

All Electron User
programs work on
BBC Micros with
OS 1.2 and Basic II

A Database Publication

electron user

Vol. 2 No. 3 December 1984 £1

Three terrific Christmas games

Line up the Christmas boxes
Help Santa sort the presents
Match the Yuletide pictures

Bad Program!
Bad Program!
Bad Program!

The dreaded error
message is tamed

PLUS

Programs that grow
by themselves . . .
using interrupts . . .
two-dimensional arrays
. . . and lots, lots more



FREE CONTEST!
Five chances to win
a trio of Epic's
adventures

Software Invasion in the accustomed style...

Just feast your eyes on the excitement in store for you. All at the usual quality you expect from the best name in arcade games!

GUNSMOKE
BBC Micro 32K
Acorn Electron
£7.95 Cassette
£9.95 Disk
Author: Simon Vout

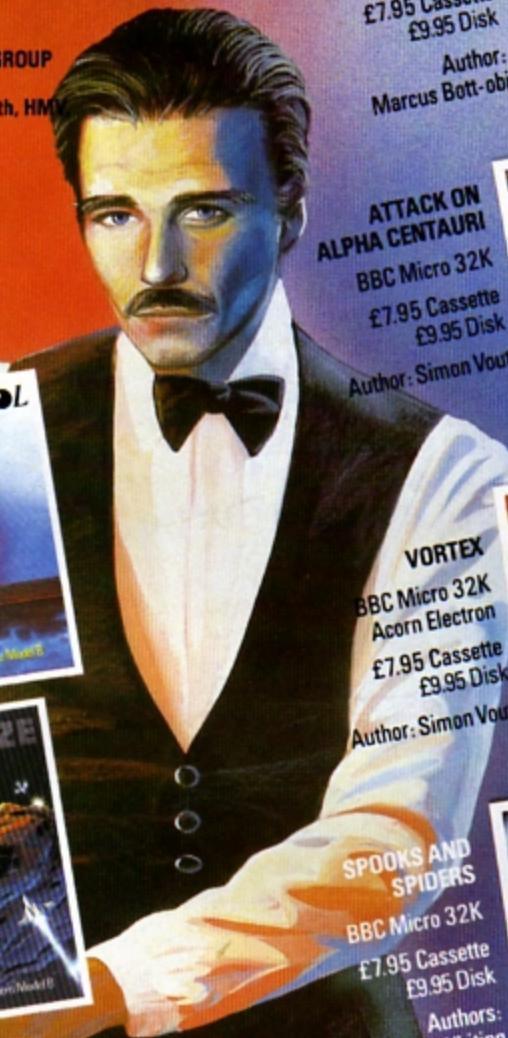


NEW RELEASES FOR THE ELECTRON!

- * **GUNSMOKE**
- * **3D BOMB ALLEY**
- SUPER POOL**
- * **COMPATIBLE WITH THE "FIRST BYTE" JOYSTICK INTERFACE!**
- * **DISTRIBUTORS, MULTIPLE STORES AND RETAILER GROUP ENQUIRIES INVITED.**
- * Software Invasion Games are available from WH Smith, HMV, all SPECTRUM shops and over 1,000 retail outlets!

FIRST BYTE
Joystick Compatible

NEW RELEASES!



ATTACK ON ALPHA CENTAURI
BBC Micro 32K
£7.95 Cassette
£9.95 Disk
Author: Simon Vout



VORTEX
BBC Micro 32K
Acorn Electron
£7.95 Cassette
£9.95 Disk
Author: Simon Vout



SPOOKS AND SPIDERS
BBC Micro 32K
£7.95 Cassette
£9.95 Disk
Authors: Mr. & Mrs. S. Whiting



3D BOMB ALLEY
BBC Micro 32K
Acorn Electron
£7.95 Cassette
£9.95 Disk
Author: Simon Vout



SUPER POOL
BBC Micro 32K
Acorn Electron
£7.95 Cassette
£9.95 Disk
Author: Dr. Robin J. Leatherbarrow

STAR MAZE
BBC Micro 32K
£7.95 Cassette
£9.95 Disk
Author: K. M. Williams

SUPER POOL
A superb Pool game with a difference! Features include time restricted shots, variable cue strength, superb real time graphics, sound and optional joystick control.

Level 1 Pot balls in any order.
Level 2 Pot balls in correct order.
Level 3 Pot and hit balls in correct order.

STAR MAZE
Travel through the Star Maze to discover rare jewels and transport them back to the mother ship. On the way, your mission is threatened by asteroids, bird hatching Eggs which lay homing mines, rotating alien satellites and alien fighters. Most deadly of all are the maze walls which will vapourise your ship on contact. A very unusual game!

Do you write your own programmes? If you have an unusual programme which can meet our standards you could be earning a substantial sum every week! Why not take advantage of our sought after reputation. Write now!

HOW TO ORDER

You may purchase any of the games above direct. All you have to do is write your name and address on a piece of paper, item(s) required, enclosing your cheque/P.O. made payable to SOFTWARE INVASION.

Please allow 7 to 14 days for delivery. (Orders are normally despatched within 48 hours.)
Overseas orders: Please add 0.75p per game.

News

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

5



Beginners

More DIM views, this time of two-dimensional arrays.

8

Notebook

A simple program, simply explained.

12



Show time

Your invitation to what promises to be our biggest show ever.

13



Christmas Snap

Our seasonal version of the traditional game.

16

Autodata

How to make your programs grow and grow.

18



Mouser

Coordinates and cats combine for education and for fun.

48



Christmas Carol

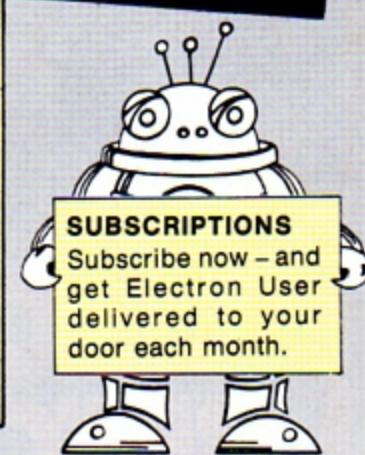
Christmas greetings combine with interrupt driven carols.

51

Micro Messages

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

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 8383 (Editorial) 061-456 8500 (Advertising)
Subscriptions: 061-480 0171 Telex: 667664 SHARET G. Prestel: 614568383.

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

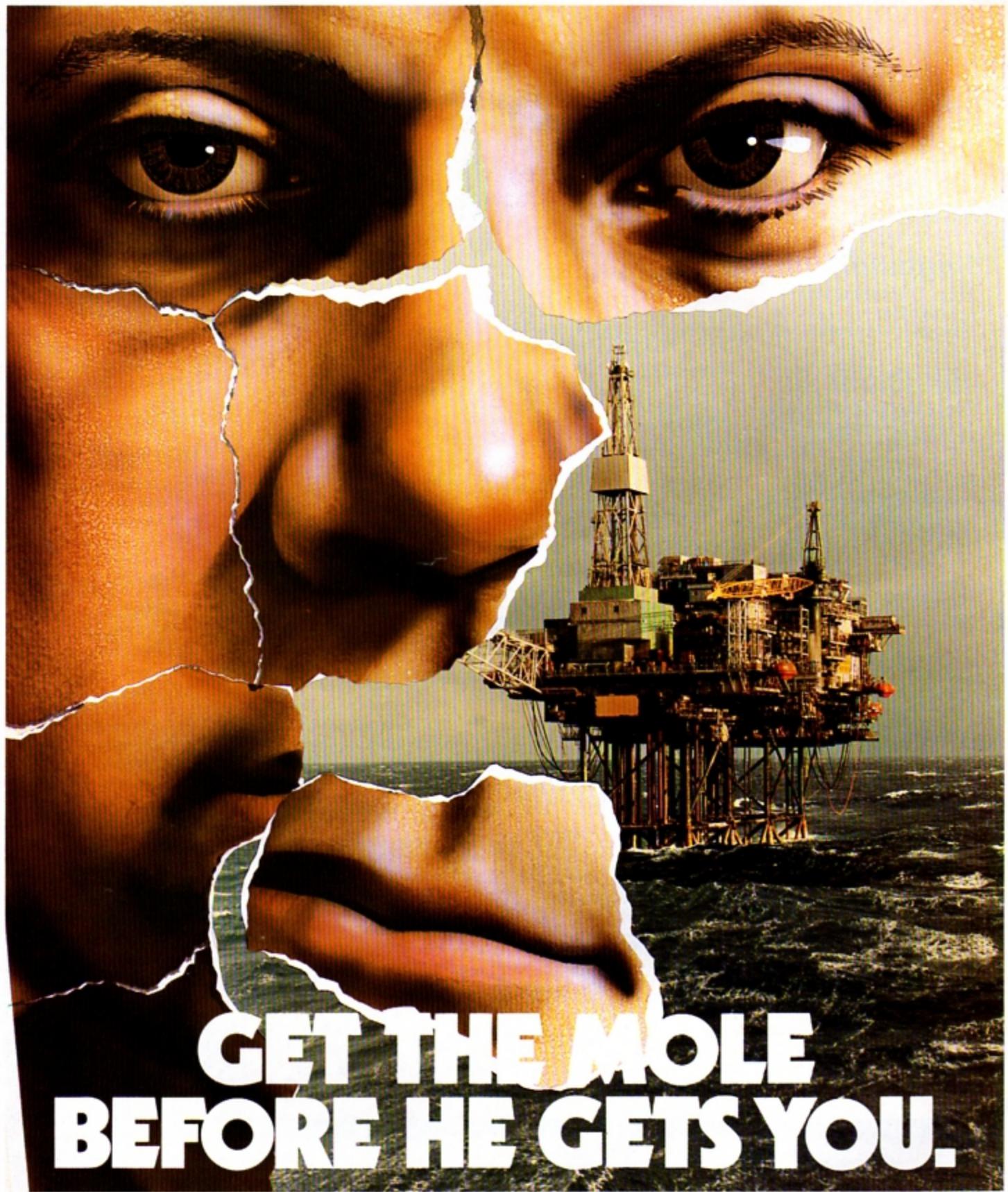
Electron User is an independent publication. Acorn Computers Ltd, manufacturers of the Electron, are not responsible for any of the articles in this issue or for any of the opinions expressed.

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 (IR £16)
£20 Europe
£20 Rest of world (surface)
£40 Rest of world (airmail)

© 1984 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.



**GET THE MOLE
BEFORE HE GETS YOU.**

Available for B.B.C. Model "B",
Electron, Spectrum and Commodore 64.
Espionage is available from all leading
High Street Stores and quality Computer
Games Specialists.

price £8.95

Retailers contact Mr. Len Fisher on
our telephone hot-line (0642) 227223.

espionage

MODULAR RESOURCES LIMITED



A GAME OF INTRIGUE & SKILL IN THE OIL BUSINESS.

electron NEWS

Computer English as she is spoke..

A NEW O-level English course from LCL for the Electron features a talking computer facility.

Micro English consists of 24 programs on



disc or cassette that can be used as a self-tuition course or for revision.

A special feature is an accompanying audio cassette, controlled by the Electron and synchronised to 1/100th of a second, enabling the micro to talk to the student.

Adhering closely to the O-level exam the course emphasises the rules of English and their application in practice, with nearly 600 questions.

The program costs £24.50 and consists of either two discs plus audio cassette, or program and audio cassettes.

Electron No 3 in top micro list - Acorn

A DRAMATIC claim that the Electron is now number three in the list of best selling home computers has come from Acorn.

The company has also forecast that the machine will enjoy bumper Christmas sales, doubling its market share from 7 to 14 per cent.

In all, Acorn predicts that between 150,000

and 200,000 Electrons will be sold over the festive season.

"This is remarkable in view of the fact we entered the market in a slack sales period and faced a tough battle to establish ourselves", a company spokesman told *Electron User*.

The announcement that the machine is now at number three caused a few eyebrows to be raised within the industry.

However a survey of the High Street retail giants brought mixed reaction to the claim.

"As far as we are concerned it is basically true", said Martin Cresswell of W.H. Smiths. "However one should appreciate that Sinclair products fill the first two positions - and their sales are way out in front of the Electron.

"But of the pack that is following the Spectrum, the Electron is in the lead followed very closely by the Commodore 64".

Over at Currys, merchandise director Richard Ford adopted a

"wait and see" attitude when interviewed.

"I'll be able to answer the question as to whether or not the Electron is number three come Boxing Day. But I don't think the company's claim may be too way out.

"But one thing you can say for certain is that Acorn as a company will be the number three supplying company by Christmas".

It was left to David Gilbert, Dixons' marketing manager, to pour cold water on the claim.

"According to our sales the Electron is probably about number five or six at present", he said, "for Sinclair, Amstrad, Commodore 64 and the BBC Micro are all ahead of it.

"But it is being heavily promoted leading up to Christmas, and this may substantially increase the sales figures for the machine.

"It will be interesting to see how the Electron does in fact shape up to some of the other deals being offered on the High Street".

Record show

THE Electron and BBC Micro User show in December is on target to smash all records.

Advance ticket sales have never been heavier and almost 150 stands are booked inside the 20,000 square feet of London's New Horticultural Hall.

With a host of hardware and software firms competing for attention, prices are likely to be keen.

The show will feature a number of specialist

stands to give Electron and BBC Micro fans the most up to date information about their micros.

And there'll be a staggering range of software, books, add-ons, robots, gadgets - and much more.

Also on hand will be a team of experts to brief visitors about the exciting new applications opening up for micro buffs - and to help out with advice on any problems.

Boredom breeds a winner



POPULAR new Electron adventure game *The Magic Sword* owes its existence to an eight-year-old boy's dislike of text-dominated programs.

Richard Hollis, of Frome, Somerset, was keen to use the computer his family bought last year but found most

of the programs rather dull.

So his mother Kristin and brother Martin, aged 12, decided to write a program Richard would enjoy.

That first attempt at program writing was successful and led to games that were eventually published in lead-

ing computer magazines, including *Electron User*.

The Magic Sword is the mother-and-son team's most ambitious project to date — an adventure game for five to nine-year-olds that is marketed on cassette by Database Publications at £8.95. The pro-

gram allows youngsters to explore a fairy tale world peopled by familiar story book characters and sprinkled with magic.

A special bonus is a free 48 page full colour book that recounts all the events leading up to the start of the adventure.

Kristin Hollis wrote the book with help from Martin, who drew the illustrations. Martin devised the game program based on his mother's design and graphics.

Now the pair are hoping to have more games published for Electron users.

After the bomb fell

THE scene is one of total devastation. People are desperately hunting for food and medical supplies while trying to dodge rampaging mutants and the odd flood.

And it's all happening

in Accrington.

"In fact, it's what any visitor might see if he visited Accrington on a Saturday evening", says computer programmer Duncan Evans.

"However on this

occasion things are perhaps a little bit worse because it has just been under nuclear attack".

The story line is to be found in a new strategy game for the Electron produced by Vampyre Software of Leeds. Written by Duncan Evans, it is called "Red Sky Over Accrington".

Mind you Duncan and his partner Mark Ulyatt readily admit they have never even been to Accrington.

"It's just one of those names that lends itself to things like this", says Mark.

The Electron cassette version of the game is now available at £6.90.

Colour plotter for under £200

DATAFAX, distributor for Sakata Shokai, is bringing out a colour plotter printer this month with A4 paper handling capability for under £200.

The Sakata SCP-800 is the first new product

to be launched here since the Japanese firm appointed Datafax.

The Electron-compatible machine also has a 210mm paper roll option and graphics and listings versatility.

Enter Plus 3 drive

ACORN was unveiling its Plus 3 self contained disc interface and 3½in single-sided disc drive for the Electron at the Compec show in November.

Also being introduced were word processing package View and



Making maths fun

EDUCATIONAL software house Applied Systems Knowledge has launched the first in a projected series of own-label learning programs for the Electron.

Number Painter is a mental arithmetic program for children aged five to 14, aimed at the home education market.

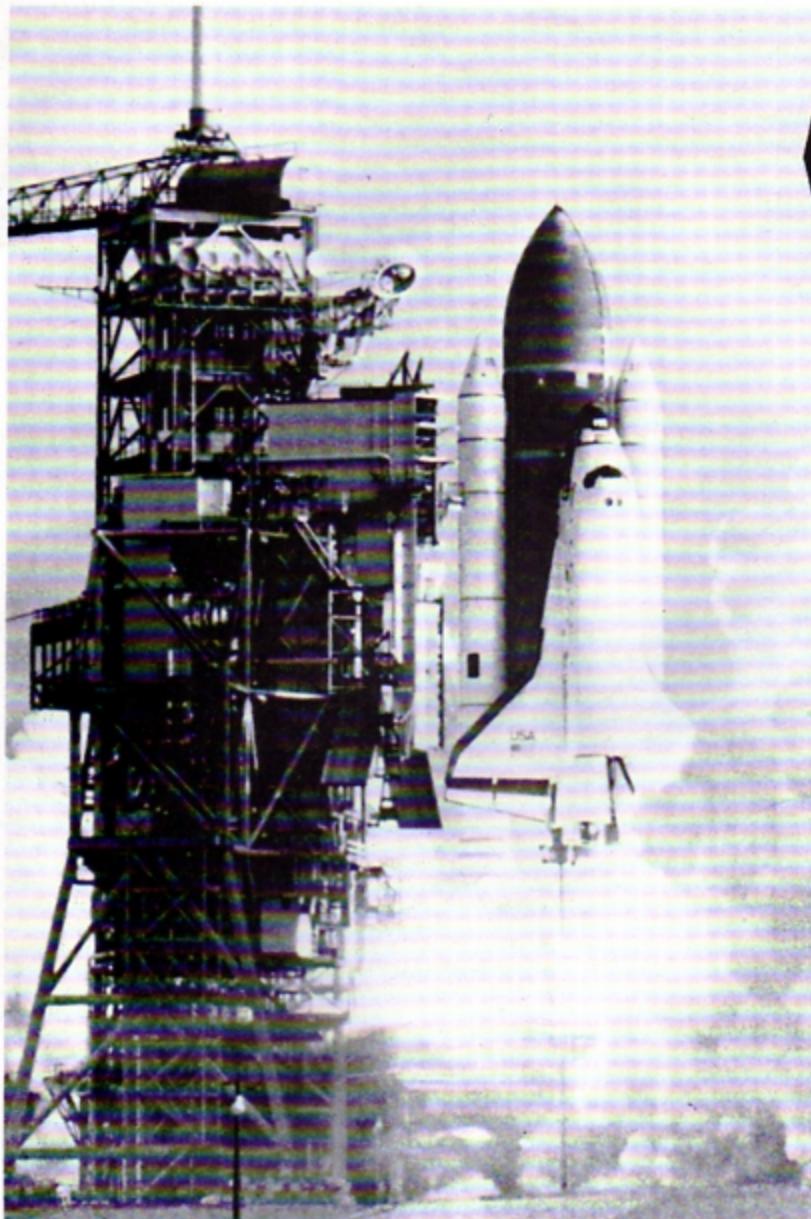
Costing £8.95 it is an arcade style game designed to improve mental arithmetic ability in problems involving addition, subtraction, multiplication and division.

Players are challenged to make a given number with a self-imposed time limit using only the numbers shown on the screen.

The numbers are collected by Mr Painter who must be manoeuvred up and down ladders and prevented from falling off. Four different speeds cope with different dexterity levels — Mr Plod, Mr Walker, Mr Swift and Mr Speedy.

spreadsheet program Viewsheet, formerly only available to BBC Micro owners.

An Acorn spokesman said prices for the new products had not been finalised, but View and Viewsheet would cost in the region of £50 each.



Shuttle trip is the prize

SECONDARY school children throughout the country now have the chance to win a five-day trip to America with an opportunity to witness a scheduled shuttle launch from the J.F Kennedy Space Centre in Florida.

The offer is the major prize in the first ever national computer competition for schools – called 'The Cub British Schools Computer Challenge' – sponsored by monitor manufacturers, Microvitec.

Apart from viewing the shuttle launch the rest of the five-day itinerary for the winning team of three and their teacher will include a visit to the futuristic Epcot Centre in Orlando.

To get to Cape Canaveral contestants will have to successfully answer a series of computer questions to take them up to the

quarter final stage.

From there they will have to shine in a number of computer tasks still to be finalised.

Support for the company's sponsorship has been expressed by local government minister Kenneth Baker.

"A challenge for schools of this kind will undoubtedly help build upon the considerable enthusiasm for the use of computers in education which has developed over the last few years", said Mr Baker.

The competition is open to teams of three contestants with a maximum upper age limit of 16.

Entry forms are to be distributed shortly for the start of the preliminary rounds in November.

Negotiations are at present being held for the televising of the final stages in April or May next year.

Stargazing

MASTERMINDS and stargazers are among the people Mirrorsoft is catering for with its five new programs

Electron users are now being offered Astronomy, developed in conjunction with the London Planetarium, Personality Profile, adapted from the best

selling book by psychologist Professor Hans Eysenck, a Weight Control program introduced by Professor Justin Joffe, a Psychic Ability test developed by Hans Eysenck and Carl Serjent and a Mastermind Quiz and Editor based on the BBC series.

Birdwatching

A GAME for Electron users produced in conjunction with the Royal Society for the Protection of Birds has been named Microdealer UK Educational Program of the Year.

Osprey, priced £9.95 from Bourne Educational Software, encourages interest in bird watching and wildlife preservation through a game involving protecting osprey nests from poachers.

● See review on Page 29.

Have case, can travel

NOW you can take your Electron anywhere... Jenart Design of Bishops Nympton, South Molton, Devon, has launched a carrying case for the machine.

The company, which specialises in computer cases and dust covers, is the brainchild of development engineer Bob Artless.

He set it up shortly after seeing his son struggling off to school with his home computer tucked precariously under his arm.

"I had visions of



him dropping it and that would have been £200 down the drain", recalls Bob. "So I designed a case for him and it all started from there".

The Electron case costs £10.99, which includes VAT and postage. It can only be ordered direct from Jenart Design.

Making a million

ELECTRON users can now try their hands at running a software company thanks to Millionaire, just launched by Incentive Software.

Versions are available for the Electron and BBC Micro.

It includes graphics of your house which increases in size as your profits grow.

Players start with £500 to market a program. By careful marketing and maybe the odd dodgy deal with Honest Harry you can move from humble beginnings to a millionaire's estate.

But beware – such a deal could put you on the wrong side of the law.

LAST month we took a look at one way our Electron can handle lists of numbers and names.

We saw that we could use a line like:

```
10 DIM scores(20)
```

to set up 21 variables all with the same name except for the different numbers in the brackets following it.

These variables were called elements in an array and the numbers in the brackets were called subscripts.

The DIM command in the line above would set up array with variables *scores(0)*, *scores(1)*, *scores(2)* and so on up until *scores(20)*. Each of these variables would initially have the value zero.

We learnt that we could also dimension arrays of string variables, *DIM name\$(10)* setting up an eleven element array starting at *name\$(0)* and carrying on until *name\$(10)*.

Initially these are set to the null string – that is, a string that doesn't contain anything.

Finally we saw how we could combine these arrays and FOR ... NEXT loops to provide some very useful ways of handling lists. Using variables as subscripts we could print out every other name or mark or display a list in reverse order.

Last month's final program, this month's Program I,

```
10 REM PROGRAM I
20 REM OLD PROGRAM VIII
30 DIM name$(3), mark(3)
40 FOR topofclass=1 TO 3
50 PRINT "Enter name of
number ";topofclass
60 INPUT name$(topofclas
s)
70 PRINT "Enter ";name$(
topofclass);"'s mark."
80 INPUT mark(topofclass
)
90 NEXT topofclass
100 INPUT "Enter number o
f position " position
110 PRINT name$(position)
;" got ";mark(position);" m
arks."
```

Two-dimensional arrays – gateway to the database

showed how we could set up two arrays in parallel, *name\$(3)* and *mark(3)*.

The FOR ... NEXT loop just sets up the array. The real work is done by lines 100 and 110.

Line 100 asks you to give a value to the variable *position*. The next line uses this variable to print out elements *name\$(position)* and *mark(position)*.

You'll notice from the above that we only used one number to get two pieces of information.

If we had dimensioned another array, such as *age(3)*, we could have had the program printing out the name, age and mark of the child in whichever position we wanted.

