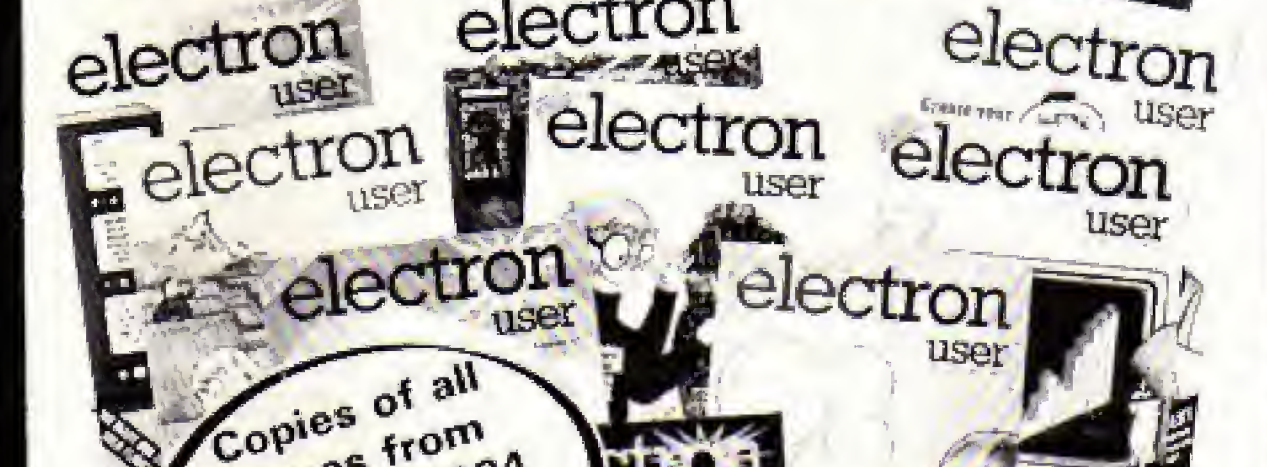
A subscription will ensure you get your own personal copy **HOT OFF THE PRESSES** month after month for the next year.

Every owner of an Electron – and everyone thinking of buying one – needs to get *Electron User* every month. It's the brightest, most authoritative yet completely independent guide to a machine that has so much potential you will never tire of reading about its remarkable capabilities.

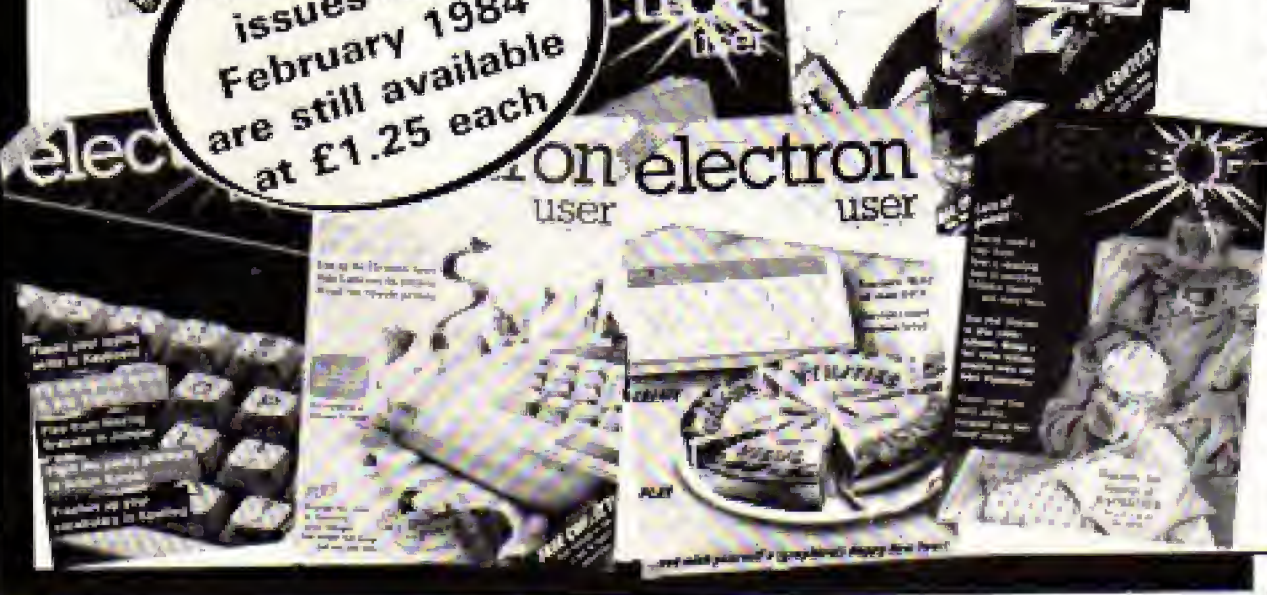
You can buy *Electron User* from your local newsagent or station bookstall. Or you can take out a 12 months subscription and have it delivered to you by post.