We could have had a fourth or fifth array set up in parallel if we wished, to hold even more information.

These parallel arrays, lists of values and information in an ordered sequence are a very simple form of what is known as a database.

They are a way of collecting information together in an ordered manner that allows us to manipulate – or pick and choose – the items we want, using a key or pointer.

In the very simple database of Program I we used one pointer *position* to give us two pieces of information held in the arrays *name\$(position)* and *mark(position)*.

Now let's turn our attention

to the situation shown in Figure 1. Here we have 16 desks in a classroom. Each desk is numbered and the name of the child sitting at that desk is shown. Also shown is the mark the child got in the spelling test.

From what's already been covered, it should be fairly obvious that we can use arrays to hold this information. Take a look at Program II:

```
10 REM PROGRAM II
20 DIM name$(16)
30 FOR child=1 TO 16
40 PRINT "Name of child
at desk ";child
50 INPUT name$(child)
60 NEXT child
70 PRINT "You've now set
up an ordered list ""of n
ames in the array name$()."
```

Here the string variable *name\$(16)*, dimensioned in line 20, is used to hold the name of each child. The array uses the desk number as the pointer.

When you've run the program, if you want to know the name of the child sitting at desk 11, just enter the direct command:

```
PRINT name$(11)
```

and the answer should be REG. Similarly:

```
PRINT name$(15)
```

will give IVY.

Using techniques we learnt

last month, we could have the Electron print out the names of each child at each desk in order or reverse order, or even every other child.

We could also use an array to store all the children's marks, as shown in Program III:

```
10 REM PROGRAM III
20 DIM mark(16)
30 FOR child=1 TO 16
40 PRINT "Mark of child
at desk ";child
50 INPUT mark(child)
60 NEXT child
70 PRINT "You've now set
up an ordered list ""of m
arks in the array mark()!"
```

Here the array *mark(16)* holds the results of the spelling test. If you want to know the mark Eileen got, just find her desk number and tell the Electron to:

```
PRINT mark(11)
```

and you should get the result 15.

All right, you've run Program II and then Program III and now we have two ordered lists. Let's use them to tell us the name and mark of the child in desk 3.

```
PRINT mark(3)
```

should give you the answer 12 but, alas:

```
PRINT name$(3)
```

column 1	column 2	column 3	column 4	
row 1	1 TOM 10	2 DICK 9	3 HARRY 12	4 FRED 7
row 2	5 SUE 14	6 DOT 7	7 LIZ 9	8 JO 11
row 3	9 JIM 18	10 BILL 16	11 REG 12	12 JOHN 8
row 4	13 SALLY 18	14 JILL 14	15 IVY 11	16 EILEEN 15

Figure I: Beginners class

just gives you the error message "Array". An error message means that something has gone wrong.

What's happened is that your first array `name$(1)` was overwritten when you entered and ran Program III. The DIM statement of line 20 told it to set aside some memory space for a list of numeric variables and this it did.

As you hadn't told the Electron that you wanted to keep the string array `name$(1)`, it simply used that same bit of memory space for the new list. Micros can be very stupid at times.

Never mind, Program IV

```

10 REM PROGRAM IV
20 DIM name$(16), mark(1
6)
30 FOR desk=1 TO 16
40 PRINT "Enter name of
child at desk ";desk
50 INPUT name$(desk)
60 PRINT "Enter ";name$(desk);"'s mark."
70 INPUT mark(desk)
80 NEXT desk
90 PRINT "You have finis
hed entering data."
100 FOR loop = 1 TO 5
110 INPUT "Enter number o
f position " position
120 PRINT name$(position)
;" got ";mark(position);" m
arks."
130 NEXT loop

```

will allow you to enter all the information in Figure I and it also lets you interrogate the database five times.

All this means is that the second FOR ... NEXT loop in the program allows you to use the desk number as a pointer to tell you the name and mark of five children. The program is very similar to Program I so making use of the Copy key should save you a lot of typing.

So now we have the information displayed visually in Figure I tucked away inside our micro in the form of two arrays. As you've seen, we can do a lot with such information.

We could add all the marks together and find the average, or we could find the average of the first five desks or the last five.

Try it and see, by varying the last lines of Program IV. And don't drive yourself mad typing in 16 names each time. Change line 30 to:

30 FOR desk= 1 TO 4

and just deal with the first row while you get the hang of things.

To sum up the above, our use of arrays has structured the data of Figure I in two lists that we can manipulate or use.

But what if we wanted to calculate the average mark of each row and each column of desks in turn? We could do it using the arrays we have now but it wouldn't be easy.

Or again, what if we wanted

the names and marks for the kids in the bottom left corner or the top right? Again, we could do it but it wouldn't be straightforward.

Having the arrays ordered one after the other might not be the best way of ordering things.

Wouldn't it be nice if we could store the information in Figure I in such a way that we could refer to each desk not by one number but by the row and the column of the desk? Then we could find out about Eileen by referring to row 4, column 4.

Instead of our lists being in an ordered sequence, they could be in a sort of grid, mimicking the classroom itself.

As you might guess, there is a way of doing this and it involves our old friend the DIM statement.

We use it to dimension what is known as a two-dimensional array, an array which has two subscripts. Don't worry if you don't follow this, read on and all will be explained.

Let's create a two-

dimensional array. We do this with a line like:

20 DIM desk(4,4)

in a program.

You'll notice that it's very much like the previous DIMs we've dealt with but that there are now two numbers in the brackets, separated by commas. These two numbers are what make it a two-dimensional array.

What happens when the Electron executes line 20 is that it sets up 25 variables. All are shown in Figure II.

As you can see, the variables range from `desk(0,0)` and `desk(0,1)` all the way to `desk(4,3)` and `desk(4,4)`. The DIM statement has, as before, set up a series of variables with the same name stem but with varying subscripts.

The difference is that in a two-dimensional array we have two subscripts in the brackets of an array element.

If you look at Figure II you'll see that we've set up 25 variables and a closer look will

desk(0,0)	desk(1,0)	desk(2,0)	desk(3,0)	desk(4,0)
desk(0,1)	desk(1,1)	desk(2,1)	desk(3,1)	desk(4,1)
desk(0,2)	desk(1,2)	desk(2,2)	desk(3,2)	desk(4,2)
desk(0,3)	desk(1,3)	desk(2,3)	desk(3,3)	desk(4,3)
desk(0,4)	desk(1,4)	desk(2,4)	desk(3,4)	desk(4,4)

Figure II: A two-dimensional array

From Page 9

show that each variable has its own unique pair of subscripts.

You might also notice that the variables seem to fall naturally into ordered series.

One example is:

```
desk(2,0)  
desk(2,1)  
desk(2,2)  
desk(2,3)  
desk(2,4)
```

Looking at Figure II, they all seem to fall into one column. Notice that the first subscript in each variable, 2, stays the same, while the second subscript goes from 0 to 4.

Anyone who thinks of nested FOR ... NEXT loops here goes to the top of the class.

Again, looking at Figure II you might pick out a row formed by the variables:

```
desk(0,3)  
desk(1,3)  
desk(2,3)  
desk(3,3)  
desk(4,3)
```

By now you might be seeing why it's called a two-dimensional array. If you were so inclined, you could name any of the elements of the array as *desk(x,y)* where *x* and *y* are variables.

When *x* is 2 and *y* is three, then the element we are naming is *desk(2,3)*.

Of course, if we called the subscript variables *column* and *row*, any element of the array could be referred to as *desk(column, row)*. Or, equally as well, *desk(row, column)*.

Taking the latter case, if, in the course of a program, *row* is 4 and *column* is 2, then the element *desk(row, column)* is *desk(4,2)*.

Looking at Figure II again, you'll see that if you ignore all the elements that have a zero in them – effectively, the first row and column – what's left is very much like a map of the classroom in Figure I.

In fact we can use our two-dimensional array to hold the desk numbers, the array mimicking the classroom.

Of course, we've done this before in the one-dimensional

arrays we learnt about at first.

The difference is that this time we can get at the information row by row, or column by column or, even any combination of the two.

Program V shows this in action.

Line 20 dimensions a two-dimensional array while the nested FOR ... NEXT loops work their way around the class. If you can't follow that, work it out on a piece of paper.

While *column* is 1, *row* goes from 1 to 4 with the inputs being stored in the variables *desk(1,1)*, to *desk(4,1)*.

Once the program has built up the array it enters another series of FOR ... NEXT loops. These print out the values of *desk(row, column)*, but they do it selectively.

The variable *row* only has values 1 and 3 – look at the STEP – while *column* cycles from 1 to 4 for each of these two values.

The result is that only the desk numbers for the boys are printed out. Can you alter the program so that it prints out the girl's desk numbers?

As you can see from the above, we've used an array with two subscripts to hold information. The fact that it has two subscripts means that we can do more things with it than with an ordinary array.

We can use FOR ... NEXT loops to deal with whole rows

```
10 REM PROGRAM V  
20 DIM desk(4,4)  
30 FOR column=1 TO 4  
40 FOR row=1 TO 4  
50 PRINT "Enter the number of the desk in row ";row  
";", column ";column  
60 INPUT desk(row,column)  
  
70 NEXT row  
80 NEXT column  
90 CLS  
100 PRINT  
110 PRINT "The boy's desks are numbered:"  
120 FOR row=1 TO 3 STEP2  
130 FOR column=1 TO 4  
140 PRINT desk(row,column)  
  
150 NEXT column  
160 NEXT row
```

```
10 REM PROGRAM VI  
20 DIM name$(4,4),mark(4
```

```
,4),desk(4,4)  
30 FOR row=1 TO 4  
40 FOR column=1 TO 4  
50 PRINT "Enter the name of the child in the desk in row ";row  
";", column ";column  
60 INPUT name$(row,column)  
n!
```

```
70 PRINT "Enter "name$(row,column)"'s mark"  
80 INPUT mark(row,column)  
)
```

```
90 PRINT "Enter "name$(row,column)"'s desk number"  
100 INPUT desk(row,column)  
)
```

```
110 NEXT column  
120 NEXT row  
130 FOR delay=1 TO 200  
140 CLS  
150 PRINT
```

```
160 PRINT "THE RESULTS FROM THE MIDDLE FOUR DESKS:"  
170 FOR row=2 TO 3  
180 FOR column=2 TO 3  
190 PRINT name$(row,column)  
n" in desk number ";desk(r  
ow,column);" scored ";mark(r  
ow,column)
```

```
200 NEXT column  
210 NEXT row
```

or columns at a time.

Notice that while the elements of a two-dimensional array have two subscripts, that element only takes one value. In Program V *desk(1,1)* held only the value corresponding to desk number one.

The second subscript doesn't let us hold any more information, it just allows us to deal with it better.

You could use three two-dimensional arrays to hold the whole of the information in Figure I.

Program VI does this, setting up three two-dimensional arrays in line 20. Then come the familiar nested loops to enter all the information into the arrays.

This is much the same as the previous program, except that we're also using a string array. The interesting point comes after line 150 where we use our faithful nested

loops to pick out and print the details of the middle four desks.

If you can't follow how that's done, just make up versions of Figure II using *mark()* and *name\$()* and you'll see why the loops have the values they do.

Finally run Program VII. It sets up the database as before, storing the classroom information in two two-dimensional arrays.

It then asks you to enter a row and column number and gives you the information on the child who sits at that desk.

```
10 REM PROGRAM VII  
20 DIM name$(4,4),mark(4  
,4)
```

```
30 FOR column=1 TO 4  
40 FOR row=1 TO 4  
50 READ name$(row,column)  
,mark(row,column)  
60 NEXT row  
70 NEXT column
```

```
80 CLS  
90 PRINT  
100 INPUT "Give me a row number " row
```

```
110 INPUT "Give me a column number " column  
120 PRINT name$(row,column)  
n" scored ";mark(row,column)  
n"; marks"
```

```
130 DATA TOM,10,SUE,14,JO  
M,18,SALLY,18  
140 DATA DICK,9,DOT,7,BIL  
L,16,JILL,14  
150 DATA HARRY,12,LIZ,9,R  
EG,12,IVY,11  
160 DATA FRED,7,JO,11,JOH  
N,8,EILEEN,15
```

As you can see, setting up the database allows you to ask all sorts of questions about the class. But then you've probably got all sorts of questions about Program VII itself.

What's all this DATA and READ?

Well, the answer to that comes next month. For the time being, just look on them as ways to avoid typing in all those names and marks.

Meanwhile, just play around with a few two-dimensional arrays, giving them values and seeing what you can do with them. You'll find them very handy.

KAY-ESS

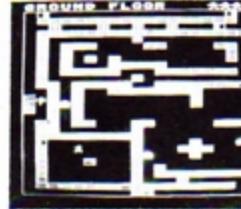
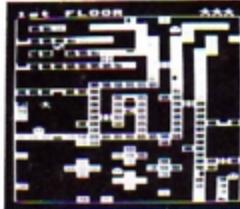
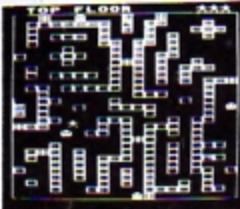
COMPUTER PRODUCTS

PROFESSIONAL
PROGRAMS FOR
THE MODEL B
AND ELECTRON

EACH TAPE ONLY £5.95 (Except H.O.H.)



HOUSE OF HORRORS (B)(E) £6.95



Turn off the lights and gather around for the most creepy game of the year. How you laughed at those superstitious fools in the village when they warned you not to go near the old house. The climb up the rocky path under the afternoon sun was swift and within an hour you had passed through the outer gates of this once great house. The dust and cobwebs hadn't bothered you as you climbed the old stairs to the towers on the top level. Did you notice how low the sun had fallen before the sounds of locks clicking reached your startled ears? How can the moon be out already and what's that moving towards you? This all action game will have you ducking and diving from the GHOSTS and ZOMBIES, and matching wits with a MUMMY, WEREWOLF, and VAMPIRE. 5 floors full of odd CORRIDORS, BROKEN FLOORBOARDS, and riddled with SECRET PASSAGES await you. Superb sound effects and graphics. Can be played using either keyboard or joysticks. Top table. Pause option.

EARLY YEARS (B)(E)

For children between 3-6 years of age. These two packages give an adult or older child a means to take a younger child through a series of simple game type tasks to enforce ideas. The emphasis is on learning through fun. Topics covered include subtraction, addition, recognition, colour, shapes, sizes, sounds/notes, co-ordination, distances, estimates, directions.

EARLY YEARS 1

- A) MICKEY THE MONKEY and his apple tree make subtraction fun.
- B) COLOUR BLOCKS bring sizes and colour into perspective.
- C) MERRY MUSIC turns the keyboard into a musical keyboard.
- D) FUNNY FACES presents a line up, which one is the suspect?
- E) FRED THE FROG needs co-ordinated help to get across the pond.

EARLY YEARS 2

- A) THE POND seems very active today.
- B) SPEED is required to keep the cake on the conveyor belt.
- C) DIRECTIONS seem to be needed by everyone in Orion village.
- D) ORDER the blocks.
- E) SID THE SPIDER needs some help to get out of the maze.

Watch out for HOUSE OF HORRORS at local dealers.

Dealer enquiries welcome.

All prices are FULLY inclusive for UK orders. Please add £1 per tape for non-UK addresses. Cheques/P.O.'s should be made payable to KAY-ESS Computer Products.

When ordering please state BBC or Electron.



Joystick
Compatible

Available for:
(E) Electron (B) BBC Model B
FREE with all orders (Except H.O.H.)
our 3 level version of
NOUGHTS AND CROSSES!!!

ALSO AVAILABLE:

STAR HAWKS (B) (E) - DESIGN (B) (E)
- HANGMAN (B) (E) - SPACE TRAFFIC
CONTROLLER (B) (E) - HORSES (B) (E)
- SPACE TANK (B)

KAY-ESS Computer Products,
11 Buttercup Close,
Romleahs Park,
Harold Wood,
Essex RM3 0XF.

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

QUAL-SOFT

£9.95

(inc. VAT and p.p.)

QUAL-SOFT Comments: About our previous advert Mr J. Carter of Barnsley 'phoned'. *'Don't tell us what people think of LEAGUE DIVISION ONE for the BBC B, what do they say about SOCCER SUPREMO for the ELECTRON?'* Copy dates being what they are, we hadn't sold any ELECTRON games when we wrote last month's advert, but now, at the time of writing we've been delivering the program for four weeks (probably eight to twelve as you read this), and, to escape accusations of selective quotes, we will give you the first paragraph of the first letter we had received about SOCCER SUPREMO, from J. Hooley of Twickenham: *'Many thanks for the fantastic game. As soon as I received it, there was no stopping until the end of the season'. We make that about six hours continuous play!* And in the first telephone conversation P. Wright of Swansea began: *'This is by far the best game I've found for the ELECTRON'.* OK Mr Carter?

"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:

Notebook

How long
is a piece
of string
handling?

THIS month's Notebook uses string handling techniques to show how the MOD function produces a series of remainders that cycle in value.

If MOD and DIV baffle you take a look at Dave Robinson's article in the September *Electron User*. Or just sit back and watch the number patterns.

27 MOD 5
3
4
0
1
2

PRESS SPACE

10-30 Surely you don't need these explaining to you!
40 This changes the text colour to yellow, palette colour 3.
50 See 10.
60, 190 Form a FOR...NEXT loop with control variable divisor. The divisor is the number that comes after the MOD. As the loop cycles this goes from 1 to 5. Sets up a string variable *string\$* and fills it (for the time being) with spaces. This will later hold a series of results from a MOD with the same divisor but different dividends. The number of spaces is determined by divisor.
70 Make up a FOR...NEXT loop which has the control variable dividend changing from 1 to 50.
80,160 This uses the function STR\$ to turn the result of dividend MOD divisor into a string. This is placed in the variable *addtostring\$*.
90 Isn't as complicated as it looks. First it joins *addtostring\$* and *string\$*. It then uses RIGHT\$ to take the first divisor letters of this new string and give it to *string\$*.
100 Just prints the top line. This varies with dividend and divisor.
110 Form yet another FOR...NEXT loop, this time with control variable *stripstring*. This splits *string\$* into its separate components and prints them underneath each other.
120, 140 A delay loop. Leave it out and see what happens.
130 Holds things up until a key is pressed.
150 Overprints the top line with spaces. Again, leave it out and see what happens.
170
180

Outer loop — Cycles 5 times changing dividend
Inner loop — Cycles 50 times changing dividend

```
10 REM REMAINDER
20 REM TOM PRATT
30 MODE 6
40 VDU 19,1,3,0,0,0
50 VDU 23,1,0,0,0,0
60 FOR divisor=1 TO 5
70 string$=STRING$(divisor," ")
80 FOR dividend=1 TO 50
90 addtostring$=STR$(dividend MOD divisor)
100 string$=RIGHT$((string$+addtostring$),divisor)
110 PRINT TAB(10,8);dividend;" MOD ";divisor
120 FOR stripstring=1 TO divisor+1
130 PRINTTAB(10,8+2*stripstring)MID$(string$,stripstring,1)
140 NEXT stripstring
150 FOR delay= 1 TO 200:NEXT delay
160 NEXT dividend
170 PRINT TAB(10,23)"PRESS SPACE":wait=GET
180 PRINT TAB(10,8)STRING$(8," ")
190 NEXT divisor
```

yellow text
sets up a string of spaces
Middle loop
holds things up
erases heading

Trevor Roberts



New Horticultural Hall,
Westminster, London SW1

Thursday, December 6,
to Sunday, December 9.

We're back in the heart of London

*... with lots and lots
of Christmas bargains!*

We confidently predict this will be the biggest Show we have ever organised – with more exhibitors than ever before and with many hundreds of really worthwhile bargains for all our visitors.

Among our exhibitors are all the leading names in the world of the BBC Micro and Electron. And they are determined to make this a show to remember, with a whole host of hardware and software bargains.

You'll be able to try and buy some fascinating new games that are destined to be the top sellers this Christmas – and well into 1985. You'll be able to inspect new hardware packages that will make your micro much more powerful and much more versatile.

And on our Technical Advice Stand you'll be able to discuss any problems you may have with experts who know the BBC Micro and Electron inside out.

*Don't miss the show where all
your questions are answered!*

This voucher is worth £1 per head

Reduced prices for School/College Groups

Entry only £1 per student if bookings are made in advance. Send your cheque (made payable to Database Publications) and SAE to:

Electron & BBC Micro User Show
68 Chester Road, Hazel Grove,
Stockport SK7 5NY
Tel: 061-456 8383

Valid for a minimum of 10 people

SAVE MONEY with our Special Travel and Hotel Offer

Visitors to the Show can obtain cut-price rail tickets from ANY station in the United Kingdom – plus special reduced prices at London hotels. Write to:

Travel Offer, P.O. Box 1, St. Albans AL1 4ED with SAE or Telephone: St Albans 34475 quoting: The Electron & BBC Micro User Show.



By handing in this voucher
at the door you save £1 off
the normal admission
price of £3 (adults) and
£2 (children).

(Valid for a maximum of 4 people)
(Only 1 voucher per person)

10am-5pm, Thursday, December 6

10am-5pm, Friday, December 7

10am-5pm, Saturday, December 8

10am-4pm, Sunday, December 9

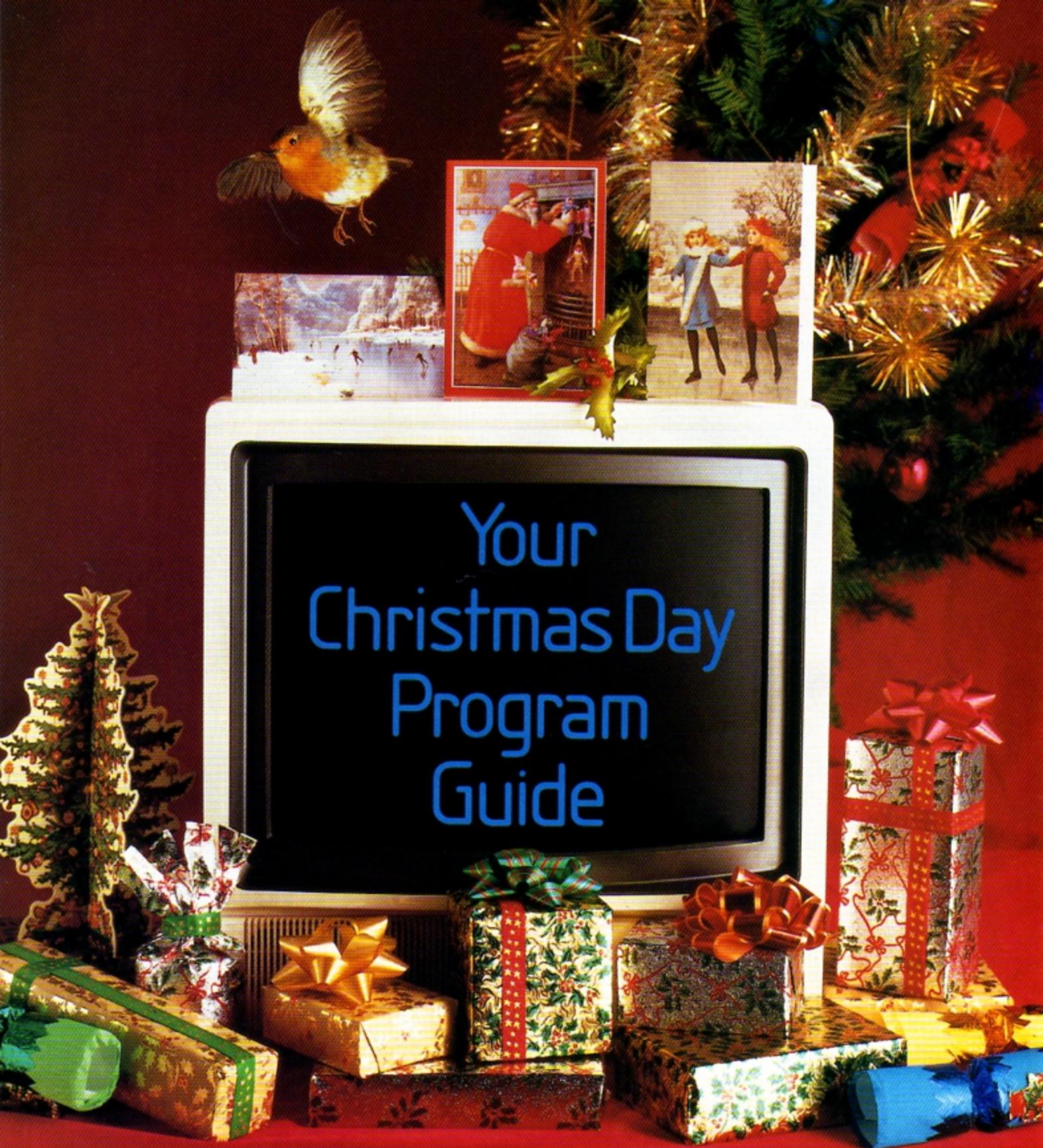
**New Horticultural Hall
Westminster, London SW1.**

Number attending:

1 2 3 4

EU/12

Organised by
The Micro User
and Electron User



Acornsoft announce tidings of great joy for both BBC Micro and Electron owners: eight brand new programs for Christmas.