Complete set of our first four issues for only £2.00



Copies of all issues from February 1984 are still available at £1.25 each



### Your Electron needs protecting!

Protect your Electron with our luxury dust cover made of soft pliable water-resistant vinyl, bound with strong cotton and decorated with *Electron User* logo.

£3.95



### Keep your collection of *Electron User* complete with these handsome binders

Bound in attractive red pvc with the *Electron User* logo in gold blocking on the spine, this binder will hold 12 magazines firmly secured in place by metal rods. £3.95

# FREE

## Cassette worth £3.75 if you subscribe NOW!

If you take out a subscription to *Electron User* now you will receive completely free one of the monthly cassettes of *Electron User* listings. Choose which one you want from those illustrated below.

*This free gift is for a limited period, so subscribe now!*

### Cassette tapes of *Electron User* programs

Save typing in programs from *Electron User* by sending for these program-packed tapes.



You can also take out a subscription for the 12 monthly tapes for £40.



# ORDER FORM

All prices include postage, packing and VAT, and are valid to May 31.

Please enter number required in box £ p

<b>Electron User</b>	UK £12	<input type="checkbox"/>
<b>annual subscription</b>	EIRE £13 (IR £16)	<input type="checkbox"/>
	Overseas (Surface) £20	<input type="checkbox"/>
	Overseas (Airmail) £40	<input type="checkbox"/>

Selected free cassette \_\_\_\_\_ (month)  
Commence with \_\_\_\_\_ issue **TOTAL** \_\_\_\_\_

**Electron User introductory issues**

Complete set of 4

£2.00 UK  
£2.25 Overseas (Surface) **TOTAL** \_\_\_\_\_

**Electron User back issues**

February 1984   
March 1984   
April 1984   
May 1984   
June 1984   
July 1984   
August 1984   
September 1984   
October 1984   
November 1984   
December 1984   
January 1985   
February 1985

£1.25 UK  
£1.50 Overseas (Surface) **TOTAL** \_\_\_\_\_

Airmail prices on application **TOTAL** \_\_\_\_\_

**Electron User tapes**

£3.75 (UK & Overseas)	26 introductory programs	<input type="checkbox"/>
	Lunar Lander	February 1984 <input type="checkbox"/>
	Chicken	March 1984 <input type="checkbox"/>
	Spacehike	April 1984 <input type="checkbox"/>
	Rally Driver	May 1984 <input type="checkbox"/>
	Money Maze	June 1984 <input type="checkbox"/>
	Golf	July 1984 <input type="checkbox"/>
	Castles of Sand	August 1984 <input type="checkbox"/>
	Haunted House	Sept 1984 <input type="checkbox"/>
	Breakfree	Oct 1984 <input type="checkbox"/>
	Star Fighter	Nov 1984 <input type="checkbox"/>
	Christmas Box	Dec 1984 <input type="checkbox"/>
	Space Battle	Jan 1985 <input type="checkbox"/>
	The Kingdom of Graal	Feb 1985 <input type="checkbox"/>
	Mr Freeze	Mar 1985 <input type="checkbox"/>

**TOTAL** \_\_\_\_\_

**Cassette tape annual subscription**

£40 (UK & Overseas)  
Commence with \_\_\_\_\_ tape (state month) **TOTAL** \_\_\_\_\_

**Dust Cover**

£3.95 (UK & Overseas) **TOTAL** \_\_\_\_\_

**Binder**

£3.95 UK  
£5.00 Overseas **TOTAL** \_\_\_\_\_

Payment: please indicate method (✓) **TOTAL** \_\_\_\_\_

Access/Mastercharge/Eurocard  
 Barclaycard/Visa  
 American Express

Card NO. \_\_\_\_\_

Expiry Date \_\_\_\_\_

Cheque/PO made payable to Database Publications Ltd.

Name \_\_\_\_\_

Address \_\_\_\_\_

Signed \_\_\_\_\_

Send to: *Electron User*, Freepost, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY. (No stamp needed if posted in UK) *Please allow 28 days for delivery*

You can also order by phone

Don't forget to quote your credit card number and give your full address

Telephone:  
**061-480 0171**  
24 hours

E  
EU/3

If you already own the popular games of 'CHUCKIE EGG', 'KILLER GORILLA' or 'MONSTERS' # then our UPGRADES:

'CHUKEE' will provide 3 Speeds, 8 New Screens, Variable Extended Jump and Select Start Level (1-41).

**DESIGN YOUR OWN SCREENS!!!**

'KILLA' will provide upto 15 Levels of Play, Variable Extended Jump, Extra Lives, Practice Modes and Pause.

'MONSTAS' will provide 4 Skill Levels Variable Extended Jump, Conveyor Belt Effects, User Defined Keys, Configurable Monsters Extra Lives and Pause.

All programs available for BBC or Electron. **State Machine, £3.25** each. 2 or 3 on one cassette **£5.00** or **£7.00** by return post first class.

**BIT TWIDDLERS**

Dept, EU3 158 Church End, Harlow, Essex CM19 5PF.

\* & # are trademarks of A&F Software and Acornsoft

## 'IMAGE' V.2 - FOR THE ELECTRON/BBC

'Image' was the ultimate tape back-up copier, now it's even better

You can be completely assured that this is the best and most able program of its type available. It can deal with:

- Locked programs
- Programs of any length
- 300 and 1200 BAUD
- Files
- ?'s (Ctrl codes) in filename
- Multiple copies
- False or trick block info.
- Changing filename
- Continuous data streams
- Locking and unlocking programs

It is VERY IMPORTANT INDEED purchasers take note that 'IMAGE' is for sale strictly for making BACK-UPS of your own software for your own use, for protecting your own programs, or as an aid to putting software on disk. Any person found using the program for illegal purposes runs the risk of being prosecuted.

To receive your copy of 'Image', send a cheque or P.O. to the sum of **An Astounding £4.80** to:

**PETER DONN, Dept. (EU),  
33 Little Gaynes Lane, Upminster, Essex RM14 2JR.**

Please state BBC or Electron version.  
V1 owners can obtain V2 by sending £1.50 + V1 without cost.

# STRIPPER II

The professional tape back-up copier for the BBC/Electron, the best around!

This copier, which has been selling for over a year for use on the BBC, is updated regularly to ensure you can make security back-ups of all your own commercial software.

As Stripper II is so good we must insist on personal use only, please.

Copes admirably with locking, false addresses, control code filenames, long programs, in fact any protection you will find on your latest software releases. It even lets you put locking protection on your own programs.

New low price, just **£3.95** cheque or P.O. to receive Stripper II by return post. Educational orders welcome.

**FREE OFFER** - when Stripper III is released, owners of Stripper II can return the voucher sent to them and get the new version absolutely free!

Orders to:

**Aggressive Software, 14 Elmore Road,  
Sheffield S10 1BY**

DeMONCULE SOFTWARE PRESENTS

# BLITZ



THE EXCITING NEW VERSION OF THE CLASSIC COMPUTER/ARCADE GAME WITH FULL COLOUR HI-RES GRAPHICS AND SUPERB SOUND. WITH DIFFERENT BONUS SHEET 5, TOP 10 SCORES, & BARRAGE BALLOONS.