Each one makes an ideal stocking filler. They're all available at your local Acornsoft stockist. For the address or to order by credit card simply ring 0933 79300.

Or you can send off for our catalogue by writing to Acornsoft, c/o Vector Marketing, London Road, Wellingborough, Northants NN8 2RL.

Alternatively, however, you could always take a chance and drop a line to Father Christmas.

ACORN SOFTWARE
Software for the BBC Micro and Electron.

9.00 Firebug

A fast moving arcade-type game in which you are a fireman, trying to rescue some oil drums and take them to the safety of a water tank. Your opponent is the firebug who runs around lighting fires which move slowly towards the drums and fire extinguishers, destroying them if contact is made.

10.00 Maze

A gripping graphics game where you enter a top secret installation with the aim of stealing secrets from a rival company. The security system, however, has many levels each consisting of a maze of corridors patrolled by armed robot guards. Complete with full colour 3-D graphics, sound effects and a high score table.

12.00 Elite

A superb 3-dimensional graphics game that's light years ahead of any other. You are a space trader who roams the universe, making your living from buying and selling cargo in your Cobra space craft. On your travels, you will encounter aggressors who are eager to put an end to your dealings. Be warned, only the fittest will survive.

1.00 Crazy Tracer

A crazy adventure in which you guide a paint roller round the edge of a maze of rectangles, while avoiding the monsters which are trying to stop you by crushing the roller. Beware – as the game progresses, so the number of monsters chasing you will increase.

3.00 Go

'Go' is a board game for two players which originated in China 3000 years ago and is now more popular than Chess in the Far East. It requires strategic insight, intuition and a strong, calculating mind. If you wish, you can also challenge the computer at differing degrees of difficulty. A yelly good game indeed.

4.00 Watch Your Weight

At last, a weight-loss program designed especially for you. With it, your computer becomes an expert wholly conversant with and sympathetic to your needs, and will help you choose an appropriate and individual weight-loss plan. The program also includes a calorie counter and a series of apposite menu suggestions to help stimulate your imagination when you just can't think what to eat.

5.00 The Complete Cocktail Maker

With everything from a Black Russian to a Betsy Ross, this program contains data sufficient to concoct a staggering 320 different cocktails.

6.00 Paul Daniels' Magic Show

Stun and amaze your friends with your astounding feats of magic. This program contains ten tricks to be performed by or with your BBC Micro/Electron. Hurry though – it's bound to disappear fast.

CHRISTMAS Snap is Phil Tayler's Electron version of the old card game for young children.

A multicoloured Christmas object, such as a tree, appears on the left of the screen and a series of similar objects appear one after the other on the right.

Your job is to tell the Electron when the colours of the objects match exactly. Only you don't say "Snap" - you press the space bar.

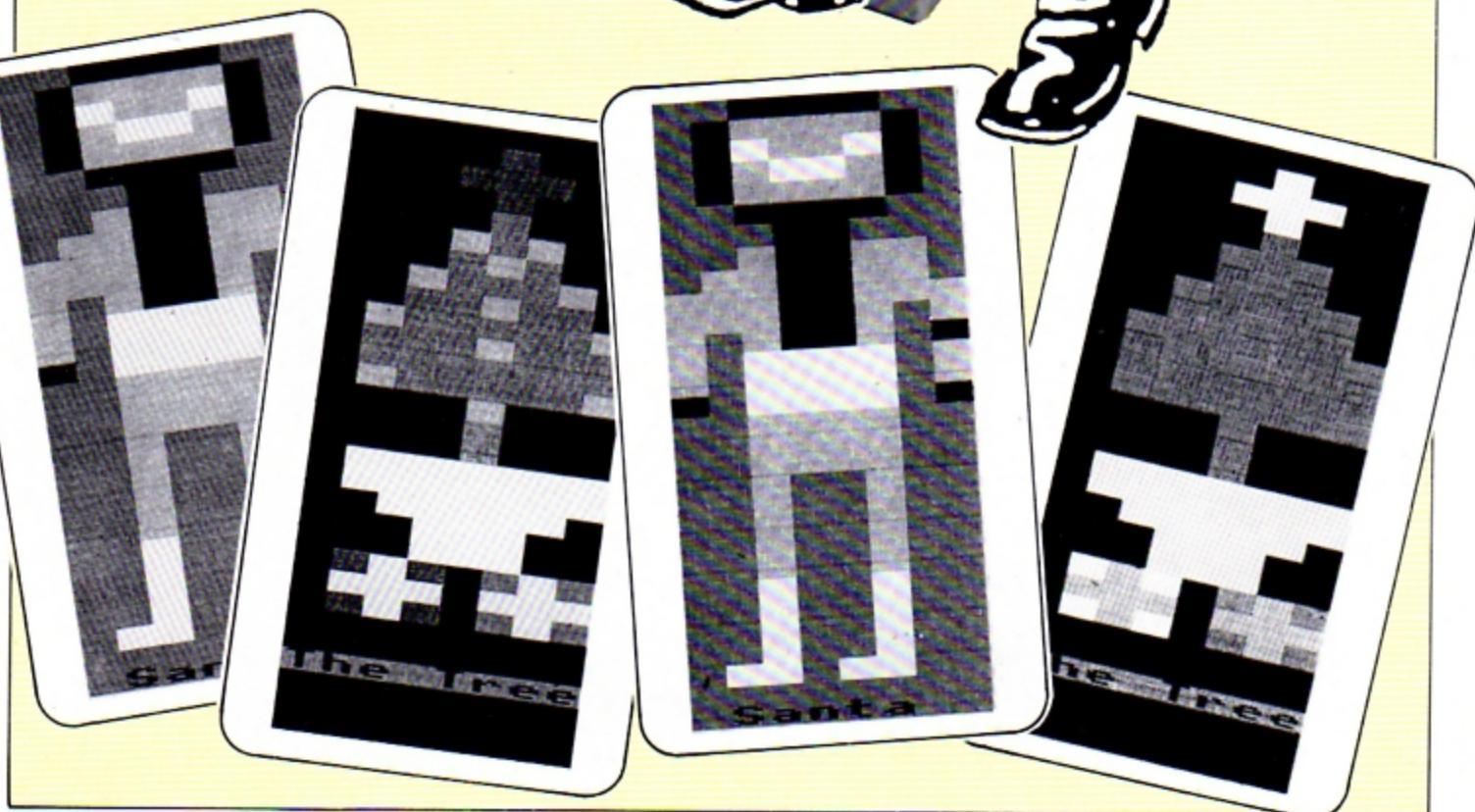
If you're wrong you'll get a short beep. If you're right... well play the game and find out.

Happy Christmas.
Happy Christmas. Snap!

CHRISTMAS SNAP!



Full listing starts on Page 55



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.

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 _____

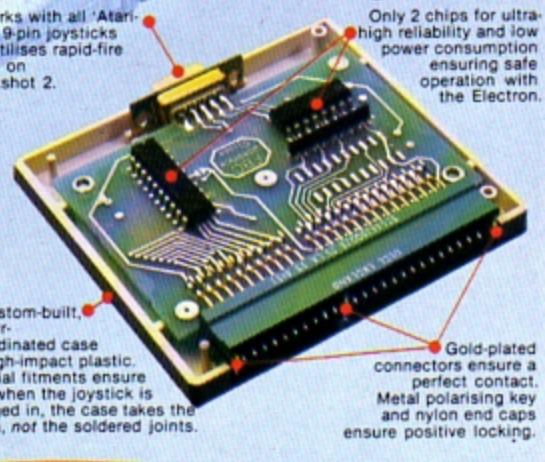
Address _____

Tel _____

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 fitments 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**

A GENUINE FIRST BYTE
ADD-ON

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

Let your programs add their own data statements

This easy-to-follow utility by JOHN WOOLLARD is a helpful aid to school programming

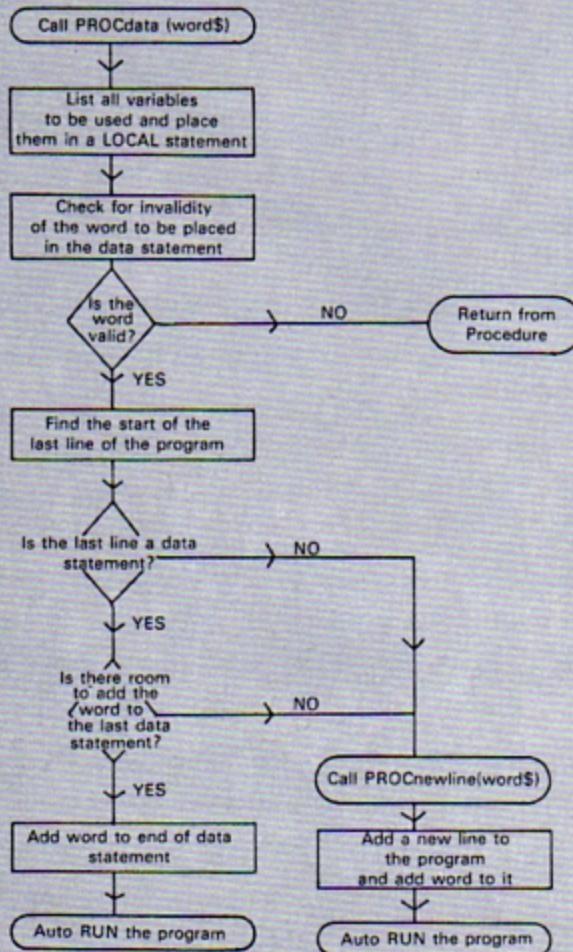


Figure 1: Flow diagram

MANY of the programs I use in my school have a series of data statements in them, containing information like word lists, vocabulary, names and questions and answers.

Frequently these need to be extended. However the users—both teachers and pupils—may not be able to stop a program and add their own data statements. So it was necessary to develop a way whereby a program could add data statements to itself.

Autodata contains two procedures which can be used to do this.

If you want a copy of Autodata and don't feel up to typing it in yourself then send off for this month's *Electron User* tape (see Page 44).

But if you wish to see how your Electron remembers programs and organises its memory then switch on and we'll begin.

Enter line 70—making sure that the character after OLD, RENUMBER, *FX210,1 and LIST is a ";"

70 *KEY10OLD:MRNUMBER:IM
IN*FX210,1:MLIST:IM

Now run it and press Break. If your screen displays:

Acorn Electron
BASIC
>

and nothing else, then go to the beginning and start again (or go to the bottom of the class and learn to read).

Line 70 programs the Break key. It tells the computer what to do after someone has pressed Break.

OLD causes the computer to re-remember the program in memory.

RENUMBER renames the program from line 10 in steps of 10 (the default values). After pressing Break check that the computer has not "failed" to renumber.

If a "failed at" message appears then that line has a GOTO, GOSUB, ON GOTO or ON GOSUB instruction which needs changing. Of course, none of my friends uses GOTOS or GOSUBs so it does not apply to them!

I find the renumber instruction useful when developing programs. If I need to insert a

lot of extra lines then pressing Break makes room for more.

*FX210,1 is for the benefit of your family! As I do most of my programming in bed and in the early hours the place is rather quiet. It does not matter how careful I am—I'm bound to accidentally press the Copy key causing an offensive beep. So *FX210,1 (which cuts off all sound output) is my salvation.

By the way, it also stops the sound output of most arcade type games so even those don't cause grievous annoyance to the rest of the world.

Ctrl + N sets the page mode on (see VDU14). This means the automatic scrolling of the screen is stopped and the computer waits until Shift is pressed, which reveals another page of printing. Pressing Escape enables the user to edit the program.

After Break has been pressed and the instructions given so far are completed the computer lists the program one screenful at a time.

When the development of a program is finished I usually change the line to:

70 *KEY10OLD:MRUN:IM

If Break is pressed, either accidentally or on purpose, the program reruns itself from the beginning. Pressing Ctrl + Break then typing OLD + Return allows a programmer access to the program.

Back to the problem in hand—the development of a routine that will allow the program to add data statements to itself.

The first step is to draw a flow diagram which shows the algorithm of the proposed program. That should contain all the facilities required in the final program and the precise order of action. See Figure 1.

All the action will be contained in a procedure called PROCdata().

However to keep the structure of that procedure simple it was necessary to call from within PROCdata() another procedure called PROCnewline().

Now the algorithm of the program has been set out it is necessary to convert this into statements in Basic.

Line 210 lists all the

variables used in the procedure. This is most important if it is going to be used in a variety of programs.

We do not want to use the variable *last%* and change its value if it is used in another part of the program.

By placing it in a LOCAL statement even if *last%* is used in the rest of the program this procedure will not change its value.

Lines 220 to 280 check for invalidity in the word to be added to the data statements. It is important that the word does not contain quotation marks.

If they were inserted in a data statement an error would occur when the statement is read. All quote marks are removed and replaced by apostrophes.

If the word is a null string then the process aborts. If the word is longer than 247 characters then it must abort because it is not possible to have a data statement of that length.

Finally, in this checking section there is a check for embedded commas.

If a comma exists in the word it is necessary to surround the word with quotes before placing it in the data statement. Line 280 does that.

Before we can understand the workings of the rest of the procedure it is necessary to look at the way the computer remembers a Basic program in memory. The Electron User Guide (Pages 127 to 129) gives an outline of this.

A Basic program is stored starting at the value of the variable PAGE and extending through to the value of the variable TOP. Type PRINT PAGE and press Return and you will get the result E00.

Type PRINT TOP, then press Return. If you have no program in memory the value printed will be E02.

The difference between the two values indicates the size of the program. The longer the program the higher the value of TOP. Type in another line and see.

Type in lines 80, 490 and 500 of the main program. Line 80 sets the function key so that if it is pressed PROCquery is called.

PROCquery is contained in lines 490 and 500 and displays the contents of each location of memory from PAGE (the start of the Basic program) to TOP (the end of the Basic program).

By pressing function key 1



Memory location	Value
PAGE (E00)	13
PAGE+1 (E01)	Low byte of the first line number.
PAGE+2 (E02)	High byte of the first line number.
PAGE+3 (E03)	Length of the first line
(E04)	
	Characters of the first line.
	13
	Low byte of the second line number.
	High byte of the second line number.
	Length of the second line.
	Characters of the second line.
	13
	Low byte of the last line number.
	High byte of the last line number.
	Length of the last line.
	Characters of the last line.
TOP-2	13
TOP-1	255
TOP	

Figure II: Memory map showing the structure of a Basic program

```

10REM AUTODATA
20REM
30REM (C) Electron User
40REM
50REM W.John Woollard
60REM
70*KEY10OLDIMIN#FX210,11
MREN.1ML.1M
80*KEY1PROCquery!M
90MODE6
100PRINT'''AUTODATA"
110PRINT'''To add a word
to the DATA statements
type the word then Return."
120PRINT'''Press ESCAPE t
o stop the program, then t
ype LIST and press Return t
o see the new statements.
"""
130INPUTword$
140PROCdata(word$)
150STOP

```

```

160REM end of control mod
ule
170REM
180DEFPROCdata(data$)
190*KEY6!LRUNIM
200PRINT'''Press FUNC 6
now !";REPEAT:UNTILGET=12
210LOCALtop%,last%,count%
,len%,v%
220IFLEN(data$)=0THENENDP
ROC
230IFLEN(data$)>247THENEN
DPROC
240IFHIMEM<TOP+&100THENEN
DPROC
250REPEAT:v%=[INSTR(data$,
CHR$34)
260IFv%>0THENdata$=LEFT$(data$,
v%-1)+'''+RIGHT$(data$,
$,LEN(data$)-v%)
270UNTILv%>0
280IFINSTR(data$,CHR$44)>

```

```

0THENdata$=CHR$34+data$+CHR
$34
290len%=LEN(data$)
300top%=TOP:last%=TOP-2
310REPEAT:last%=<last%-1:U
NTIL?last%>13AND?(last%+?(<
last%+3))=13
320IF?last%+4)>220THENP
ROCnewline(data$)
330IFLEN(data$)+(TOP-last
%)>230THENPROCnewline(data$)
)
340?(<last%-2)=44
350FORcount%=<1T0len%:?(to
p%-2+count%)=ASC(MID$(data$,
.count%)):NEXT
360?(<last%-1+len%)=13:?(to
p%+len%)=255?(<last%-3)=?(<
last%+3)+1+len%
370END
380DEFPROCnewline(data$)
390hi%?(<last%+1):lo%?(<

```

This listing is included in this month's cassette tape offer. See order form on Page 47.

From Page 19

the memory can be displayed. Try adding a line to the end of the program to see the change to the output of the procedure.

If you analyse the result you may be able to see the pattern in Figure II.

To allow us to peek inside the memory of the computer (and to change the content of the memory) there are indirection operators. Pages 129 and 130 of the Electron User Guide describe their use.

We will be concerned with the use of "query" - the byte indirection operator.

To peek at the contents of a location, say &E00, type PRINT ?&E00 and press Return. The number printed will be between 0 and 255 inclusive.

To change the value at a particular location, say &E00 type ?&E00=32. The value 32 is placed in location &E00.

Warning! Typing ?&E00=32 will cause the

computer to state bad Program if an attempt to LIST, RUN or SAVE it is made.

Lines 300 and 310 search the memory of the computer starting at just below TOP and working downwards until the next end of line character is met.

Line 310 not only checks it is an end of line character (13) but that it occurs immediately before the start of the next line.

If the ?(last%+?last%+3) is equal to 13 then the 13 encountered is actually a character in the middle of the final line of the program.

The value last% generated at this point is equal to the end of the penultimate line of the program.

Line 320 checks to see if the last line is a data line. The key word for a data line is represented by CHR\$44.

If the last line is not a data statement then PROCnewline() is called to add a new line to the program.

If the last line is a data

statement then line 330 checks that there is enough room to add the word to the end of it. If there is not then PROCnewline() is also called.

It must be noted that both PROCdata() and PROCnewline() do not end with ENDPROC but with END.

This is necessary because after we have artificially extended the last line or added a new line the computer needs to go through the action of OLD, RUN or SAVE before attempting any processing. That is the reason for lines 190 and 200 of the procedure PROCdata().

Line 190 sets up the function key 6 so that when it is pressed it generates CHR\$12 then RUN then CHR\$13 (for example Return).

Line 200 puts up the prompt "Press FUNC6 now!" and waits until CHR\$12 is generated from the keyboard.

There is no significance in the number 12 other than that it is not possible to enter that

value by accidentally pressing any one key.

By pressing function key 6 not only is CHR\$12 generated but the keyboard buffer is also loaded with RUN+Return. So, when the program is ENDED and the cursor appears, the program starts again. On the flow diagram this is referred to as "auto RUN".

Lines 340 to 360 and lines 390 to 470 poke the data statements into the correct memory locations.

Lines 390 to 410 calculate the value of the next line number for the new data statement. It is 10 above the last line number.

These two procedures can now be incorporated into any Basic program which requires the addition of new data statements. Simply typing PROCdata("Electron User")+Return will add that phrase to the last line, if it is a data statement, or add a new line to the program with that phrase as data.

BBC ELECTRON LEAVE REALITY BEHIND YOU WITH... MP ADVENTURE GAMES

A new range of REAL-TIME ADVENTURES we believe the most advanced available for the BBC & Electron: intelligent characters that will move around independently, multiple command statements, advanced text compression methods allow even more detail.

*****NEW*** SADIM CASTLE** It is said that those who enter the haunted estate of Sadim Castle do not return! Long ago tragedy struck here and since then many people have died in mysterious circumstances. Can you uncover the secret and break a terrible curse?

*****NEW*** VALLEY OF THE KINGS** Far away across the desert lie the lost pyramids of Kaculud, rumoured to contain fabulous treasure and a legendary golden mask. The pyramids may now have been discovered although strange happenings have caused the archaeologists to abandon their camp. Your task is to find the golden mask.

Further titles coming shortly: CROWN OF MARDAN, THE FALLEN EAGLE.

Our original and popular text adventures are also available: FIRIENWOOD, WOODLAND TERROR, BLUE DRAGON, SURVIVOR. Please send for full details.

Text Adventure Prices: £7.50 (Cass) £10.50 (Disc) including VAT & postage with UK. State which machine when ordering. Dealer enquiries welcome.

MP SOFTWARE LTD.
165 Spital Road, Bromborough, Merseyside L62 2AE.
Tel: 051-334 3472



4 GREAT PROGRAMS

for BBC B and ELECTRON from

Superb Quality Software



MONEYWISE

£9.95

An up to date yet simple to use personal/home money management system based on not one but TWO main accounts with extensive analysis, budget forecasting, annual summary and 3D BAR CHARTS. Other facilities offered are - SAVE complete program to DISC, PASSWORD protected datafiles (tape or disc), PRINTOUTS of all accounts and 3D Bar chart screen dump.



POLAR PERILS

£7.95

A 100% m/code multiscreen arcade/adventure game set in very hostile Arctic waters. No gun - just your wits and your spear for survival against many sub-zero hazards including ravenous polar bears who will hop on to your tiny ice floe without warning! Great skill is required to get through to the final screen of this very addictive adventure and only then do you realise that your difficulties have only just begun as you try to collect ice blocks in your kayak for your igloo home!



SUPERGOLF

£7.50

Amazingly realistic - the ball speeds into the air, slows, curves down and rolls. Bunkers, water, O.O.B., and a variable gusting wind to cope with! up to 4 players with score card for each!

'I do know a good game when I see one and Supergolf is just that.' ELECTRON USER



TRAFAVGAR

£8.00

Command your own fleet! battle plan unfolds to sea level view of individual engagements. Cannonballs smash into hulls and tear holes in sails! Magazines explode! Ships sink! Fire ships can be sent downwind! Flags are struck and prizes taken! *'Trafalgar is a good combination of Arcade Action and Strategy - a game for the younger war game addict.'* C&VG.

Buy Two - Deduct 20%

All programs available on 40T disc - add £2.00

SQUIRREL SOFTWARE

Dept E,
4 BINDLOSS AVENUE, ECCLES, MANCHESTER M30 0DU
24 Hour answering service - 061-789 4120

Cheques, P.O.s  Same day despatch

ANSWER BACK

SENIOR AND JUNIOR

QUIZZES

GENERAL KNOWLEDGE

THE ULTIMATE CHALLENGE

BBC (32K)•ELECTRON



Juniors (Ages 6 to 11)

-defeat the mythical dragon and save the princess of KOSMOS Castle...

Seniors (Ages 12 and Over)

-pit your wits against the KOSMOS robot and destroy the Alien invaders...

But be warned, in your endeavours you will learn vast amounts of fascinating facts on an enormous variety of subjects. The ANSWER BACK quizzes provide an incredible adventure in education by allowing successful challengers to participate in the compelling games.

Each program comes complete with 15 enormous quizzes providing a total of 750 questions and 3000 optional answers. But that is only the start, you can also create and save an unlimited number of quizzes of your own design.

FEATURES INCLUDE

- Multiple choice answers
- True or False?
- Find the missing letters
- "Pass" facility
- Immediate correction of errors
- Timer option
- Performance summary
- Re-run of questions passed or incorrectly answered
- Full facilities for creating and saving an unlimited number of new quizzes

Available from your computer store or by mail order Price £10.95

Other educational titles include...
'The French Mistress'
'The German Master'
'The Spanish Tutor'



KOSMOS SOFTWARE LTD, 1 Pilgrim Close, Harrington, DUNSTABLE, Beds. LU5 6LX

Please supply the following programs for the BBC/ELECTRON computer

ANSWER BACK Junior Quiz £ 10.95

ANSWER BACK Senior Quiz £ 10.95

Mr/Mrs/Miss...

Address...

Postcode...

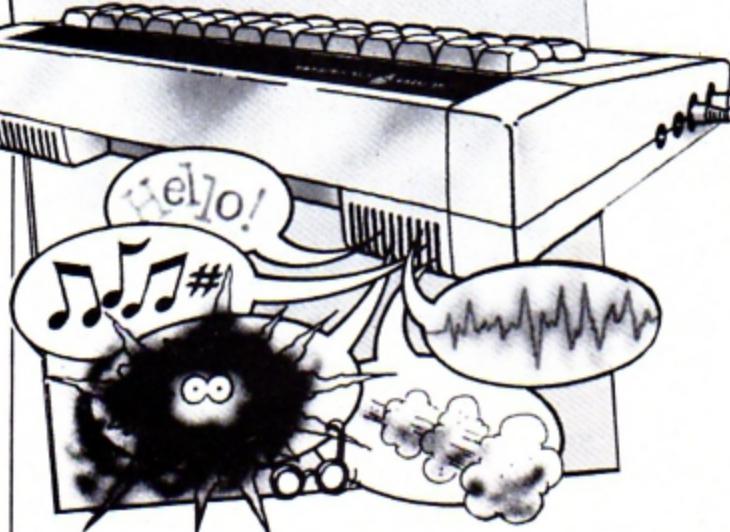


KOSMOS SOFTWARE LIMITED
1 Pilgrim Close, Harrington
DUNSTABLE, Beds. LU5 6LX
Tel. (05255) 3942

Part III of NIGEL PETERS' series on making the most of the Electron's sound channels

Sound advice on how to jump the queue

Noise & music



LAST time we talked about using the SOUND command to write music and I left you with a simple tune-writing program.

By now you should be able to understand what the sound commands are doing when you run Program I. Lines 20, 40 and 60 just use SOUND to play three notes, one after the other. Lines 30, 50 and 70 put the messages on the screen. Nothing difficult there. Try running it again and see if you notice anything odd about the messages. They seem out of step with the notes, don't they?

We know the Electron executes the program lines one after the other in numerical order. Looking at the listing would lead us to suspect that line 20 would make a noise, then the message "Sound 1" would appear on the screen, fulfilling line 30.

Next the Electron would play the note ordered in line 40 and then go on to print the message of line 50, "Sound 2".

Finally line 60's SOUND command would be obeyed and the message "Sound 3" would come up on the screen.

That's what we might expect - but it's not what happens. All the messages appear on the screen while the first note is playing. They stay there stubbornly while the second and third notes are sounded.

Run Program I again and you'll see what I mean. Some parts of the program are being obeyed before others.

It looks like the Electron has executed lines 20 and 30, then lines 50 and 70 before going back to process 40 and 60.

What's happened is the result of the way the Electron's Operating System (OS) is designed. As you know, Electron Basic is very, very fast. It can whip through a simple Basic program like a dose of salts.

However when you come to the SOUND command we're operating on a different time scale. We don't want the sound over and done with in a fraction of a second. We'd never hear it!

We want the note to last for however long we've set the *duration* parameter.

The problem now arises, do we hold up the program while the note plays? If we've set *duration* to 40, do we really want our masterpiece to grind

```
10 REM PROGRAM I
20 SOUND 1,-15,52,10
30 PRINT "Sound 1"
40 SOUND 1,-15,56,10
50 PRINT "Sound 2"
60 SOUND 1,-15,60,10
70 PRINT "Sound 3"
```

to a halt for two seconds while the Electron makes a noise?

What would you feel about a games program that stopped for a few seconds every time it made a sound?

One way round this would be for the Electron to pass every SOUND command it came across over to a special part of the micro that dealt only with producing noises. Then it could get on with the program while the sound generator made the sound.

If, as it was working its way through the program, it came across another SOUND command it would pass the handling of this to the sound generator and carry on.

This is what happened in Program I. The Electron got to line 20 and delegated producing the noise to the sound generator. It was then free to get on with line 30.

Coming to line 40 it found another SOUND command which it immediately passed to the sound generator and went on to line 50, printing the required message.

Line 60 was passed over to the sound producing part of the micro and line 70 was obeyed, displaying the final message.

As each sound has to last for its full *duration* - in this case one second - the messages are printed before the sounds get a chance to finish playing.

It doesn't take your Electron

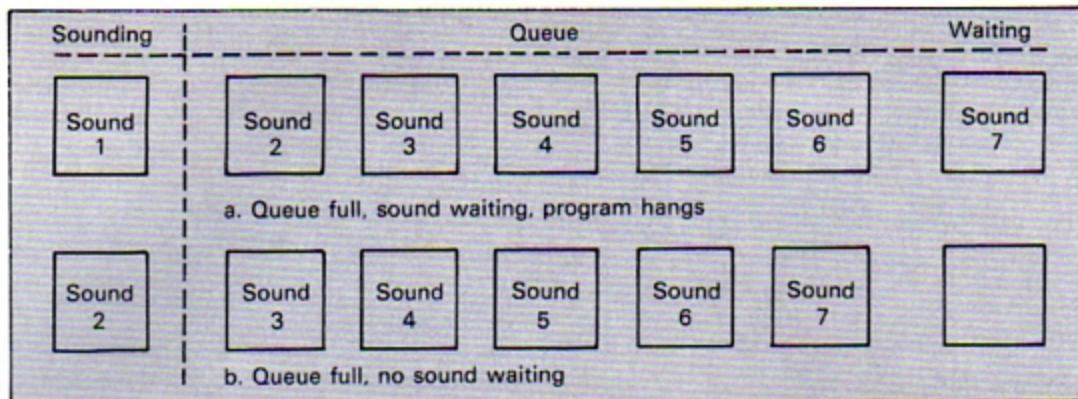


Figure 1: How the queue works

three seconds just to print three messages on the screen. The program finishes, but the sounds keep on sounding long after.

You can picture the sounds as being put in a queue. The first note is played for as long as its *duration* parameter specifies, then the next note in the queue, and then the one after that.

Meanwhile the program itself carries on regardless, shoving any SOUND commands it gets onto the sound generator's queue.

It's almost as if the Electron said to itself: "Ah, here's another SOUND command. I don't want to have to wait for it to play its full length so I'll put it in the queue and get on with the next line of the program".

With this system of queueing sound commands in mind, have a look at Program II.

```
10 REM PROGRAM II
20 SOUND 1,-15,52,20
30 PRINT "Sound 1"
40 SOUND 1,-15,56,20
50 PRINT "Sound 2"
60 SOUND 1,-15,60,20
70 PRINT "Sound 3"
80 SOUND 1,-15,64,20
90 PRINT "Sound 4"
100 SOUND 1,-15,68,20
110 PRINT "Sound 5"
120 SOUND 1,-15,72,20
130 PRINT "Sound 6"
140 SOUND 1,-15,76,20
150 PRINT "Sound 7"
```

Did you notice the slight hesitation between Sound 6 and Sound 7 appearing on screen?

From what we've learnt about the sound queue, we might have expected all seven messages to appear while the sounds take their turn in the

queue. But this isn't what happens.

The program merrily displays the first six messages while the first sound is playing. The last message has to wait for the first note to finish playing before it appears – hence the hesitation.

The explanation lies in the fact that the sound queue only has a limited number of places. In fact it only has places to store five notes, apart from the one that's playing.

When it is asked to store yet another, it accepts it but at a price. The price is that the program halts until the first note has stopped playing and the stored sounds can shuffle up the queue. Figure 1 shows this diagrammatically.

This is what's happened in Program II. The first SOUND command is obeyed and the message printed. The sounds produced by lines 40, 60, 80, 100 and 120 are put in the queue while the messages of lines 50 to 130 are displayed.

When it comes to the SOUND command of line 140 the Electron tries to put it in the sound queue but finds it full. The result is that the program is halted until that SOUND command can be processed.

When the first note has finished playing, the second note starts to be played while the other notes move up one place in the queue.

This leaves room for the SOUND command of line 140 to join the queue. When this happens the queue is no longer full and the program carries on and displays the final message.

Program III uses a FOR...NEXT loop with loop control variable *noise* to produce 50 sounds one after the other. Each sound is a semitone higher than the

previous one.

Line 40 sees to this by making the pitch parameter of the SOUND command equal to *noise* multiplied by four.

```
10 REM PROGRAM III
20 FOR noise=1 TO 50
30 PRINT "Noise number:
":noise
40 SOUND &11,-15, 20+4*n
oise,20
50 NEXT noise
```

Here you can see that after the first six notes – one sounding and five in the queue – the messages appear at one second intervals.

This is because the program has to wait for a space in the queue before it can print the message. Then as soon as it's done this it runs into the next line's SOUND command and so the program hangs again.

So, from what we've covered so far, you should see that the Electron's OS has a queue for storing commands. The capacity of the queue is limited and once it is full any program running has to wait until a vacancy occurs.

Making programs hang is just one problem caused by the queue. There is another.

Imagine a game where you're quite happily zapping aliens. Each time one bites the dust you get a satisfying explosion. These sounds will go into the sound queue.

Now suppose that you're zapped (it comes to us all). The program should make a sad losing noise but what happens if the sound queue is still full of explosions?

What we need is a way to tell the Electron: "Forget the note that's playing, ignore the queue – this is the noise that comes next".

We do this by fiddling with

the *channel* parameter of the SOUND command. Program IV shows this in action:

```
10 REM PROGRAM IV
20 FOR noise=1 TO 50
30 PRINT "Noise number:
":noise
40 SOUND &11,-15, 20+4*n
oise,20
50 NEXT noise
```

What's happening is that we've put &1 in front of the *channel* parameter in line 40. This has told the Electron that this is the noise to make next, and it's to make it immediately.

As the FOR...NEXT loop is producing 50 notes, one after the other, each note cuts short the preceding one. Only the final note runs for the full second.

The slightly less frenetic Program V shows the use of &1 in front of the *channel* parameter. Notice that lines 30 and 50 hold up the program, waiting for a key to be pressed:

```
10 REM PROGRAM V
20 SOUND 1,-15,50,200
30 wait=GET
40 SOUND &11,-15,98,40
50 wait=GET
60 SOUND 1,-15,50,200
```

Line 20 produces a sound which, in the normal course of things, would last for 10 seconds. As the sound queue is empty, the program carries on to line 30 and waits for you to press a key. When you do it goes on to line 40.

Because line 40 has &1 in front of the SOUND command's *channel* parameter, the Electron immediately plays

From Page 23

this note. The first note, if it's still playing, is cut short.

The program then goes on to the next line which again holds things up until a key is pressed. When this happens, it moves on to the SOUND command of line 60 and, if the note produced by line 40 is still playing, puts it in the queue.

So to have a note played immediately we put `&1` in front of its *channel* parameter.

You'll notice that in all the examples so far I've stuck to a *channel* parameter of 1. This makes sense because the Electron only has one sound channel, as opposed to the noise channel we'll be coming to later.

However you might remember I told you that in order to be compatible with the BBC Micro, the Electron would also accept *channel* parameters of 2 and 3. It will,

but be careful.

On the BBC Micro you have three channels and all three can play a note at the same time, producing chords. On the Electron, although *channel* can be 2 or 3, only one note is played at a time.

And if you chop and change channels in an Electron program you might not get the effects you want. Take a look at Program VI:

```
10 REM PROGRAM VI
20 SOUND 1,-15,50,80
30 wait=GET
40 SOUND 2,-15,150,80
```

Notice that when you press a key in order to satisfy line 30, the first note immediately ends and the second begins. This is because they are using different *channel* parameters, 1 and 2.

When the Electron comes across a *channel* parameter which is different from the one

Value	Noise
0	High pitch
1	Middle pitch
2	Low pitch
3	As 1
4	Short periodic
5	Medium periodic
6	Long periodic
7	As 5

Figure II: The noise channel

it's been playing notes on, it stops using the old channel immediately.

Any note that is playing is cut short and the queue ignored, while the note with the new *channel* parameter is played.

The effect is exactly the same as if you had used the same *channel* parameter but with `&1` put in front of it.

Program VII shows this. Each time you press the key, the note that is playing is cut short because the following SOUND command is on a different channel.

```
10 REM PROGRAM VII
20 SOUND 1,-15,50,200
30 wait=GET
40 SOUND 2,-15,99,40
50 wait=GET
60 SOUND 1,-15,50,200
```

This is also why noises made by programs written for the BBC Micro can sound a bit strange on the Electron. If they try to use all three channels at the same time, the Electron interprets this as three notes one after another.

Since the *channel* parameter is changing, so the notes cut each other short, with the odd sounding results.

Finally, what if the *channel* parameter is 0? Try Program VIII which demonstrates the various sounds available on this, the noise channel:

```
10 REM PROGRAM VIII
20 FOR noise=0 TO 7
30 SOUND 0,-15,noise,60
40 SOUND 0,-8,noise,20
50 NEXT noise
```

As you'll have heard, when the *channel* parameter is 0 you get six different noises.

Notice that with the noise channel, the *pitch* parameter is used in a rather different way. It can only have values from 0 to 7. Each value – except 3 and 7 – produces a different kind of noise. This is shown in Figure II.

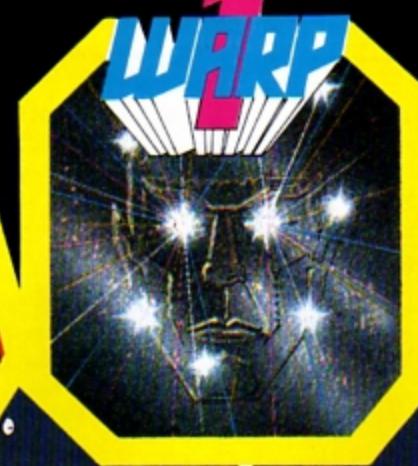
The values 3 and 7 are just there for compatibility with the BBC Micro. On the Electron they just repeat the sounds produced by *pitch* parameters of 1 and 5.

And that's it for this month. I'll leave you to experiment with the various strange sounds available on the noise channel. Have fun. Next month we're going to look at the ENVELOPE command.





WONGO...bounce along the great wall of China to diffuse sizzling bombs... avoiding arrows and grenades. BBC...£7.95 ELECTRON...£7.95



WARP 1...strategical space adventure. Space...the final frontier... Your task...to seek out a lost starship captain, explore new galaxies and boldly go where no man has gone before. BBC...£7.95 ELECTRON...£7.95



ULTRON...Super fast 'zap-em' game. Swarms of invading enemies to blast out of the sky before reaching the mothership. BBC...£7.95 ELECTRON £7.95

Merry Xmas

SANTA



MERRY XMAS
SANTA

Hi-res full colour arcade action as Santa delivers presents on Xmas Eve...avoid hostile snow men flying snowballs, slippery roofs, icicles etc. whilst devouring as much cake, pudding, and sherry as possible on the way...but be careful Santa doesn't get tipsy who knows what might happen SPECTRUM £5.95 ELECTRON £7.95 BBC...£7.95

AVAILABLE FOR
48K SPECTRUM
BBC & ELECTRON



Mail Order:
Send cheque or P.O.
payable to 'ICON' or
quote your credit card no.

Selected titles available from: John Menzies, Co-op, Harrods, Wildings, Granada and most good computer shops.

Dealers contact: Centresoft, Express, Tiger, Drakes, R&R.

ICON

SOFTWARE

65 HIGH STREET, GOSFORTH, TYNE & WEAR, NE3 4AA.
TEL: (091) 2846966

SCRAPBOOK

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

This is where we keep a record – a scrapbook would you believe – 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 sound and 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.

BACH



John Close uses the function keys to turn your Electron into an organ

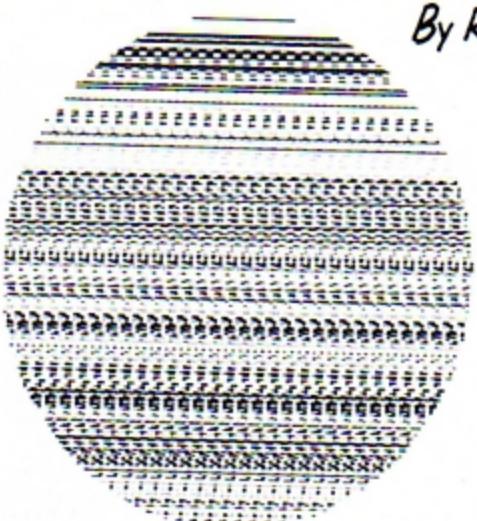
```

10 REM BACH
20 REM JOHN CLOSE
30 REM USE THE FUNCTION
KEYS!
40 MODE6
50 VDU23,1,0;0;0;0;
60 PRINTTAB(17,11)"B A C
H"
70 PRINTTAB(11,13)*Phras
e now playing:"
80 *KEY1*cdedqfedefqfade
FIM"
90 *KEY2*eCfDqEfDqEDCbaq
dIM"
100 *KEY3*eCfCqCaCfDqDaDb
DIM"
110 *KEY4*cdfeqacbDCbCDE
bIM"
120 *KEY5*ECDbCabqCbCabqa
fIM"
130 *KEY6*fqabgbCabCDBCD
EIM"
140 *KEY7*cEdDeCfbqabfCeD
dIM"
150 *KEY8*EeDdCcDDEDCbaf
eIM"
160 *KEY9*qCaDbEDbDaCgbfa
bIM"
170 *KEY0*eCfDdbeCcadbdb
CIM"
180 $S=" c d ef g a bC D
E F G A B"
190 REPEAT
200 INPUTTAB(12,15) T$
210 FOR NX= 1 TO LEN T$
220 P% = INSTR($S,MID$(T$,
NX,1))
230 SOUND 1,-15,P%*4,
240 NEXT
250 UNTIL FALSE

```

LAMP SHADE

By Rog Frost



```

10 REM***LAMP SHADE ***
20 REM***BY ROG FROST ***
30 MODE5
40 VDU23;8202;0;0;0;
50 REPEAT
60 PROCCIRCLE(640,512,50
, RND(4), RND(4), RND(20))
70 UNTIL 0
80 END
90 DEFPROCCIRCLE(X,Y,R,C
,S,L)
100 LOCALI,J
110 FOR I=Y+R TO Y-R STEP
-L
120 S=S+1
130 C=C+1: IF C>7C=1
140 SCOLS,C
150 J=SQR(ABS(R*R-(I-Y)*(
I-Y)))
160 MOVE X-J,I
170 DRAWX+J,I
180 IF INKEY(-74)VDU19,RN
D(3),RND(8)-1,0,0,0
190 NEXT
200 ENDPROC

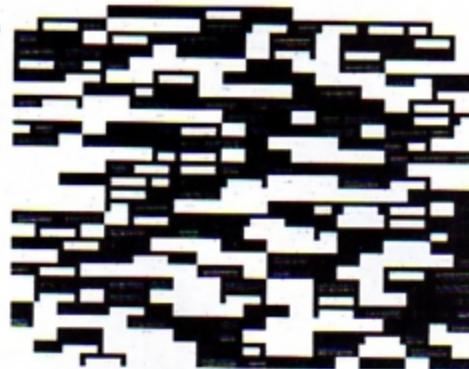
```

PLAYTIME



Stunning Ripple patterns
from P.Bhandari

```
10 REM PLAYTIME
20 REM P.BHANDARI
30 MODE 2
40 VDU 23,1,0;0;0;0;
50 step% = 8
60 end% = 8
70 col% = 0
80 FOR mix% = 0 TO 4
90 FOR x% = 0 TO 1279 STEP
P step%
110 col% = col% + 1: IF col% =
end% col% = 1
120 GCOL mix%, col%
130 MOVE x%, 0
140 DRAW 1279 - x%, 1023
150 NEXT
170 col% = 0
180 FOR y% = 0 TO 1023 STEP
4
190 col% = col% + 1: IF col% =
end% col% = 1
200 GCOL mix%, col%
210 MOVE 1279, 1023 - y%
220 DRAW 0, y%
230 NEXT
250 NEXT
260 REPEAT UNTIL FALSE
```



BLOX

A light and
sound show
from John Close

```
10 REM BLOX
20 REM JOHN CLOSE
30 REM PRESS KEY FOR A NEW PATTERN
40 VDU 23,221,255,255,25
5,3,3,3,3,3
50 VDU23,222,255,255,255
,192,192,192,192,192
60 duff$ = CHR$(222) + CHR$(221)
70 ENVELOPE 3,2,3,4,-230
,-230,-230,-10,126,0,0,-126
,126,126
80 REPEAT
90 MODE2
100 VDU 23,1,0;0;0;0;
110 FOR IX = 1 TO 255
120 COLOUR 128 + RND(5) + 1:6
COLO,RND(5)+1
130 XX = RND(20) - 1
140 X1% = RND(20) - 1
150 Y% = RND(28)
160 Y1% = RND(28)
170 PRINTTAB(XX, Y%) duff$;
180 PRINTTAB(X1%, Y1%) duff$;
190 PRINTTAB(XX, Y%) duff$;
200 PRINTTAB(X1%, Y1%) duff$;
210 SOUND 1,3,IX,XX DIV 9
220 NEXT
230 A = GET
240 UNTIL 0
2
10 REM LIGHTS
20 REM JOHN CLOSE
30 MODE 2: X = 3.5
40 VDU23,1,0;0;0;0;
50 REPEAT
60 FOR J = 1 TO 20 STEP X
70 FOR K = 1 TO 20 STEP X
80 COLOUR RND(6) + 128
90 GCOLO,RND(15) + 1
100 PRINT TAB(J, K + 5) CHR$3
2
110 PRINT TAB(K, J + 5) CHR$3
2
120 NEXT: NEXT
130 UNTIL FALSE
```

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

Software Surgery

THE COLUMN THAT TAKES A LOOK INSIDE THE LATEST RELEASES

Help save the Ospreys!

Osprey!

Bourne Educational Software

I WAS lucky enough to be able to spend two weeks in Scotland this summer and the highlight of the trip was my visit to see the nesting Ospreys at Loch Garten.

So, when *Osprey!* arrived in the office, I grabbed it with enthusiasm.

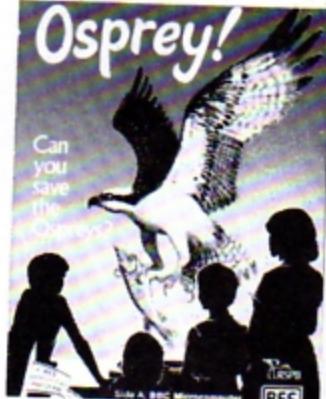
Produced in conjunction with the RSPB, and with an excellent 32 page colour booklet to complement the program, it's a fascinating simulation of the problems faced by the osprey as a Scottish breeding bird.

You take the part of the manager of a nature reserve where ospreys are nesting. The booklet has given you an outline of the history of the osprey and you have to pick which year you want the simulation to start.

The earlier the year, the harder the game is. Your aim is to make sure that the birds successfully breed and rear their chicks.

To do this, you have to decide what your limited number of wardens is going to do during the vital spring and summer seasons.

Some are needed to chase



away the egg stealers, while others have to manage the site and keep disturbance from the visitors to a minimum. Also wardens have to be spared to make people aware of the ospreys and to encourage public support.

And, just like real life, when you've made your choices and allocated your resources you have to sit back and watch what happens.

The graphics are beautiful, painting a picture of the reserve and the nest site. You can watch the ospreys as they swoop down to fish and take them to the nest.

Sadly, if you haven't allocated enough wardens to guard duty, you can also watch the egg thieves at work. Even the visitors can be a nuisance, their cars disturbing the birds if

you haven't picked the right number of site wardens.

And to make it worse, factors totally out of your control such as the weather affect the final result.

Your success or failure at one reserve is taken as representative of the whole of Scotland and after the spring season you're shown how the osprey population has fared under your protection.

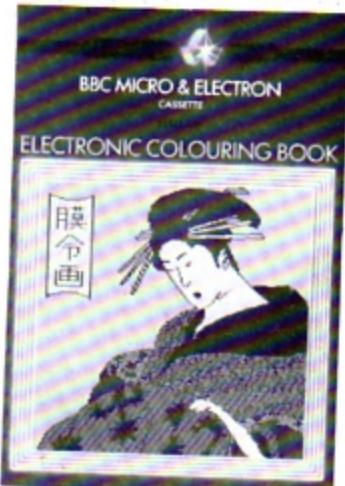
You continue until you reach the year 1981 or you've run out of ospreys - a horrible thought. You can then compare your efforts with the magnificent results the RSPB achieved in reality which are shown in the booklet.