ONLY  
**£7.95**

Please send your cheques/PO's (payable to C. RICE) to:  
**DeMONCULE SOFTWARE C/O KENPRI,  
278 MILL LANE, ST. HELENS WA9 4ET.**

# ADVERTISERS INDEX

Aggressive Software	62
Alligata	34
Bit Twiddlers	62
Bud Computers	2
DeMoncule Software	62
Peter Donn	62
Elkan Electronics	50
Epic Software	14
First Byte	15
Golem Ltd	59
Icon Software	63
Kosmos Software	22, 36
Minic Business Services	62
National Micro Centres	26-27
Oxford Computer Services	4
Qualsoft	29
Superior Software	64
Vine Micros	59

# BRIDGE?

**CONTRACT BRIDGE PLAYER?**

... THEN YOU NEED ...

## BRIDGE-MENTOR

... AN IDEAL BRIDGE PARTNER ...

\*\*\* PERFECT YOUR CARDPLAY \*\*\*

\*\*\* IMPROVE YOUR BIDDING \*\*\*

\*\*\* ARCHIVE INTERESTING HANDS \*\*\*

\*\*\* CREATE/ANALYSE RANDOM DEALS \*\*\*

With high resolution colour-graphics and sound of your micro, "Bridge-Mentor" guides you through expertly analysed hands. A host of features including the spectacular "Autoplay" option.

**BBC (32K) AND  
ELECTRON**

Ideal for individual practice, analysis and archiving.

Superb teaching aid for clubs and professionals.

Create an infinite archive of hands that can be brought to life on the screen.

Compile special teaching material.

**£9.50 inc.**

Cheques to:- \* All cassette only.

**DEPT: EU3  
MINIC BUSINESS SERVICES,  
12 WOBURN CLOSE, BUSHEY,  
HERTFORDSHIRE, WD2 3XA.**



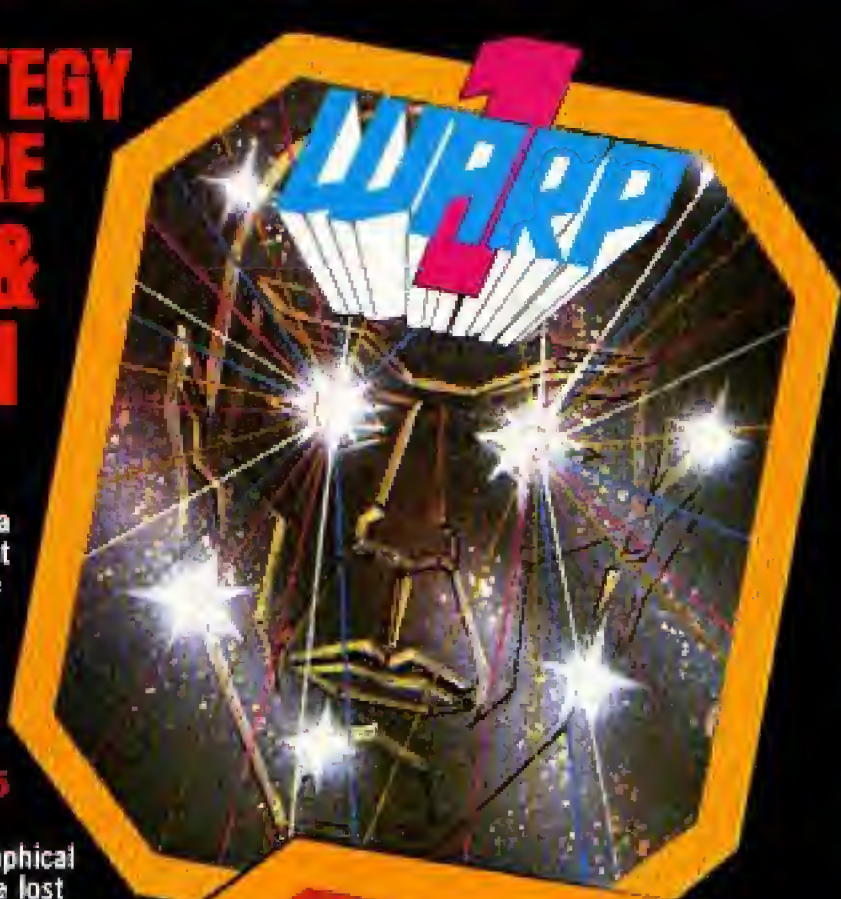


# ARCADE, STRATEGY & ADVENTURE FOR BBC & ELECTRON

**WARP 1**...command a federation starship...seek out a fellow space captain who is lost in space and boldly go where no man has gone before.