It's a smashing program. The instructions, both in the booklet and on the screen are excellent. The graphics and animation are more than adequate and the whole package has the quality that we've come to expect from Bourne.

Even the fact that it's educational - the well-illustrated booklet has a history of the Osprey and a things-to-do section - doesn't spoil the fun.

So if you haven't been to Loch Garten yet, you can console yourself playing *Osprey!* until you get the chance.

Nigel Peters



New pictures every day

Electronic Colouring Book

Addison-Wesley Software

MANY years ago, when playing with jigsaws was more fun than VDU codes, I used to love painting by numbers.

There were two main problems, however. The first was that I was too impatient to wait for the colours to dry so that they ran together.

The other drawback was

Safari fun for all - with no blood

Jungle Jive

Virgin Games

IN a time when the big game hunter is thankfully a thing of the past, *Jungle Jive* lets you release your aggressive instincts without spilling blood.

It brings all the excitement of a jungle safari onto your TV screen.

You control a little man who waddles up and down the centre of the screen. The idea is that you blast away at the animals who are closing in on



you to your left and right. You get points for each one you hit although I pretend they're just tranquillising darts.

As you're getting yourself a life ban from the RSPCA, avoid the slithering snake who dashes about at random trying to avenge all those departed elephants, crocodiles, lions and birds.

Whatever you do, don't shoot the cute little baboons which line your path. They protect you and you'd better protect them or else you're

finished.

And don't try to shoot the blue hippo. Bullets bounce off him.

You start off with the usual three lives, gaining bonus lives as your point score mounts. You lose them as you bump into things and things bump into you.

It's a nice action game that promises fun for all the family - once you've settled the arguments over who's going next.

Keith Young

From Page 29

that each picture could only be painted once, so I couldn't experiment with colours as much as I liked.

I would have loved a system which allowed me to dabble, change colours as I wanted, and where every new day meant a fresh lot of clean pictures.

Well, rather late for me, but still good fun comes this tape which is compatible with both the Electron and the BBC Micro.

Although only priced at £9.95 – a modest enough figure nowadays – it contains no fewer than 18 pictures waiting for your artistic talents.

The first four each have a file which allows the correct colours to be loaded onto the picture by first loading the picture file itself followed by the colour file.

The remaining 14 pictures do not have this facility, but this gives full rein to your imagination.

By the way, how many colours would you expect your Electron to support? Wrong! There is a palette of 35 available, including greys, pinks and so on and an area is easily filled using machine code.

A cursor is moved around the screen, and on moving into the palette can select the painting colour. By then moving the cursor to an area, it can easily be filled with the P (paint) key.

Similarly, it may be recoloured by D (delete), selecting another palette colour and then painting once more.

The speed of the fill is quite impressive as is the speed of the cursor. The picture as drawn is not final, as a mode may be selected in which it is possible to add lines exactly as required.

Thus the picture of the clown might be improved by the addition of some balloons which can be added easily.

Indeed, it is quite possible and fun to go immediately into drawing mode without loading a picture, and thus create a picture from scratch which can then be coloured using the palette. The finished result can be saved to cassette.

An amusing but not over-

useful feature is the facility to randomly alter the colours on a displayed picture, or to cycle through the basic colours in order.

The cassette box claims this program will interest those from six to 96. Well, my three and five year olds would like to be added to that list as they both think it's smashing fun.

I don't think they realise there is quite a large educational content to the program, with much evidence of planning, hand-eye coordination and discussion leading to the final polished result.

Whether it would be of real practical use in schools on cassette is doubtful. There are so many parts of the program that it cries out to be put onto disc for easier access of a particular picture, or for rapid saving of little Johnny's masterpiece.

Apart from that reservation, I am most impressed.

Phil Tayler

Defend the fleet

3D Bomb Alley Software Invasion

IS it because war is so much fun that we turn it into games? Or is it because we play so many war games that we go so eagerly to war?

These were some of the more serious thoughts sparked off by playing Software Invasion's game 3D Bomb Alley.

Mind you, there isn't too much time to think when you're actually playing the game!

The scenario is obviously based on San Carlos Bay in the Falklands. Your fleet is at anchor in a narrow sea inlet



and you are under attack from enemy planes. These appear in the far distance but rapidly grow larger as they near.

To defend yourself you have to throw up a barrage of anti-aircraft fire. The trouble is that, although you can miss the planes, if they get through they don't miss you.

You get an extra ship for each 10 planes downed and the game ends when you've

A well produced bunch

Science 1 Shards

THIS package consists of four separate programs on balances, meter reading, thermometer reading and lenses.

The meter program is on twice, in Mode 1 and in Mode 0. The Mode 0 version added nothing – I preferred the extra colours of Mode 1.

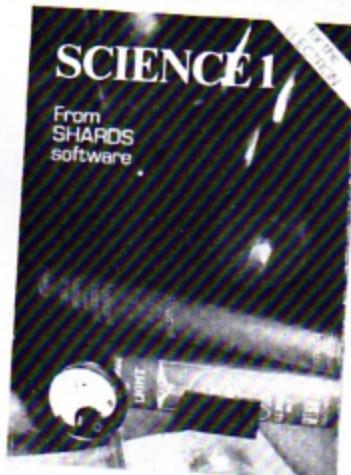
The introductory program has a noisy title plus an index. I expected the index to give single letter entry to load a program, but the options are to see the index or leave the program.

Leaving the program gives you a blank screen and it is necessary to CHAIN whichever program you want.

The trouble is, you've now forgotten their names and the sheet of information has different titles for them.

The balancing program collects your name, then gives a menu of options. You have to decide whether to be nice or nasty – there is no explanation as to what this means at this stage!

The program gives practice in working out how to balance



see-saws. The explanation is rather sketchy, but if you do get the answer correct a little diver hurls himself into a cup of liquid or, if you chose nasty, he goes splat on the floor.

If you get the answer wrong, large arrows indicate which way the see-saw tips, and then the diver splats if you are nice or splashes if you are nasty.

The meter reading program provides practice on reading the two most common school meter scales. It is well constructed, making good use of

large text, and with an option to magnify the relevant part of the scale.

The program is rather fussy, four key presses are required before an answer is put in.

The thermometer program is very similar to the meter program. It provides practice in reading 0-100°C, 0-250°F and clinical °C thermometers.

Light provides a lesson in ray optics at concave/convex lenses/mirrors. It is again well constructed using good text and attractive, simple graphics. The whole program is rather slow, particularly the 16 questions.

Overall these are well-produced programs with the meter and thermometer sections the pick of the bunch.

The major disadvantage of this educational package is total non-compatibility with the BBC Micro. If these programs are run on a Beeb, they have a nasty little trick – they clear the micro's memory.

Many schools have BBCs and Electrons. I would choose a program that would run on both machines to use in my school.

Rog Frost

lost your last ship.

It's a simple game with nice graphics and adequate instructions. The way the planes appear in the distance and then grow larger is a good technique but it's annoying when they slow down and even appear to stop when under fire. You can almost feel the micro thinking.

At first I thought that that would mean a slow game, but I soon learned differently as the planes came in at me five at a time.

The main difficulty comes from the increasing number of enemy planes. If you like action where quick reactions are at a premium then this is the one for you.

Eileen Young

Command a missile silo - it's not easy!

3-D Tank Zone

Dynabyte

IT'S not easy being the commander of a missile silo.

First there are the aircraft attacking you, then there are the tanks. And you're stuck in a hole in the ground with only anti-tank missiles and an anti-aircraft gun to ward off this unprovoked aggression. All the time your energy is getting lower and the shields weaker.

No, it's not easy...

Still, if you think you could do any better, have a go at 3-D Tank Zone.

Your Electron's screen becomes a view from the silo as you scan for the enemy. As you guide the sights of your AA gun to attack the jets and helicopters on the horizon, you have to watch the radar for tanks.

When you see one you have to turn the turret and loose off a missile, quick. At first the action seems a little slow but as the tanks get nearer and your energy drops it's all too fast.

The graphics are simple but effective. The tanks appear in 3D wire form and the missiles fly in an annoyingly realistic manner. I say annoying



because of the way they miss.

The instructions are thorough and the key controls are well-laid out.

It's an interesting game, very different from anything else I've seen for the Electron. While not the fastest program around, it should suit those looking for a change from the usual arcade remakes.

Well worth looking at.

Tony Sinclair

Enter the arena and battle it out

Arena 3000

Microdeal

IMAGINE that you're suddenly transported forward in time to the year 3000 AD. You find yourself the star of the chief entertainment of the time - the arena - where humanoids battle with mutants.

Your only defence is a death ray and you need it. If you're touched just once by a mutant you die. And some of the monsters take several blasts before they decide to die and leave you in short-lived peace!

A nightmare? No, just a brief description of Arena 3000. You, of course, play the part of the humanoid, starting with three lives but soon losing them.

Each mutant you kill adds to your points score and the

cassette inlay tells you that you gain an extra life for every 20,000 points you score.

I wouldn't know as by then I've been swamped by mutants such as *The Dreaded Oh Nos* or the *Jovial Jovian Jumpers*.

If they're jovial I don't get the joke.

And of course, every time you clear a wave of mutants along comes another of a different type.

It's not easy but it is fun. With either keyboard or joystick control, sound on/off and pause facilities and a Hall of Fame, the game is well up to standard.

The graphics are very good and the use of sound reasonable. A good version of an arcade classic.

Keith Young

Not for mere mortals... it's just too good

Nightmare Maze

MRM Software

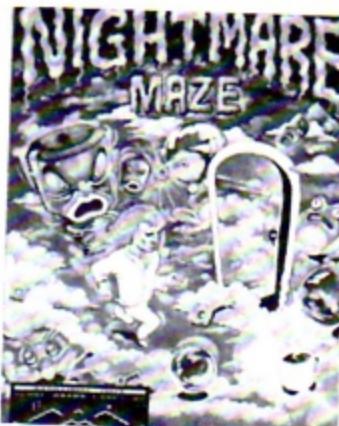
NIGHTMARE Maze, written by Mike Williams, is a descendant of Pac Man, which was popular ages ago in the arcades. Although it is easy to see the origins, the game is completely different to play.

Instead of running around the maze in between the walls, you actually run on top of them. They are drawn in perspective, as if you were looking down at an angle.

There are four screens, each with a different maze and monsters. The first is inhabited by springy things, the second by bouncing balls, the third by deadly frogs, followed by hungry hoppers. I can't confirm the last two as I always got bounced on the second screen.

The object of the game is to collect a number of keys which appear around the maze. On acquiring the last you can unlock the door which leads to the next screen.

The inhabitants of the maze don't chase you, they move in



fixed patterns. The routes taken are sufficiently complex as to make it very difficult to remember more than the first few.

Help is at hand in the form of a cup of black coffee. Drinking this awakes you from your nightmare and the nasties disappear. You soon start to dream again so you must rush round collecting the keys as fast as possible before they reappear.

The graphics are excellent and the animation very smooth with good sound to accompany the springy/bouncy hoppers.

Having said that though, I honestly didn't enjoy playing this game, the reason being that it is just too difficult and too frustrating. Maybe I'm just too old!

Not once in three weeks have I made the high score table, not even the bottom position.

The man is difficult to control when the monsters disappear, often running straight past the path you wish to turn and run along, and when you are caught, all the keys you have so painstakingly collected are lost and you must start again.

This game is for advanced arcadians only, providing an exciting new challenge to their skill. Us ordinary mortals haven't a chance, it really is a nightmare!

Roland Waddilove



Christmas Box



CHRISTMAS Box is a game of strategy designed to while away a pleasant time after a substantial Christmas dinner.

The rules are simple: Two players take turns to enter a Christmas Box into the grid by pressing a letter key from A to F.

The piles of presents build up to the top, but no further, to the accompaniment of *We Wish You A Merry Xmas*.

Q cuts the music. S starts it again.

The winner is the first person to get four of their presents in a straight line – vertically, horizontally or diagonally.

If you don't want to type the whole thing by hand then send off for this month's tape. If you want to learn some new

techniques to aid your programming then read on.

And if you want to write a program with a structured format so that it can be changed to your requirements then start key tapping.

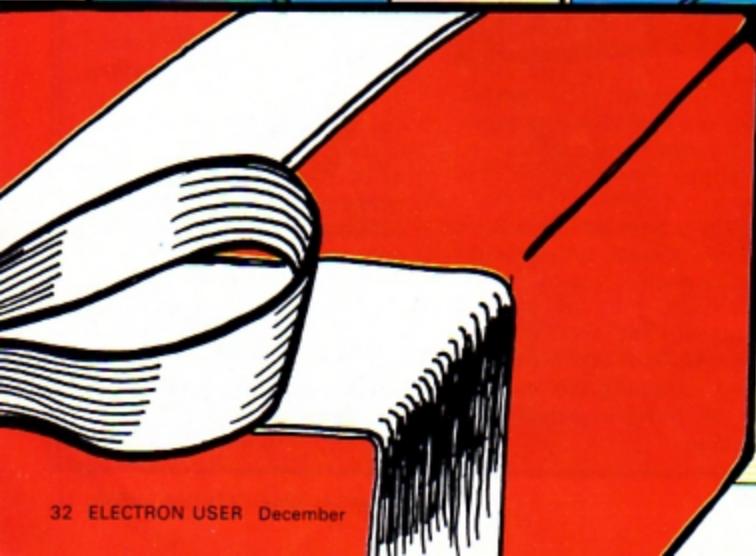
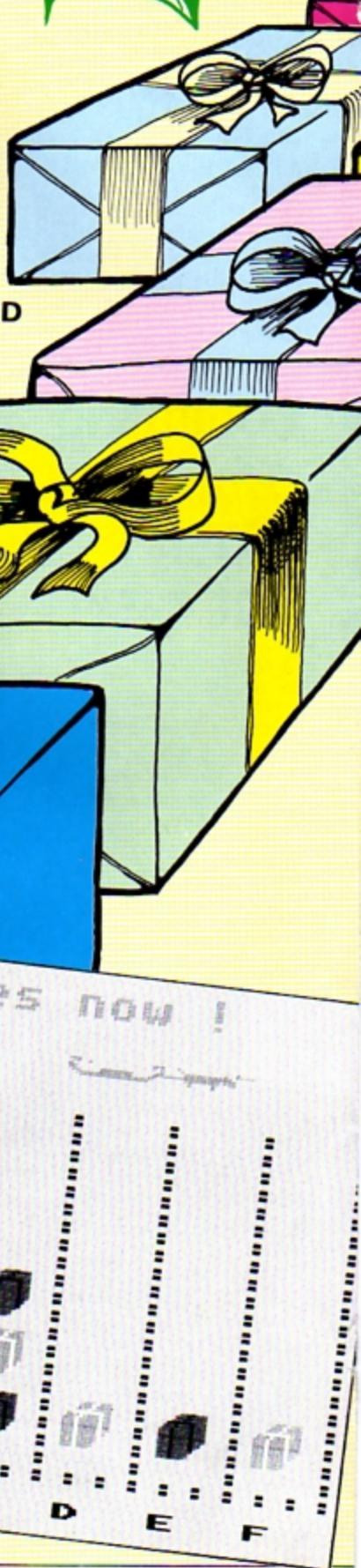
The program has been written so that another programmer can easily see how it works and therefore be able to change it to suit themselves.

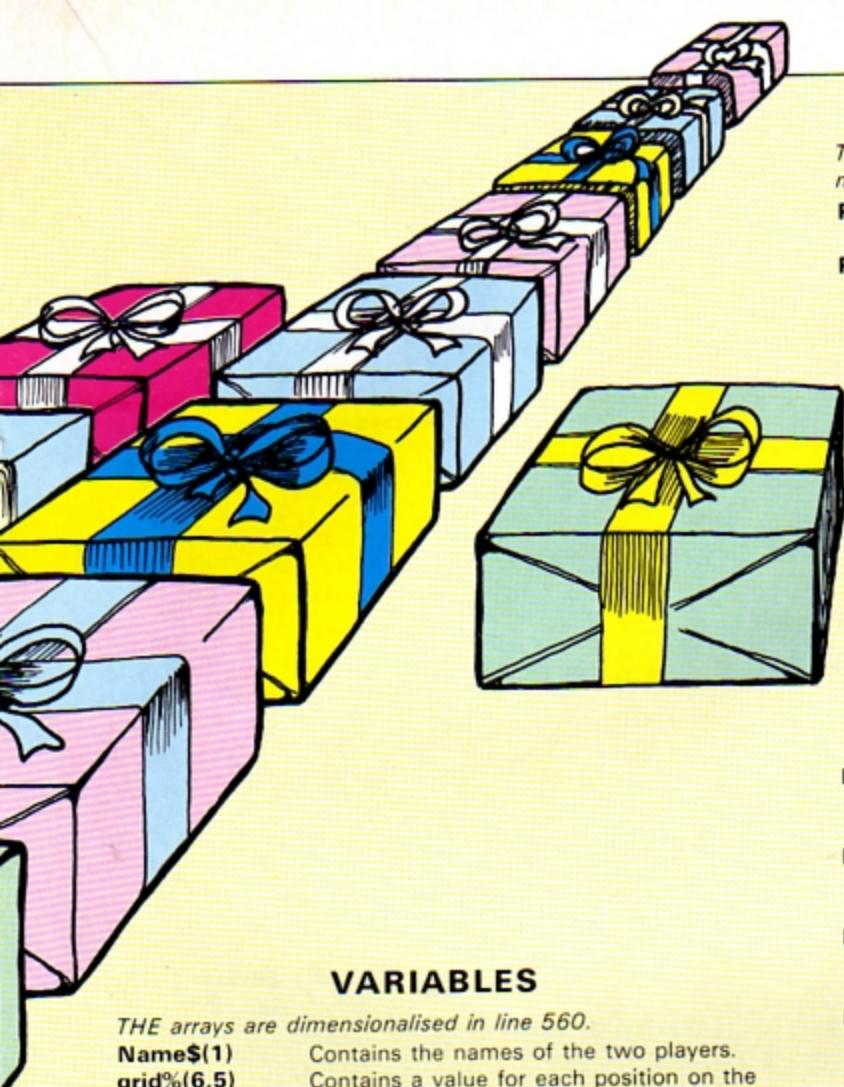
The first 19 lines do everything. They form the main control module.

The lines call procedures or functions in a sequence that creates the effect seen on the screen. There is a single REPEAT UNTIL loop which continues until one or other of the players has won.

After the end message has been displayed the program automatically starts again.

A game of strategy by JOHN WOOLLARD





VARIABLES

THE arrays are dimensioned in line 560.

Name\$(1)	Contains the names of the two players.
grid%(6,5)	Contains a value for each position on the display. Those unique values are used to determine if a player has a winning combination of Christmas Boxes.
win%(40)	Contains the totals of the 39 winning positions.
pitch%(54) and duration%(54)	Contain the pitch and duration of the 54 notes of the tune that plays during the game.
win%	A flag that starts at zero and becomes positive to show that a player has achieved a winning position.
person%	Either 1 or 0 and indicates which player's turn it is.
k, k1, k2 ...	General purpose counters that do not cross procedure boundaries.
shape\$	Contains a string that produces the moving sleigh at the top of the screen.
T%	Used to count through each note of the tune and determine the position of the sleigh on the screen.
get%	Stores the value of the player's choice of letter.
I and inkey%	Temporary stores of the result of an inkey statement.

MODIFICATIONS

THE program was designed to be flexible in its use. The tune can be changed by changing the values in the data statements of lines 1760 and 1770. The shape of the Christmas Boxes can be

changed entirely by altering lines 1690 to 1720.

The shape of the moving sleigh is set in line 470.

Obviously, all text can be changed by changing the appropriate PROCPrint calls.

PROCEDURES AND FUNCTIONS

THE following procedures and functions are called from the main control module:

PROCtitle

Sets up the initial screen with a display of grid and title.

PROCinitialisation

Dimensions all variables used, reads the data statements and assembles a machine code routine that creates double height characters. (That routine was described and explained in the July edition of *Electron User*.) The initialisation procedure also contains several *FX calls which are useful in many programs. *FX16,0 disables the analogue/digital sampling. This is an advantage only if the Plus 1 is fitted. *FX229,1 disables the action of Esc. It may be useful to set it to *FX229,0 while debugging is carried out. The other *FX calls are documented in the *Electron User Guide*.

PROCnamein

Allows the two players to enter their names and wishes them luck.

PROCdisplay

Creates an empty grid for the players to enter their Christmas Boxes.

PROCplay

Waits for the player to make a choice and enters the Christmas Box.

FNcheck

Checks to discover if that move was a winning move. If it was then win% is set to a positive value.

PROCendmessage

Displays its Christmas Greetings when the loop ends.

PROCget

Waits for a letter key to be pressed between A and F. However, if Q is pressed the sound is quietened, if S is pressed then it is enabled. The procedure uses *FX210,0 to enable all sound output and *FX210,1 to disable all sound output.

FNname

An input routine that prints the inputted characters in double height to a maximum length of 12 characters.

PROCPrint (x,y,COL,a\$)

Utilises the machine code routine created in the initialisation procedure so that whole strings can be printed in double height.

PROCshape (shape%, colour%, xcoord%, ycoord%)

Displays a coloured shape determined by the value of shape% and colour% at a position on the screen determined by xcoord% and ycoord%.

DATA

IT IS most important that these statements are entered accurately as mistakes may not appear as syntax errors but as spurious errors whose source is hard to determine.

Lines 1730, 1740 and 1750 contain the winning combinations. Lines 1760 and 1770 contain the notes of the tune.