"It's like no other game I've played before... screen layout is excellent... it's different!" Electron User.  
"Stylish graphics and a splendid hyper-space display" Pop Comp Weekly

**BBC £7.95. Electron £7.95**



**ZORAKK** the conqueror...a graphical fantasy adventure in search of the lost crown of ultimate darkness.

"frustratingly addictive strategy game"  
"utterly compulsive", better than most of the strategy games available for the beeb" Personal Computer Games.

**BBC £7.95  
Electron £7.95 Dragon £7.95**



**Merry Xmas, SANTA**... arcade fun as Santa delivers presents. Jump flying snowballs... dodge falling icicles...avoid hostile snowmen and eat and drink as much as possible on the way. Excellent full colour action which can be enjoyed all the year round.

**BBC £7.95  
48K Spectrum £5.95**

**Space Station ALPHA**... a graphical space battle to save the earth from invading clyons.

"Tying colours and realistic sounds" Printout.

**BBC £7.95. Electron £7.95**



**ULTRON**...Super fast, super smooth, 'zap-em' game, blast the swarms of invading aliens to reach the mother ship...multi screen.

**BBC £7.95  
Electron £7.95**

**WONGO**...bounce along the Great Wall of China to diffuse sizzling bombs avoiding arrows, boulders and gremlins on the way. Multi screen increasing in difficulty and speed.

**BBC £7.95 Electron £7.95**



# ICON

## SOFTWARE

65 HIGH STREET, GOSFORTH,  
TYNE & WEAR, NE3 4AA.

Selected titles available from:  
John Menzies, Co-op, Harrods,  
Wildings, Granada TV and most  
good computer shops.

Dealers contact: Centresoft, Express Marketing,  
R & R comp. games,  
Drakes and Tiger Distribution.

To order direct from us:  
Send cheque or P.O. payable  
to ICON or quote your  
Access No.



Tel: (091) 2846966

# THE BEST BBC MICRO SOFTWARE

\* PRODUCED BY AN INDEPENDENT SOFTWARE HOUSE \*

\* TOP QUALITY MACHINE-CODE PROGRAMS \*

ACORN  
ELECTRON  
BBC  
MICRO

**SPECIAL OFFER!**

Order both directly from us and deduct £1 from each.



## OVERDRIVE (32K)

£7.95

A highly-addictive multi-stage 3D race game. You steer your car left and right, accelerate and decelerate as the opposing cars weave about the road. There are five different stages including night, snow, desert and riverside scenes. To qualify for the next stage, you must finish in the top twelve. Incredible graphics give the impression that you really are taking part in the race. Highly recommended, and destined to become another top-seller for Superior Software.



**BOTH TITLES ARE NOW AVAILABLE FOR THE B.B.C. MICRO AND THE ACORN ELECTRON.**



## SMASH AND GRAB (32K)

£7.95

An excellent and original arcade-style game in which you take the role of a robber aiming to snatch bags of gold from the bank. A policeman is after you ... he is able to jump at you or squat down and try to hit you with his truncheon. You must also keep clear of the flying police cones and floating dustbin lids. There are three fascinating screens of action including play streets with bouncing balls, one-way streets, conveyor belts, traffic lights and police-boxes. A novel and amusing game.

**WE PAY UP TO 20% ROYALTIES FOR HIGH QUALITY BBC MICRO AND ELECTRON PROGRAMS.**



**SUPERIOR SOFTWARE LTD.**

Dept. EU12, Regent House,  
Skinner Lane, Leeds 7  
Tel: 0532 459453



### OUR GUARANTEE

- (1) All our software is available before we advertise.
- (2) All our software is despatched within 48 hours by first-class post.
- (3) In the unlikely event that any of our software fails to load, return your cassette to us and we will immediately send a replacement.