**Full listing starts
on Page 34**

Christmas Box listing

From Page 33

```

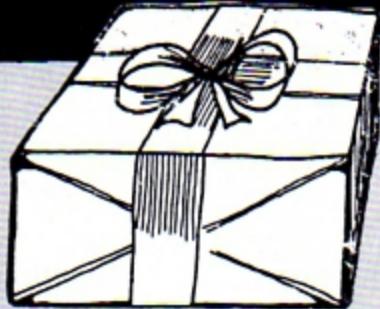
10 REM Christmas Box
20 REM John Woollard
30 REM (C) Electron User
40 REM Happy Christmas
50 Program$="Christmas B
ox"
60 MODE2:VDU23,1,0;0;0;0
:0
70 COLOUR132
80 PROCtitle
90 PROCinitialisation
100 PROCnamein
110 PROCdisplay
120 go%:0:win%:0
130 REPEAT:go%:go%+1
140 person%:go%MOD2
150 PROCplay
160 win%:FNcheck
170 UNTILwin%
180 PROCentmessage
190 RUN
200 DEFPROCtitle
210 CLS
220 COLOUR8
230 PRINTTAB(10-LEN(Program$)DIV2,1)Program$
240 COLOUR7
250 FORK1=1TO20:FORK2=0TO
5
260 PRINTTAB(1+3*k2,7+k1)
;: "
270 NEXT:NEXT
280 PRINTTAB(1,28);STRING$(19,":");
290 FORK1=1TO5:FORK2=1TO6
300 PROCshape(1,RND(16)-1
,k2*191-50,k1*110+120)
310 NEXT:NEXT
320 COLOUR6
330 PRINTTAB(2,3)"See Ele
ctron User"
340 PRINTTAB(2,4)"for ins
tructions."
350 COLOUR7
360 ENDPROC
370 DEFPROCinitialisation
380 DIM dblp &FF:FOROpt=0
T02STEP2:P%:dblpl:[OPT Opt:S
T&70:STX&79:STY&7A:LDA#10:
LDX&70:LDY#0:JSR&FFF1
390 LDA#23:JSR&FFEE:LDA#2
55:JSR&FFEE:LDA#71:JSR&FFEE
:JSR&FFEE:LDA#72:JSR&FFEE:J
SR&FFEE:LDA#73:JSR&FFEE:JSR
&FFEE:LDA#74:JSR&FFEE:JSR&F
FEE:LDA#31:JSR&FFEE:LDA#79:
JSR&FFEE:LDA#7A:JSR&FFEE:LD

```

```

A#255:JSR&FFEE
400 LDA#23:JSR&FFEE:LDA#2
55:JSR&FFEE:LDA#75:JSR&FFEE
:JSR&FFEE:LDA#76:JSR&FFEE:J
SR&FFEE:LDA#77:JSR&FFEE:JSR
&FFEE:LDA#78:JSR&FFEE:JSR&F
FEE:LDA#31:JSR&FFEE:LDA#79:
JSR&FFEE:LDA#7A:ADC#1:JSR&F
FEE:LDA#255:JSR&FFEE:RTS:J
NEXT
410 *KEY10OLD:ML.INIM
420 *FX16,0
430 *FX4,1
440 *FX210,0
450 *FX11,0
460 *FX229,1
470 shape$=CHR$32+CHR$249
+CHR$254+CHR$253+CHR$252+CH
R$251+CHR$250+CHR$8+CHR$8+C
HR$8+CHR$8+CHR$8+CHR$8
480 VDU23,254,0,0,0,0,255
,255,255,255
490 VDU23,253,7,5,1,3,7,1
2,248,240
500 VDU23,252,16,8,7,7,25
5,7,2,2
510 VDU23,251,0,1,255,255
,253,252,8,8
520 VDU23,250,32,62,128,1
92,192,192,0,0
530 VDU23,249,224,100,128
,192,192,96,63,31
540 ENVELOPE1,1,48,96,48,
1,1,1,128,0,0,-126,126,126
550 TX=-1
560 DIMName$(1),grid%(6,5
),score%(1),total%(1),win%(1
,40),pitch%(54),duration%(54
)
570 FORK1=1TO39:READwin%(k1
):NEXT
580 FORK1=1TO5:FORK2=1TO6
590 grid%(k2,k1)=2^(k1+(k
2-1)*5)
600 NEXT:NEXT
610 len%:54
620 FORK1=0TOlen%
630 READpitch%(k1),durati
on%(k1)
640 NEXT
650 ENDPROC
660 DEFPROCnamein
670 CLS:PROCPrint(1,2,130
,"Hello,")
680 PROCPrint(1,5,130,"Pl
ease type in your")
690 PROCPrint(1,8,130,"na
me then [RETURN]")
700 *FX21
710 Name$(0)=FNname
720 CLS:PROCPrint(1,2,130
,Name$(0))
730 PROCPrint(1,5,130,"Pl
ease type in your")
740 PROCPrint(1,8,130,"fr
iend's name ")
750 *FX21
760 Name$(1)=FNname
770 CLS
780 PROCPrint(4,4,130,Nam
e$(0))
790 PROCshape(0,1,100,880
)
800 PROCPrint(4,7,130,Nam
e$(1))
810 PROCshape(1,2,100,760
)
820 PROCPrint(1,18,130,"G
ood Luck"):#FX21
830 I=INKEY(300):CLS:ENDP
ROC
840 DEFPROCdisplay
850 COLOUR7
860 FORK1=1TO20:FORK2=0TO
6
870 PRINTTAB(1+3*k2,7+k1)
;: "
880 NEXT:NEXT
890 PRINTTAB(1,28);STRING
$(19,":");
900 FORK2=1TO6
910 PRINTTAB(3*k2-1,30);C
HR$(k2+64)
920 NEXT
930 PROCPrint(4,1,129,"g
oes now !")
940 ENDPROC
950 DEFPROCplay
960 PROCshape(person%,per
son%+1,100,975)
970 REPEAT
980 PROCget
990 IFgrid%(get%,0)=5THEN
SOUND1,-15,23,20:get%:0
1000 UNTILget%:0
1010 PROCshape(person%,per
son%+1,getX*191-50,grid%(g
et%,0)*110+220)
1020 grid%(get%,0)=grid%(g
et%,0)+1
1030 total%:person%:total
%:person%+grid%:get%,grid%
(get%,0)
1040 ENDPROC
1050 DEFFNcheck
1060 check%:0
1070 FORK1=1TO39
1080 IF(total%:person%):AND
win%(k1):=win%(k1)THENcheck
%:k1
1090 NEXT
1100 =check%
1110 DEFPROCendmessage
1120 FORK1=1TO5:FORK2=1TO6
1130 IF(win1(win%):ANDgrid%
(k2,k1))=grid%(k2,k1)THENPR
OCshape(3,8,k2*191-50,k1*11
0+120)
1140 NEXT:NEXT
1150 PROCPrint(4,1,129,STR
ING$(12,""))
1160 PROCPrint(4,1,129,Nam
e$(person%))
1170 *FX21
1180 inkey%:INKEY(900)
1190 CLS
1200 PROCPrint(1,1,129,Nam
e$(person%))
1210 PROCPrint(1,4,129,"is
the winner")
1220 PROCPrint(1,9,129,"Me
rry Christmas")
1230 PROCPrint(1,11,129,"f
rom all at")
1240 PROCPrint(1,13,142,"E
lectron User")
1250 PRINTTAB(1,29),"Press
Return"
1260 *FX21
1270 REPEAT:UNTILGET=13
1280 ENDPROC
1290 DEFPROCget
1300 *FX21
1310 REPEAT
1320 TX=TX+1
1330 IFTXMOD14=0THENPRINTT
AB(0,4)STRING$(40,"")
1340 PRINTTAB(TXMOD14,5)sh
ape$;
1350 SOUND1,1,pitch%(TXMOD

```



```

len%),duration%(T%MODlen%)/
1.3
1360 SOUND1,0,0,1
1370 inkey%=(INKEY(7)OR32)
-96
1380 IF inkey%==17 THEN SOUND1
,0,0,99:FX210,1
1390 IF inkey%==19 THEN FX210
,0
1400 UNTIL inkey%>0 AND inkey
%7
1410 get% = inkey%
1420 ENDPROC
1430 DEFN name
1440 REPEAT
1450 Name$=""
1460 PROCPrint(1,11,130,ST
RING#(18,""))
1470 REPEAT:G=GET
1480 Name$=Name$+CHR$(G)
1490 PROCPrint(1,11,130,Na
me$)
1500 SOUND1,-15,230,1
1510 UNTIL G=13 OR G=127 OR LEN
(Name$)>12
1520 UNTIL G>127 AND Name$<>
CHR$13
1530 =Name$
1540 DEFPROC Print(x,y,COL,
a$)
1550 COLOURCOL-128
1560 FORK=1 TO LEN(a$)
1570 AZ=ASC(MID$(a$,K,1))
1580 IF AZ>127 AND AZ<144 THEN
COLOURAZ-128:AZ=32
1590 IF AZ<32 THEN AZ=32
1600 X% = x+K-1:Y% = y:CALL dbl
P
1610 NEXT
1620 COLOUR7
1630 ENDPROC
1640 DEFPROC shape(shape%,c
olour%,xcoord%,ycoord%)
1650 GCOL0,colour%
1660 GCOL0,128+(colour%-1)
*7
1670 COLOURcolour%
1680 MOVE xcoord%,ycoord%
1690 IF shape%==0 THEN PLOT1,0
,0:PLOT1,0,0:PLOT81,0,-50:P
LOT81,50,0:PLOT81,0,50:PLOT
81,-50,0:PLOT81,40,30:PLOT8
1,50,0:PLOT81,-40,-30:PLOT8
1,40,-20:PLOT81,-40,-30:PLO
T0,20,15:PLOT3,0,50:PLOT3,-
50,0:PLOT0,45,15:PLOT3,-40,
-30:PLOT3,0,-50:ENDPROC
1700 IF shape%==1 THEN PLOT1,0
,0:PLOT1,0,0:PLOT81,0,-50:P
LOT81,50,0:PLOT81,0,50:PLOT
81,-50,0:PLOT81,40,30:PLOT8
1,50,0:PLOT81,-40,-30:PLOT8
1,40,-20:PLOT81,-40,-30:PLO
T0,20,15:PLOT3,0,50:PLOT3,-
50,0:PLOT0,45,15:PLOT3,-40,
-30:PLOT3,0,-50:ENDPROC
1710 IF shape%==3 THEN PLOT0,0
,-25:VDU5:COLOUR15:PRINT"X"
:VDU4,23,1,0;0;0;0;0;0:COLOUR
1:ENDPROC
1720 ENDPROC
1730 DATA30,60,960,1920,30
720,61440,983040,1966080,31
457280,62914560,1.00663296E
9,2,01326592E9
1740 DATA67650,2164800,692
73600,135300,4329600,138547
200,270600,8659200,27709440
0,541200,17318400,554188800
,1082400,34636800,1.1083776
E9
1750 DATA532610,17043520,5
45392640,1065220,34087040,1
,09078528E9,69904,2236928,7
1581696,139808,4473856,1431
63392
1760 DATA60,10,80,10,80,5,
88,5,80,5,76,5,68,10,52,10,
68,10,88,10,88,5,96,5,88,5,
80,5,76,10,60,10,76,10,96,1
,0,96,5,100,5,96,5,88,5,80,1
,68,10,60,5,60,5,68,10,88,
10,76,10,80,20
1770 DATA60,10,80,10,80,10
,80,10,76,20,76,10,80,10,76
,10,68,10,60,20,88,10,96,10
,88,5,88,5,80,5,80,5,108,10
,60,10,60,5,60,5,68,10,88,1
0,76,10,80,20,252,0

```

This listing is included in this month's cassette tape offer. See order form on Page 47.

A WINNING HAND



PETTIGREWS DIARY £7.95
 Three Part 90K adventure taking you from the serene Oxfordshire countryside through bustling London streets, to an epic journey through Europe. Superb family entertainment filled with intrigue, action and suspense. "Pride of place for the most original and entertaining Adventure - it's got to be value for money" *Computer and Video Games* (adventure supplement).

EMPIRE £6.95
 Exciting game of strategy with eight levels of difficulty. Destroy the evil Empire before it conquers the world - Played on a world map. "This is a terrific game - visually attractive and highly addictive" *Home Computing Weekly*.

MYSTERY OF THE JAVA STAR £7.95
 Enormous four part adventure with 3 playing levels. Mount an expedition to the South Seas to discover the secrets of the 18th century shipwreck. Absorbing family fun with puzzles, challenges and interesting facts.

AVAILABLE NOW AT ALL GOOD STOCKISTS or by MAIL ORDER (send cheque/PO) to: **SHARDS SOFTWARE** Roycraft House, Linton Rd, Barking, Essex. OR Telephone through your Access/Visa order to 01-591 7666.

Selected
Stores



ELECTRON JOYSTICKS

If you have a Plus 1 interface then you have paid a fair amount of money for the A/D converter. Don't waste it by using switched joysticks which will not run programs that need full analogue function. An analogue joystick can easily be made to simulate the "faster" action of a switched joystick if needed, but a switched joystick cannot be made to simulate an analogue one correctly. Reality is analogue. For instance, objects have to be accelerated to a speed, they do not obtain speed instantaneously and the acceleration is proportional to the force applied. There is very little

software around at present that makes full use of analogue joysticks because it requires greater skill to both write the program and use it. As people become bored with the current games, the additional skills needed for the analogue joystick will become more important. ACORN SOFTWARE's Aviator and Snooker are good examples of full analogue use. The co-ordination between hand and eye cannot be achieved if the rate that something moves on the screen is determined in software without regard to the exact position of the joystick or the pressure applied to it or the speed with which it is deflected.



Fully analogue, ACORN compatible, sprung return to centre joysticks to work with your Plus 1 interface or other analogue interfaces. Already in use by tens of thousands of BBC owners, these British made joysticks are fully guaranteed and backed by our years of experience in the video games industry.

▲ **Delta 3b "twin"** - A pair of analogue joysticks wired to a single plug. One joystick has red fire buttons the other green. £19.95

▲ **Delta 3b "single"** - A single joystick but with the fire buttons of two joysticks. £12.00

Coming soon: Conversion software utilities that will make non standard games work on these standard joysticks.

Delta 3b joysticks should be available where you bought your Plus 1 or other analogue interface. If not they can be ordered direct from our factory.

Voltmace Limited
Park Drive
Baldock
Hertfordshire
SG7 6EE
Tel (0462) 894410

Voltmace delta 3b

Prices include VAT and P&P.



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



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.

The Definitive Adventures for 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-
.....qty THE WHEEL OF FORTUNE £9.95 £11.95
.....qty CASTLE FRANKENSTEIN £7.95 £9.95
.....qty THE QUEST FOR THE HOLY GRAIL £7.95 £9.95
.....qty THE KINGDOM OF KLEIN £7.95 £9.95
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 _____

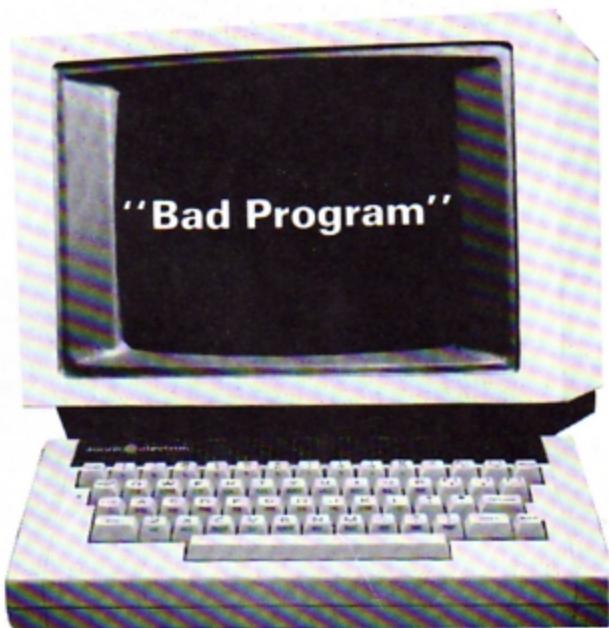
ADDRESS _____

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

NOT SO BAD AS IT SEEMS



**It happens to us all at some time or other! So let
DAVE ROBINSON show you how to cope with
that dreaded error message "Bad Program".**

**IF you've ever had the
dreaded message "Bad
Program" appear on your
Electron's screen and been
frustrated by the apparent
loss of your programs, then
this article is for you.**

This loathsome error message usually occurs when you load a program from cassette, although it can happen if adventurous poking into memory goes astray.

What happened is that the Basic program memory has

somehow been corrupted and the Electron can't deal with what it finds there.

In this article my aim is to show you when and how this condition will be met and, more importantly, what you can do about it.

Before I explain what checks are regularly conducted on your program in memory, I'll try and explain where and how your Basic programs are stored.

As I'm sure you are aware,

the Electron stores Basic programs in a series of addresses in RAM starting at PAGE and ending at TOP-1.

PAGE and TOP are the names given to the address pointers signifying the start and end of the memory space occupied by the program. PAGE is set to address &E00 (3584 decimal) when you first switch on, though you can change this, as we will be doing later.

Each line of every Basic program is stored in a series of numbers. These represent the characters shown on the screen when listing a program. There are also four extra numbers, which will be explained shortly.

The numbers in memory are properly called bytes – a byte being the contents of the addresses referred to, and having a value between 0 and &FF (255 decimal).

Program 1 will display the memory of a Basic program, both the addresses and their contents being shown, with an explanation of each byte.

However, before you type it in, look at Figure 1 for an

Byte	Explanation	Value
1	Start of line	&D(13)
2	Line No. high	0-&7F(127)
3	Line No. low	0-&FF(255)
4	Length of line	5-&FF(255)

Figure 1: The first four bytes

From Page 37

explanation of those four extra bytes in front of every line of your Basic program.

Byte 1 is always set to &D (13 decimal) to signify the start in memory of a line of Basic. Bytes 2 and 3 are the line numbers of your program. The number is held in two bytes as one byte can only hold values up to &FF (255 decimal).

Byte 4 is the total count of bytes in each line of Basic, including the four extra bytes.

The Electron takes care of all this itself. It only concerns you as a programmer when things go wrong or you wish to become ambitious and write utility programs like Program II.

The end in memory of any program is usually signified by having &FF (255 decimal) in byte 2. In practice any number above &7F (127 decimal) will be counted as the end.

Type in Program I carefully. Leave out the ON ERROR until you're sure it works.

Figure II shows a typical display from this program. The four columns are as follows:

Column 1 – address in hex.

Column 2 – contents in hex.

Column 3 – contents in decimal.

Column 4 – explanation of byte.

When you run Program I you'll be examining the program itself. This lets you compare the listing with the actual bytes stored in memory.

To freeze the display at any time press Ctrl and Shift together. To stop scrolling completely, press Esc.

The first four bytes have already been explained. Byte 5 and onwards are coded in memory as one of two things. It could be the Ascii code of the letter typed in – see the User Guide for a full set of Ascii character codes – otherwise it's a token number.

Each Basic keyword, such as PRINT, has its own special code called a token which allows it to only occupy

```

10 REM PROGRAM I           150 IF byte%>1 PRINT"Star
20 MODE6:ON ERROR VDU26:  t"
END                         160 IF byte%>2 PRINT"Line
30 PRINTTAB(3,1)"Address  High"
"TAB(12)"Contents"TAB(24)"C 170 IF byte%>3 PRINT"Line
haracters"                 Low"
40 PRINTTAB(3,2)STRING$(3 180 IF byte%>4 PRINT"Len
33,"=")                     th"
50 VDU28,0,24,39,3         190 IF byte%>4 AND ?addr%
60 addr%>PAGE             >7F PRINT"Token"
70 end%>FALSE             200 IF byte%>4 AND ?addr%
80 @%>8                   >7F PRINT CHR$(?addr%)
90 REPEAT                   210 addr%>addr%+1
100 PRINTTAB(3)STRING$(33 220 IF end%>TRUE THEN byt
,-")                      e%>line%
110 IF ?(addr%+1)>7F THE 230 NEXT
N end%>TRUE               240 UNTIL end%>TRUE
120 line%>?(addr%+3)      250 PRINT"addr%,?addr%,?
130 FOR byte%>1 TO line%  addr%:SPC(3);
140 PRINT"addr%,?addr%,?  260 PRINT"END"
addr%:SPC(3);              270 VDU26
                           280 END

```

Program I

one byte of memory.

These two are easily distinguished by their value. Ascii codes stop at &7F while tokens range from &80 to &FF. Don't worry too much about tokens at this stage.

In case you have not used the ? or the & before, I'll say that the ? is to refer to the contents of an address. The & causes a number to be output in hexadecimal.

Before the Electron will allow you access to the program in memory it will check that each line conforms to the format in Figure I.

The two things that will cause the "Bad Program" error are either the first byte not being equal to &D (13 decimal) or the third byte being equal to zero.

These two checks are repeated for each line of Basic, the length of the line being added to the start address to find the address of the start of the next line.

With the knowledge gained, we turn now to consider how to recover from the bad program error.

What's needed is a short program that will examine the memory area and correct all the faults found. Program II is

program may list but it would be difficult to correct.

Program II uses two procedures to correct them. PROClane counts the number of bytes in a line, replacing any bad bytes with &23 (the Ascii code for £). PROCnumber ensures that all the line numbers are in ascending order.

When you have typed in Program II do not try to run it until you have a copy safely tucked away on tape. As said at the beginning, programs that directly poke into memory can self destruct.

To use Program II to recover a bad program it follows that you must have a Basic program in memory that requires recovering. Just type in a simple program and poke a zero into the third byte with:

?&E03=0

This should effectively produce a bad program. Once you have got a program you want to recover, enter:

*OPT 2.0

to ensure that the Electron accepts all of your program. Now move PAGE to a higher value with a direct command such as:

PAGE = &5C00

Having done this, we're now ready to try to recover the bad program. CHAIN Program II and see if it works. If not, check the listing carefully and try again – not forgetting to reset PAGE.

Assuming all goes well, you

Address	Contents	Characters
E00	D	13 Start
E01	0	0 Line High
E02	A	10 Line Low
E03	F	15 Length
E04	F4	244 Token
E05	20	32
E06	50	80 P
E07	52	82 R
E08	4F	79 0
E09	47	71 G

Figure II: Program memory after PAGE

should have restored your program. It's now up to you.

Carefully go through the listing looking for the £ sign or other mistakes, correcting all you find before trying to run the recovered program.

You'll find that for any program of reasonable size the above recovery method takes several minutes to complete.

There is a way to speed this to less than a second and it also has the advantage that no Basic memory area is used, which means that you don't have to reset PAGE. This panacea is machine code.

When you run Program II it will produce a machine code program that does the same task as Program II. This machine code routine is designed to be stored below PAGE at addresses &D01 onwards - Plus 1 owners beware!

To use Program III type it in, save a copy then run it to assemble the machine code we're going to use to replace Program II.

When Program III has finished it has generated a machine code recovery program which is now lurking below PAGE. A copy of this assembled program is saved by entering:

*SAVE "RECOVER" D01 D90

When you've got this machine code safe on cassette, just load the bad program as before, and enter:

CALL &D01

which activates the recovery program.

To reload the machine code program at any time, type:

*LOAD "RECOVER".

This does not affect any Basic program already in the Electron. This means that when you get the dreaded message you can just:

*LOAD "RECOVER"

which puts the machine code recovery program into the Electron without harming your Basic program. Then enter:

CALL &D01

to set it to work and your program will be recovered.

```
10 REM PROGRAM II          140 PAGE=&E00          250 ?(thisline%+3)=byte%  
20 REM RECOVERY (BASIC)    150 PRINT"Done"  
30 thisline%=&E00:lastli  160 END  
ne%=&E00  
40 end%=&FALSE  
50 REPEAT  
60 PRINT"thisline%"  
70 ?thisline%=&D  
80 IF ?(thisline%+1)>&7F  
THEN end%=&TRUE  
90 PROCline  
100 IF thisline%>&E00 THE  
N PROCnumber  
110 lastline%=&thisline%  
120 thisline%=&thisline%+b  
yte%  
130 UNTIL end%=&TRUE  
R byte%=&FF  
140 PAGE=&E00          150 PRINT"Done"  
160 END  
170 :  
180 DEFPROCline  
190 byte%=&4:endline%=&FALSE  
E  
200 REPEAT  
210 IF ?(thisline%+byte%)  
=&D THEN endline%=&TRUE:GOTO  
240  
220 IF ?(thisline%+byte%)  
&20 THEN ?(thisline%+byte%)  
=&23  
230 byte%=&byte%+1  
240 UNTIL endline%=&TRUE 0  
R byte%=&FF  
250 ?(thisline%+3)=byte%  
260 ENDPROC  
270 :  
280 DEFPROCnumber  
290 IF ?(thisline%+1)>&1  
astline%=&1 ENDPROC  
300 IF ?(thisline%+1)<&1  
astline%=&1 THEN ?(thisline%+1)=?&(lastline%+1)  
310 IF ?(thisline%+2)>&1  
astline%=&2 ENDPROC  
320 ?(thisline%+2)=?&(last  
line%+2)+1  
330 IF ?(thisline%+2)=0 T  
HEN ?(thisline%+1)=?&(thisli  
ne%+1)+1  
340 ENDPROC
```

Program II

```
10 REM PROGRAM III          310 RTS          630 .number  
20 REM TO PRODUCE RECOVE  320 .line          640 TXA  
R (M/C)          330 LDX #1          650 BEQ line  
30 thisline%=&70:lastlin  340 LDY #4          660 LDY #1  
e%=&72          350 .loop          670 LDA (thisline%),Y  
40 FOR IX=0 TO 3 STEP3  360 LDA (thisline%),Y  680 CMP (lastline%),Y  
50 PX=&D01          370 CMP #&D          690 BEQ nextnumber  
60 DPT IX          380 BEQ newline          700 BCS line  
70 LDA #0          390 CMP #&20          710 LDA (lastline%),Y  
80 STA thisline%          400 BCS continue          720 STA (thisline%),Y  
90 STA lastline%          410 LDA #&23          730 .nextnumber  
100 LDA &18          420 STA (thisline%),Y  740 INY  
110 STA thisline%+1          430 .continue          750 LDA (thisline%),Y  
120 STA lastline%+1          440 INY          760 CMP (lastline%),Y  
130 LDY #0          450 BNE loop          770 BEQ add  
140 LDA #&D          460 LDA #&D          780 BCS line  
150 STA (thisline%),Y          470 LDY #&FF          790 .add  
160 LDX #0          480 STA (thisline%),Y  800 LDA (lastline%),Y  
170 .start          490 .newline          810 CLC  
180 LDY #1          500 TYA          820 ADC #1  
190 LDA (thisline%),Y          510 LDY #3          830 STA (thisline%),Y  
200 BEQ next          520 STA (thisline%),Y  840 BNE line  
210 BMI end          530 LDA thisline%+1  850 DEY  
220 BPL number          540 STA lastline%+1  860 LDA (thisline%),Y  
230 .next          550 LDA thisline%          870 CLC  
240 INY          560 STA lastline%          880 ADC #1  
250 LDA (thisline%),Y          570 CLC          890 STA (thisline%),Y  
260 BNE number          580 ADC (thisline%),Y  900 BPL line  
270 .end          590 STA thisline%          910 BMI end  
280 LDY #1          600 BCC start          920 ]  
290 LDA #&FF          610 INC thisline%+1  930 NEXT  
300 STA (thisline%),Y          620 BCS start
```

Program III

ANOTHER INCREDIBLE CHALLENGE FROM BRITAIN'S LEADING SOFTWARE HOUSE

FELIX Meets The

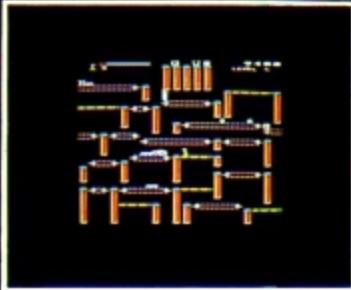
EVIL WEEVILS



Clamber up the pipes, dodging the cascading ball bearings to despatch the Weevils and Super-weevils using your spray can. Features oxygen cylinders, bonus letters, belt lever and panic button.

NEW PRICE! B.B.C. and Electron cassette: £6.95. B.B.C. disk version: £9.95.

(BBC MICRO VERSION)



electron
B.B.C. MICRO

**MICRO
POWER**

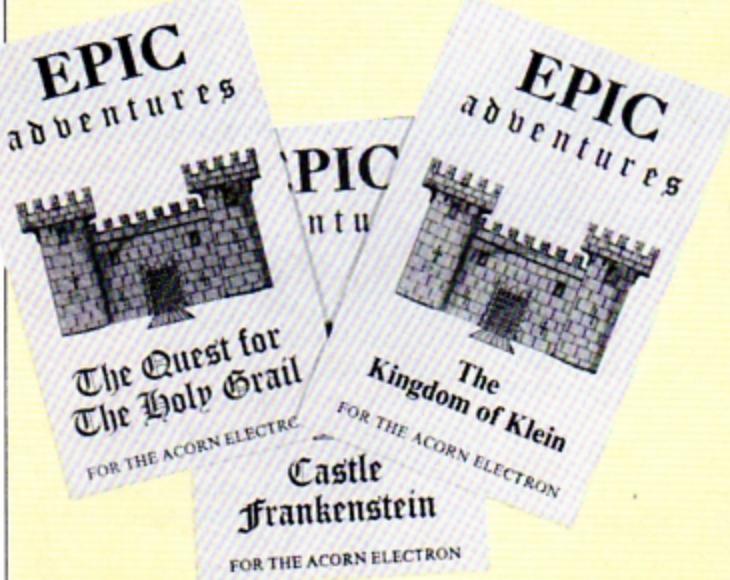
MICRO POWER LTD.,
NORTHWOOD HOUSE, NORTH STREET,
LEEDS LS7 2AA TEL: (0532) 458800
MICRO POWER SOFTWARE IS AVAILABLE FROM
SELECTED BRANCHES OF WH SMITH, BOOTS, JOHN
MENZIES, CO-OP, WOOLWORTHS AND ALL GOOD
DEALERS
AUTHORS! WE PAY 20% ROYALTIES!

"WATCH OUT
FOR THE FANTASTIC
NEW PACKAGING!"



ELECTRON USER
FREE CONTEST

Here's your chance to
WIN
these three top adventures games!



August and September contest winners



By Richard Fereday



By Michael Lythgoe

REMEMBER way back in August when we gave you the chance to win a Sign-point print port by becoming a cartoonist?

We had hundreds of very funny entries, the best Micro Kid strips ever.

Choosing the winners was a difficult task, but

now print ports are on their way to Richard Fereday, of Palmers Green, London and Michael Lythgoe of Widnes, Cheshire.

September's competition had you all trying to

sort out the mischief gremlins had caused to our programs.

Thanks for all your help - First Byte printer interfaces are on their way to Roy Preston of Midlothian and Gary Hugo of Lincoln.

CHRISTMAS is coming and *Electron User* is playing Santa Claus, courtesy of Epic Software.

We've got five sets of Epic's three classic adventures to give away in our free competition - Kingdom of Klein, Quest for the Holy Grail and Castle Frankenstein - all you need for hours of puzzles and pleasure.

And, since it's Christmas, it couldn't be easier to enter. All you have to do is to tell us why you like playing adventures.

The funniest, cleverest, most original or honest reason could make you one of the lucky five.

So, just finish the sentence on the form in not more than 20 words and send it in. The competition will close on Christmas Eve, December 24 and the judge's decision will be final.

Electron User contest entry form

Finish the following sentence in not more than 20 words:

I like playing adventures because ...

Name _____

Address _____

Send to ADVENTURE, Electron User Contest, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.



National Micro

**Everything
on this page
is 5% less
than our
normal price**

This special
offer is
exclusively for
readers of
Electron User
and applies to
mail order
sales only.



We will also send you
FREE membership of
NMC's Computer Club –
enabling you to enjoy
generous discounts on all
your future purchases!

Benefits for Club members
include a big saving of
10% on software and 5%
on hardware purchases
over £25.

Personal shoppers are
welcome at our retail stores:

National Micro Centres,
36 St. Petersgate,
Stockport SK7 5NY.
Tel: 061-429 8080

Wilmslow Micro Centre,
62 Grove Street,
Wilmslow, Cheshire.
Tel: 0625 530891

PRINTER PRICES SLASHED

	Old price	New price
Brother HR5	£170.95	£165.00
RX80	£272.00	£260.00
RX80 FT	£315.00	£295.00
EX43 only	£286.90	£279.00
EX43 +I/F	£394.25	£383.00

MONITORS

You can happily operate your Electron with your domestic TV set. But more and more users are finding that for a really crisp picture you need a special monitor. We offer a monochrome and three colour monitors:

Zenith 12" (green screen)	£81.00
Microvitec (14" colour - low res)	£217.41
Microvitec (14" colour - med res)	£326.66
Microvitec (14" colour - hi res)	£480.70

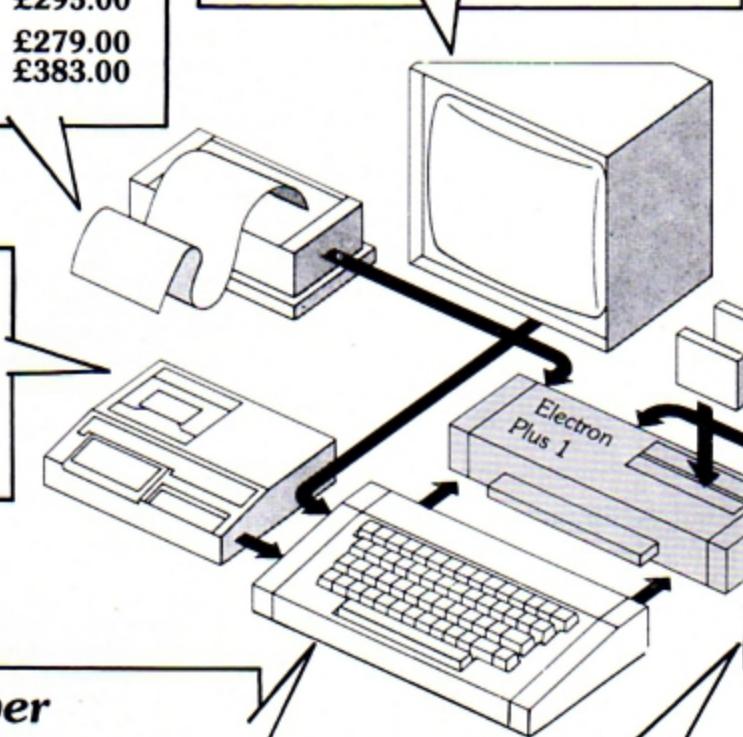
For the best of both worlds there is the 14" Nordmende, which can double as a monitor and normal TV, at a very attractive price

£238.00 (with remote control **£251.00**)

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



A FREE dust cover with every Electron

We have ample stocks of Electrons and can promise mainland delivery within 24 hours of receiving your order. With it comes an introductory cassette of 15 programs, a very comprehensive User Guide, an easy-to-understand DIY book on programming AND a free dust cover with the compliments of National Micro Centres

£189.00

Electron Dust Cover if supplied separately **£2.80**

JOYSTICKS

Use a joystick to play arcade games and watch your score increase dramatically! For serious games a joystick really is a must – and we have two we specially recommend. Both provide twin fire buttons.

Sureshot (self-centering action) .. **£15.67**

Clares (non self-centering) .. pair **£19.50**



These will take Kempston & Sureshot joysticks.

Selling well... First Byte's switched joystick interface

Since it was launched at the Electron & BBC Micro User Show the switched joystick interface from First Byte has been one of our top sellers. This plug-in cartridge takes standard Atari-style joysticks which are much more popular – and cheaper – than analogue joysticks. **£23.70**

Make light work of listings

To save your fingers most of the listings in *Electron User* have been put on tape.

On the December tape:

CHRISTMAS BOX Align the presents logically. **SILLY SANTA** Sort out the muddle. **SNAP** Match the Xmas pictures. **RECOVERY** The Bad Program message tamed. **CAROL** Interrupt driven music. **AUTODATA** A program that grows and grows. **NOTEBOOK** Simple string handling.

On the November 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 tape:

BREAKFREE Classic arcade action. **ALPHASWAP** A logic game to strain your brain. **SOUND GENERATOR** Tame the Electron's sound channels. **MULTICHArcHARACTER 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 tape:

HAUNTED HOUSE Arcade action in the spirit 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 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. **FLYING PIGS** Bacon on the wing.

On the July 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 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 commands without tears. **CRYSTALS** Beautiful graphics. **LASER SHOOT OUT** An intergalactic shooting gallery. **SMILER** Have a nice day!

On the May 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 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. **DOGDUCK** The classic logic game.

On the March tape:

CHICKEN Let dangerous drivers test your nerve. **COFFEE** A tantalising 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 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:

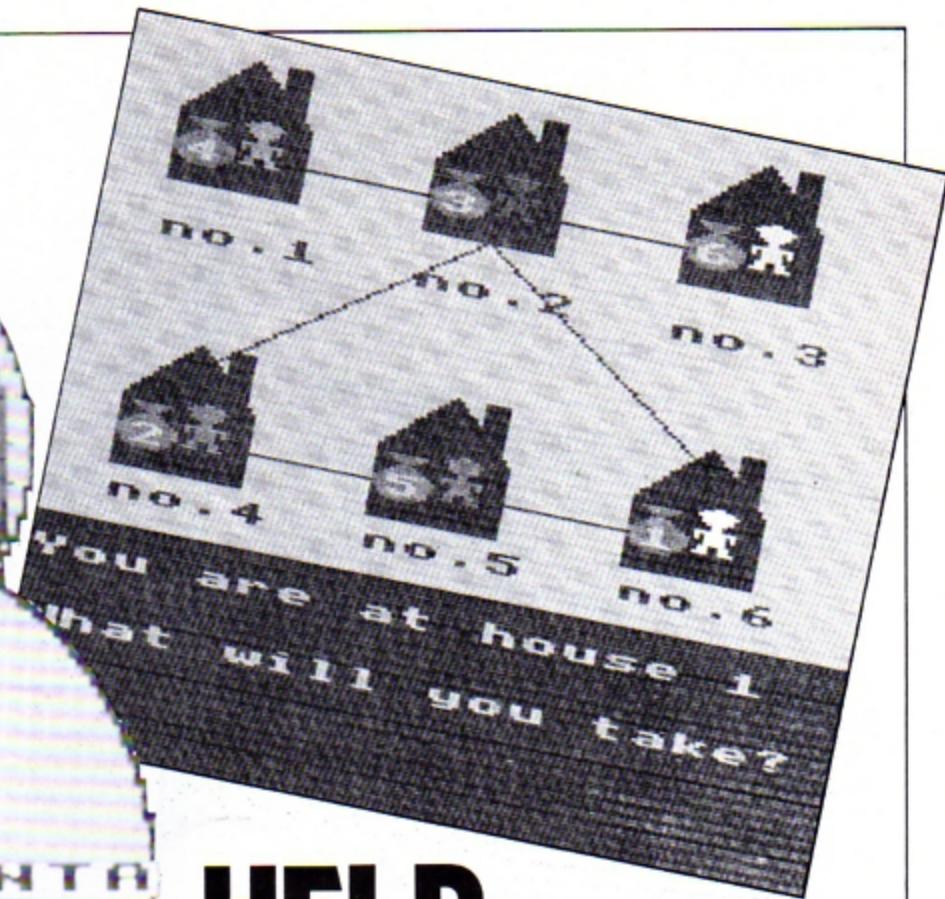
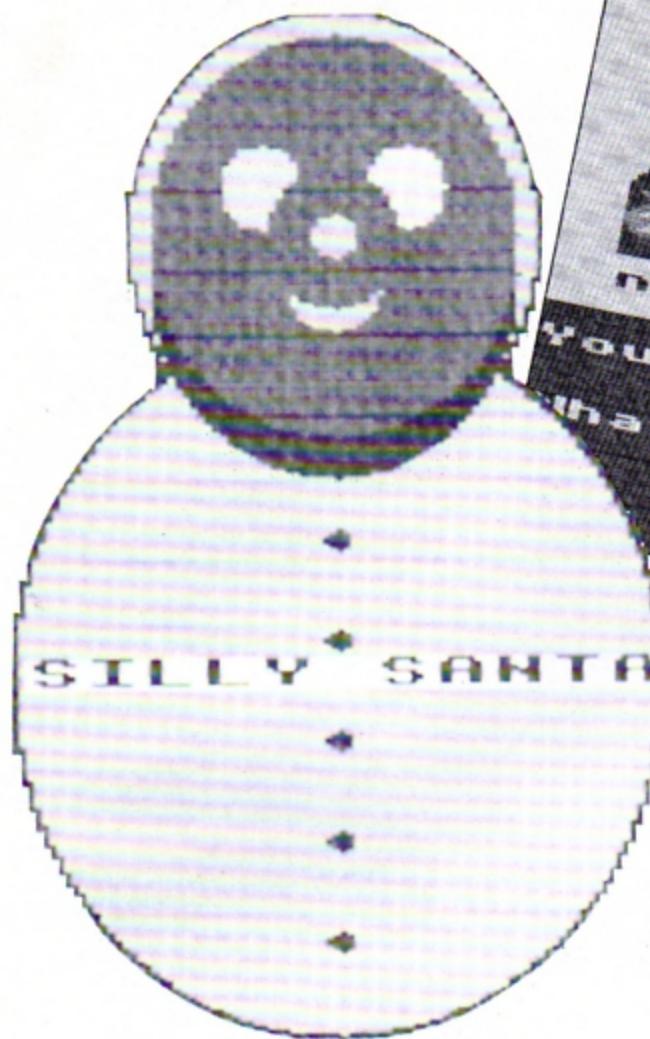
Nine programs from the December issue	£
Nine programs from the November issue	£
Seven programs from the October issue	£
Nine programs from the September issue	£
Fourteen programs from the August issue	£
Ten programs from the July issue	£
Ten programs from the June issue	£
Twelve programs from the May issue	£
Eleven programs from the April issue	£
Twelve programs from the March issue	£
Nine programs from the February issue	£
26 programs from the introductory issues	£

Name I enclose the sum of £

Address POST TO: Tape Offer, *Electron User*, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.



A delightful festive frolic from ROG FROST



HELP TO SORT OUT SANTA

PROCEDURES

PROCcredits	Prints title.
PROCcircle (X,Y,R,C)	Used for starting picture. Draws circle centred at (X,Y), radius R in colour C.
PROConstruct	Prints instruction on screen.
PROCchoose	Decides which sack is randomly left at which house.
PROCsetup	Sets up text and graphics windows, selects colours, defines characters and some variables.
PROChouse (X,Y)	Draws houses at (X,Y).
PROCsanta (X,Y,C)	Draws a man in each house at position (X,Y) and logical colour C.
PROCsack	Positions sacks in the correct places.
PROCdecide	The playing part of the game in which you make decisions.
PROCremove	Deals with the removal of sacks from the houses.
PROCdrop	Deals with leaving sacks at houses.
PROCfail	Displays fail message.
PROCsucces	Displays success message.
PROGgoodtune	Plays a happy Christmas song.
PROCbadtune	Plays a bad Christmas song.
PROChall	Prints the fastest time.

SANTA has a bug in his operating system this year. He has delivered sacks of presents to the six people in Disc Drive, but put them in the wrong houses.

You, the Electron Elf, must sort out the muddle before everyone wakes up on Christmas morning.

You do this by moving from house to house collecting or dropping sacks. You can carry up to three sacks at a time.

If you succeed, then you get

another go, but the time you are allowed is the time you took in your first go.

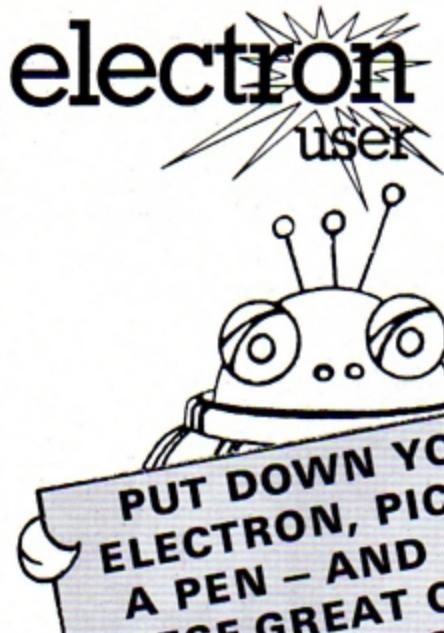
If you fail, you are given 20 seconds longer for your next turn. Appropriate tunes are played for success or failure and a record is kept of the fastest time.

The first part of the program uses a series of PROCcircles to produce a picture. The game itself follows and instructions are included within the program.

VARIABLES

T%	Time allowed to complete task.	C\$	Shuffled B\$
M%	Best time so far.	carry\$	Sacks being carried at present.
z%	Number of present house.	take\$	Sack you hope to take.
newz%	Number of house you hope to visit.	leave\$	Sack you hope to drop.
A\$(N)	Initial sack numbers.	drop\$	Sack you have dropped.
B\$	Numbers 123456		

Full listing starts on Page 58

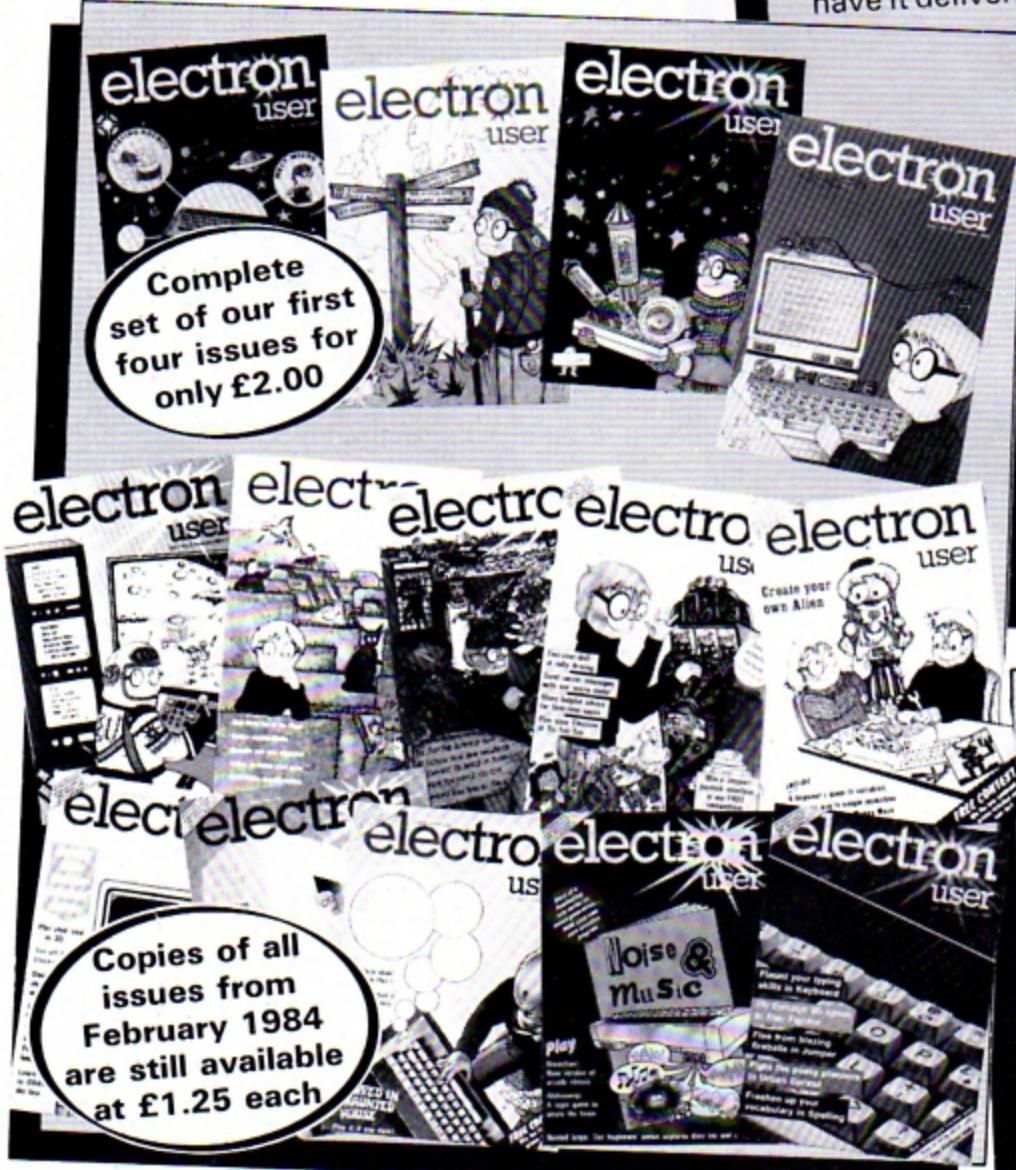


Be one of the first to get each issue

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.

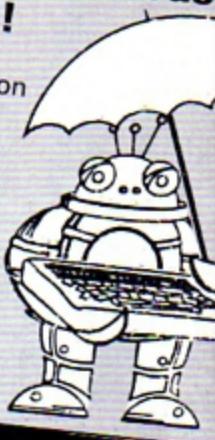
You can buy *Electron User* from your local newsagent or station bookstall. Or you can take out a 12 months subscription and have it delivered to you by post.



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!

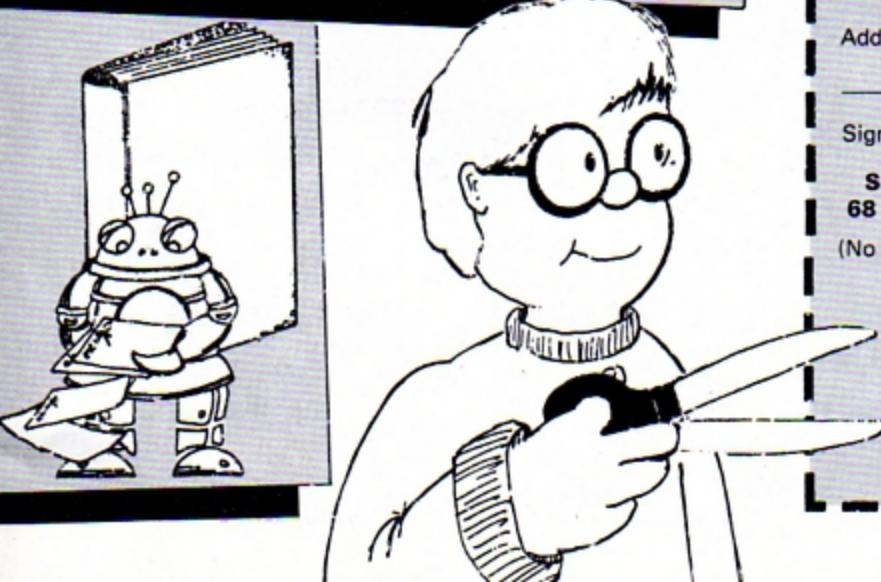
Cassette tapes of Electron User programs

Save typing in programs from *Electron User* by sending for these program-packed tapes.

£3.75 each



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 December 28.

Please enter number required in box £ p

Electron User annual subscription	UK £12	<input type="text"/>
	EIRE £13 (IR £16)	<input type="text"/>
	Overseas (Surface) £20	<input type="text"/>
	Overseas (Airmail) £40	<input type="text"/>

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 <input type="text"/>
	March <input type="text"/>
	April <input type="text"/>
	May <input type="text"/>
	June <input type="text"/>
	July <input type="text"/>
	August <input type="text"/>
	Sept <input type="text"/>
	Oct <input type="text"/>
	Nov <input type="text"/>
	Dec <input type="text"/>
	TOTAL <input type="text"/>

Airmail prices on application

Electron User tapes	26 introductory programs <input type="text"/>
	Lunar Lander February <input type="text"/>
	Chicken March <input type="text"/>
	Spacehike April <input type="text"/>
	Rally Driver May <input type="text"/>
	Money Maze June <input type="text"/>
	Golf July <input type="text"/>
	Castles of Sand August <input type="text"/>
	Haunted House Sept <input type="text"/>
	Breakfree Oct <input type="text"/>
	Star Fighter Nov <input type="text"/>
	Christmas Box Dec <input type="text"/>
	TOTAL <input type="text"/>

**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

Telephone:
061-480 0171
24 hours

Don't forget to quote your credit card number
and give your full address

MOUSER!

MOUSER was written to help children practice their skills with compass directions.

A mouse is hidden on a 12 x 12 grid labelled with

the coordinates A1-L12. Your task is to send your cat to find the mouse but you only have five goes.

Each time you make a mistake the micro will tell

PROCEDURES

PROCinstructions	Gives instructions.
PROClplace	Hide the mouse.
PROCGuess	Guess location.
PROClose	Lose game.
PROCmouse	Draws mouse.
PROCcat	Draws cat.
PROCman	Draws man.
PROCmove	Chooses direction you can move.
PROCwin	Find mouse.
PROCsouth	Go South.
PROCnorth	Go North.

you a compass direction to take.

You must tell the micro the coordinate of the square you want to search. Incorrect coordinates will not be accepted.

I would recommend you change line 100 to *FX229,1 to disable Esc when you are sure that the program has been fully debugged.

If you want to make the game easier by having more turns, just change the value of S% in lines 80, 1200 and 1660 to give more than five goes.

```
10 ON ERROR GOTO 1330
20 REM ** Mouser **
30 REM ** an educational
game for the BBC and Elect
ron computers **
40 REM ** Steve W. Lucas
**
```

```
50 REM ** (C) ELECTRON U
SER
```

```
60 MODE1
```

```
70 *KEY10 OLDIM RUNIM
```

```
80 S%5
```

```
90 VDU23,1,0;0;0;0
```

```
100 REM ** change this li
ne to *FX229,1 when you hav
e fully debugged the progra
m to disable escape key **
110 *FX210,0
```

```
120REM ** define characte
rs **
130VDU23,239,255,255,255,
```

```
255,255,255,255,255
140VDU23,240,192,128,156,
```

```
191,255,255,65,113
150VDU23,241,0,96,112,88,
```

```
204,252,0,192
160VDU23,242,48,32,32,32,
```

```
32,32,31,15
170VDU23,243,1,3,7,7,3,1,
```

```
255,255
180VDU23,244,0,128,192,24
0,176,240,248,136
```

```
190VDU23,245,7,3,1,1,0,1,
```

```
1,0
200VDU23,246,255,255,253,
```

```
129,128,129,193,0
210VDU23,247,134,128,128,
```

```
128,128,128,192,0
220 VDU23,248,7,4,13,12,4
```

```
,6,1,1
230VDU23,249,240,16,88,15
```

```
2,16,48,192,192
240VDU23,250,0,3,6,12,9,9
```

```
,9,9
250VDU23,251,128,224,176,
```

```
216,72,72,72,72
260VDU23,252,29,21,21,3,3
```

```
,3,6,4
270VDU23,253,92,212,212,2
```

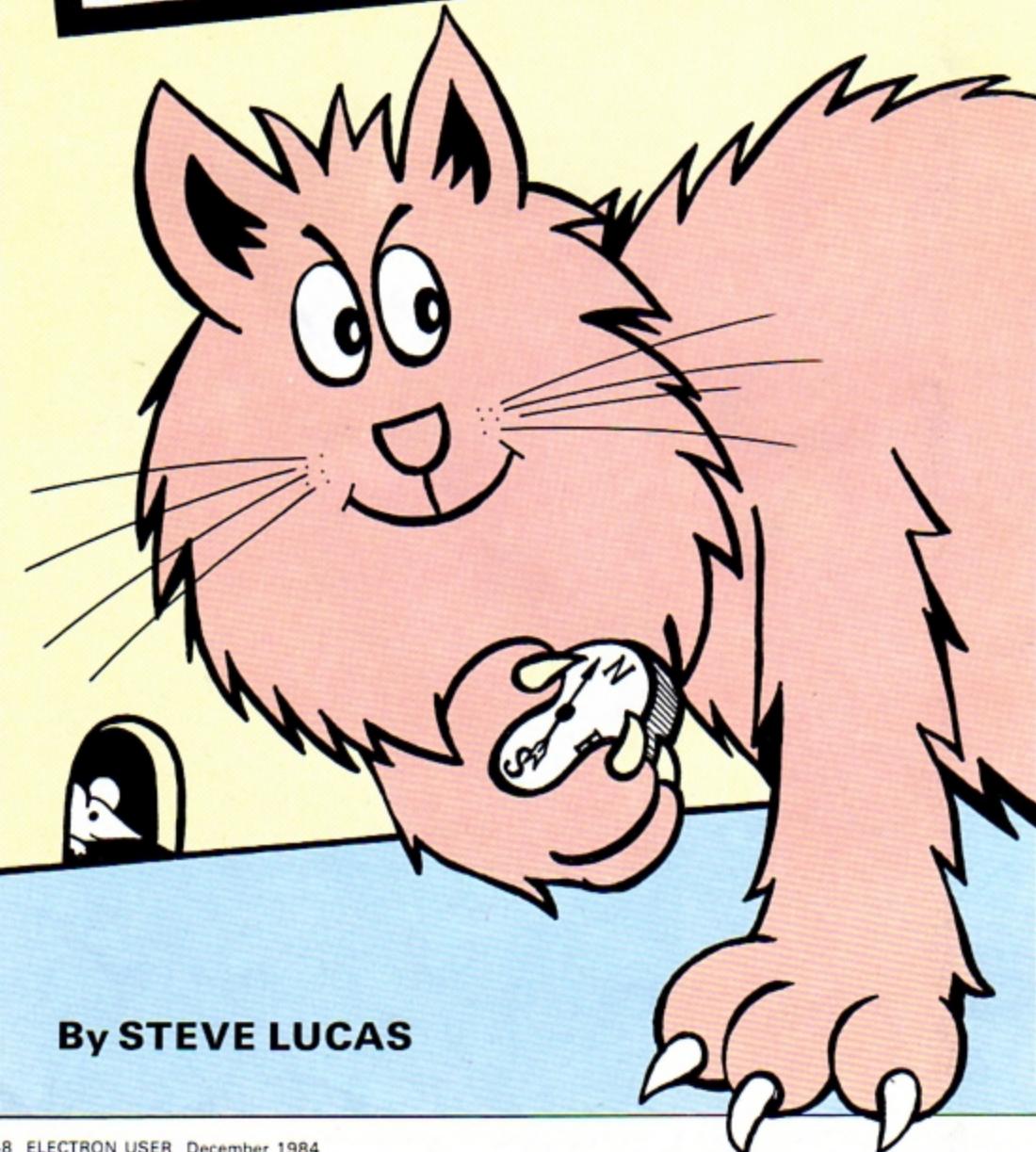
```
24,224,224,48,16
280VDU23,254,4,6,4,28,28,
```

```
,0,0,0
290VDU23,255,16,48,16,28,
```

```
28,0,0,0
300 VDU19,0,3,0,0,0,19,1,
```

```
6,0,0,0,19,2,1,0,0,0,19,3,4
,0,0,0
310 REM define envelopes
```

```
320 ENVELOPE 1,1,1,0,0,20
```



By STEVE LUCAS

0,0,0,126,0,0,-126,126,126
 330 ENVELOPE 2,1,-7,7,0,1
 0,10,0,126,0,0,-126,126,126
 340 ENVELOPE 3,1,36,-36,0
 20,20,0,126,0,0,-126,126,1
 26
 350 PROCinstructions
 360 T% = 4
 370 GCOL0,1:FOR Y% = 0 TO 8
 00 STEP 150
 380 FOR XX = 0 TO 800 STEP
 150
 390 MOVEXX,75+Y%:MOVEXX,Y
 %:PLOT85,XX+75,75+Y%:PLOT85
 ,XX+75,Y%:MOVEXX+75,Y%+75:M
 MOVEXX+75,Y%+150:PLOT85,XX+1
 50,Y%+75:PLOT85,XX+150,Y%+1
 50:NEXTXX,Y%
 400 GCOL0,2:MOVE0,0:DRAW9
 00,0:DRAW900,900:DRAW0,900:
 DRAW0,0
 410 VDU5:GCOL0,2:FORXX=1T
 012:MOVE650,75*XX-30:PRINTX
 X:NEXT:FORXX=1T012:MOVE75*X
 X-50,940:PRINTCHR\$(64+X%):N
 EXT:VDU4
 420 MOVE0,970:DRAW970,970
 :DRAW970,0
 430 MOVE1100,600:DRAW1100
 ,700:PLOT85,1116,600:PLOT85
 ,1116,700:MOVE1108,730:MOVE
 1085,700:PLOT85,1134,700
 440 VDU5:MOVE1100,760:PRI
 NT"":VDU4
 450 PRINTTAB(8)"M O U S E
 R"
 460 VDU28,31,31,39,20
 470 REM ** main game **
 480 PROCplace
 490 REPEAT
 500 PROCguess
 510 S% = S%-1
 520 T% = S%-1
 530 UNTIL S% = 0
 540 PROClose
 550 END
 560 DEFPROCplace
 570 A% = RND(12):B% = RND(12)
 580 ENDPROC
 590 END
 600 DEFPROCmouse(XX,Y%,Z%
)
 610 VDU5:MOVEXX,Y%:GCOL0,
 ZX:VDU240,241,4:ENDPROC
 620 DEFPROCcat(XX,Y%,Z%)
 630 VDU5:MOVEXX,Y%:GCOL0,
 ZX:VDU242,243,244,10,8,8,8,
 245,246,247,4:ENDPROC

640 DEFPROCman(XX,Y%,Z%)
 650 VDU5:MOVEXX,Y%:GCOL0,
 ZX:VDU248,249,10,8,8,250,25
 1,10,8,8,252,253,10,8,8,254
 ,255,4:ENDPROC
 660 DEFPROCmove
 670 PROCcat(CX*75-75,DX*7
 5-5,2)
 680 SOUND 1,2,160,10
 690 ENDPROC
 700 DEFPROCguess
 710 CLS
 720 VDU 23,1,0;0;0;0;
 730 COLOUR 3
 740 PRINT"Enter guess
 now :"
 750 REPEAT
 760 A\$ = GET\$: PRINTTAB(2,4)
 :A\$
 770 CX = ASC(A\$)
 780 UNTIL CX > 64 AND CX < 77
 790 CX = CX - 64
 800 REPEAT
 810 PRINTTAB(3,4) SPC 6
 820 INPUT TAB(3,4) "A\$
 830 DX = VAL(A\$) : IF DX > 13 OR
 DX < 1 THEN VDU7
 840 UNTIL DX > 0 AND DX < 13
 850 VDU5:GCOL0,0: MOVE 99
 0,920 :PRINT STRING\$(9,CHR\$
 (239)):VDU4
 860 PROCmove
 870 b\$ = ""
 880 IF AX = CX AND DX = BX THEN
 EN PROCwin
 890 IF DX < BX THEN PROCnor
 th ELSE PROCsouth
 900 VDU5:MOVE990,1000:PRI
 NT"G a o d b y e.":VDU4
 910 MOVE 990,960 :PRINT"q
 o :"
 920 MOVE 990,920 :PRINT b\$
 930 MOVE 973,520 :GCOL0,0
 :PRINTSTRING\$(8,CHR\$239);:G
 COL0,2
 940 MOVE 970,550 :PRINT"t
 urns"
 950 MOVE 970,520 :PRINT"1
 eft :"
 960 MOVE 900,520 :PRINT T%
 970 VDU4
 980 ENDPROC
 990 DEFPROCnorth
 1000 b\$ = "North"
 1010 IF CX < AX THEN b\$ = b\$ +
 "east" ELSE IF CX > AX THEN b\$
 = b\$ + "west"
 1020 ENDPROC
 1030 DEFPROCsouth

1040 IF BX < DX THEN b\$ = "Sou
 th"
 1050 IF AX > CX THEN b\$ = b\$ +
 "east" ELSE IF CX > AX THEN b\$
 = b\$ + "west"
 1060 ENDPROC
 1070 DEFPROCwin
 1080 CLG
 1090 RESTORE
 1100 FOR X = 1 TO 6
 1110 READ A%, B%, C%
 1120 PROCman(A%, B%, C%)
 1130 NEXT
 1140 SOUND 1,2,255,50
 1150 DATA 1000,1000,2,1000
 ,150,2,150,1000,2,150,150,2
 ,400,700,3,800,400,3
 1160 COLOUR 3
 1170 VDU5
 1180 MOVE 500,650 :PRINT "W e
 i l D o n e"
 1190 MOVE 150,350 :PRINT "Yo
 u found the mouse"
 1200 S% = 5
 1210 GCOL0,2
 1220 MOVE 300,100 :PRINT "An
 other game <Y/N> ?"
 1230 VDU4
 1240 VDU23,1,0;0;0;0;
 1250 REPEAT
 1260 yes\$ = GET\$
 1270 UNTIL INSTR("YNyn", ye
 s\$)
 1280 IF yes\$ = "Y" OR yes\$ =
 "N" THEN CLG: GOTO 360
 1290 CLG
 1300 VDU5:MOVE 200,600:PRI
 NT"G a o d b y e.":VDU4
 1310 END
 1320 ENDPROC
 1330 MODE 6
 1340 PRINTTAB(5,15) "Error
 "; ER; " in line number "; ER
 L
 1350 END
 1360 DEFPROCinstructions
 1370 CLS
 1380 PRINTTAB(15,2) "M O U
 S E R"
 1390 COLOUR 2
 1400 PRINT "'SPC(10); <C>
 S.W. Lucas 1984"
 1410 COLOUR 3
 1420 PRINT "'This is a ga
 me in which you must try to
 find the mouse which is hi
 dden on the board."
 1430 COLOUR 2
 1440 PRINT "'You must use

your cat to search for the
 mouse by telling me the co
 ordinate of the square yo
 u want to search."
 1450 COLOUR 3
 1460 PRINT "'I will then t
 ell you which direction to
 go in. You will have only
 FIVE turns to find the mous
 e in!"
 1470 REM ** you can alter
 the number of turns by alte
 ring the value of S% at the
 start of program
 1480 COLOUR 2
 1490 PRINT "'Do you want s
 ound <Y>es or <N>o ?"
 1500 REPEAT
 1510 A\$ = GET\$
 1520 UNTIL A\$ = "Y" OR A\$ = "N
 "
 1530 IF A\$ = "N" THEN #FX210
 ,1
 1540 PRINT "'Press <Space
 Bar> to start the game"
 1550 REPEAT UNTIL GET = 32
 1560 CLS:ENDPROC
 1570 DEFPROCclose
 1580 CLS:PRINT "You""didn't
 find it!"
 1590 A\$ = "ABCDEFHGIJKL"
 1600 B\$ = MID\$(A\$, A%, 1)
 1610 COLOUR 2
 1620 PRINT "It was""in "; B
 \$; B%
 1630 PROCmouse(A%*75-60, B%
 *75-25, 3)
 1640 SOUND 1,3,160,50
 1650 COLOUR 3
 1660 S% = 5; T% = 4
 1670 PRINT "Another""Game
 ""<Y/N>?"
 1680 VDU23,1,0;0;0;0;
 1690 REPEAT
 1700 yes\$ = GET\$
 1710 UNTIL INSTR("YNyn", ye
 s\$)
 1720 IF yes\$ = "Y" OR yes\$ =
 "N" THEN CLG: GOTO 360
 1730 CLG
 1740 VDU5:MOVE 200,600:PRI
 NT"G a o d b y e.":VDU4
 1750 END

This listing is included in
 this month's cassette
 tape offer. See order
 form on Page 47.

Three programs everyone's talking about!

Play the most exciting,
challenging Olympic
simulation of them all!



MICRO OLYMPICS is riding high in the charts – and no wonder. You are challenged to beat the world record in 11 realistic track and field events, with the packed stadium cheering you on to victory – or defeat! Just like the real thing, you'll want to keep on improving your sporting skill until you're up there among the greats!

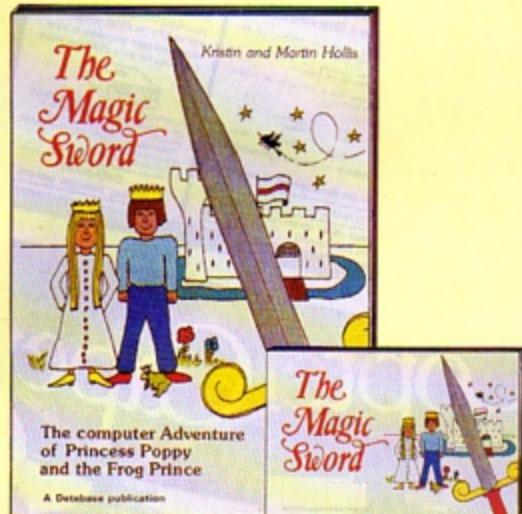


Turn your home micro into
a true office workhorse!

MINI OFFICE is the most incredibly priced introduction to business computing ever offered. The package contains four full-scale programs – word processor, database manager, spreadsheet and graphics. They include advanced features not yet available on programs costing many times as much. Plus a concise 32-page how-to-do-it booklet.

Let your children embark on a
magical micro adventure!

THE MAGIC SWORD breaks new ground in programming for the young – a complete adventure on cassette accompanied by a 48-page full colour storybook. Stunning sound and colourful graphics bring a completely new dimension to the classic computer adventure. And there's a very helpful animated compass to help you find your way around!



Send
for
them
today

Please tick copies required

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

I wish to pay by

Access

Visa

No. _____ Expiry date _____

Name _____

Address _____

Signed _____

Post to: Database Publications, Europa House,
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

	Micro Olympics	Mini Office	Magic Sword
Spectrum cassette	£5.95 <input type="checkbox"/>	N/A	N/A
Commodore cassette	£5.95 <input type="checkbox"/>	N/A	N/A
BBC 'B' cassette	£5.95 <input type="checkbox"/>	£5.95 <input type="checkbox"/>	£8.95 <input type="checkbox"/>
BBC 40 track disc	£7.95 <input type="checkbox"/>	£7.95 <input type="checkbox"/>	N/A
BBC 80 track disc	£7.95 <input type="checkbox"/>	£7.95 <input type="checkbox"/>	N/A
Electron cassette	£5.95 <input type="checkbox"/>	£5.95 <input type="checkbox"/>	£8.95 <input type="checkbox"/>



THE Editor stopped me.
"Waddilove", he slurred.
"Christmas is coming. Do
something".

After spending a week or so racking my brains trying to think of an original Christmas program I came up with Xmas Carol. It simply wishes everyone a Merry Christmas to the accompaniment of a few festive tunes.

You may be wondering where the originality is in that.

Well, it's not what it does, it's how it does it.

The program demonstrates the use of interrupts by playing carols while text is printed in different directions and with various degrees of rotation.

One of the most advanced features of the Electron, and the BBC Micro as well, is the extensive use of interrupts to

control many of the background operating system tasks.

An interrupt is a signal sent to the microprocessor telling it to stop what it is doing and switch its attention to something else.

When it has finished this task it returns to whatever it was doing and carries on as if

nothing had happened.

These background tasks include updating the clock, used by the pseudo variable TIME, processing envelopes, and maintaining the many input and output buffers and queues.

Interrupts give the impression that the Electron is capable of doing more than one thing at a time by repeatedly

VARIABLES

I%, J%	Used as loop counters.
a\$, b\$	Messages to be printed.
theta	Angle of letter in radians.
angle	Angle of letter in radians.
x%, y%	Position of letter.
X%, Y%	Position of letter.
size	Size of letter.
sin, cos	Work out SIN and COS function to save time.
letter\$	Letter to be printed.
pointer	Pointer to next note.
sound	Parameter block for Ossword call.

From Page 51

switching rapidly between tasks.

Interrupts must not have any effect on the interrupted program. If any of the processor's registers or flags are altered by the interrupt routine, then it will get in a terrible muddle when it returns, probably with disastrous consequences.

Acorn have thoughtfully provided the user with an easy to use, pre-packaged interrupt facility. Every 10 milliseconds an interrupt is generated by one of the timers inside the ULA to transfer program control to routines to deal with the background work.

In the process of carrying out this background work a number of events may be generated, such as the interval timer crossing zero. An event handling routine can be written by the user to which control is passed, when the appropriate event has been detected by the operating system.

The operating system detects all events but ignores them if they have not been enabled with a *FX14 command. If an event has been enabled then program execution indirections via the event vector at &220. (See Page 242 of the User Guide.)

The machine code routine in Xmas Carol is called 50 times a second, coincident with the start of vertical synchronisation of the screen.

PROCEDURES

PROCstore_data	Disable start of screen synchronisation display event. Switch off ADC channels. Read and store data for tunes. Define function keys.
PROCassemble	Assemble machine code routine to play carols. Set event vector to point to code. Enable event.
PROLetters	Print the message.
PROPrint	Print a character at a given angle and position.

point I%, J% when rotated through an angle theta becomes:

I%*COS (theta)-J%*SIN (theta), I%*SIN(theta)+J%*COS (theta).

An allowance has to be made for the odd shaped pixels in Mode 5 but it is fairly straightforward.

To print text round in a circle you just move to a point on the circumference x%, y% which is radius*SIN (theta), radius*COS (theta). To make the text stand outwards the angle of rotation is -theta.

The machine code is placed at &900 and the data for the

carols at &A00, which are buffers used by the cassette system.

It is safe to type in and run programs while the carols are playing, but loading or saving a program may corrupt the code so it is best to disable the routine with *FX13.4.

Don't just use the program as it is. Try experimenting with different tunes. (Lines 220, 230 and 790 must be set to the number of items in the data statement, 254 maximum.)

Alter the messages printed and see what happens if the size of the letters is changed.

Will it run in Mode 1? Try it and see. Alter it if necessary.

Xmas Carol listing

```

10REM **XMAS PROGRAM**
20REM *For Electron/Micr
o User*
30REM *By R.A.Waddilove*
40REM *Happy Christmas*
500N ERROR GOTO 1320
60PROCstore_data
70PROCassemble
80MODE 5
90PROLetters
100TIME=0:REPEAT UNTIL TI
ME>500
110PRINT TAB(5,31);"Press
Space";
120*FX21,0
130REPEAT UNTIL GET=32
140MODE 6
150PRINT**"Function Keys
:-**"f1 - LIST current pro
gram."**"f2 - delete curren
t program."**"f3 - end the
carols."
160PRINT**"Programs can b

```

e typed in and run while the carols continue to play. To LOAD or SAVE a program press f3."

170END
180
190DEF PROCstore_data
200*FX13.4
210*FX16,0
220?&70=244
230FOR I=244 TO 1 STEP-1
240READ JX:IX?&A00=JX
250NEXT
260*KEY1 "LIST07IMINLIST:
M"

270*KEY2 "NEWIM"
280*KEY3 "*FX13.4IM"
290*KEY10 "?&220=0:&221=
9:&FX14,4IM"
300ENDPROC
310
320REM pitch,length,...



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.

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.

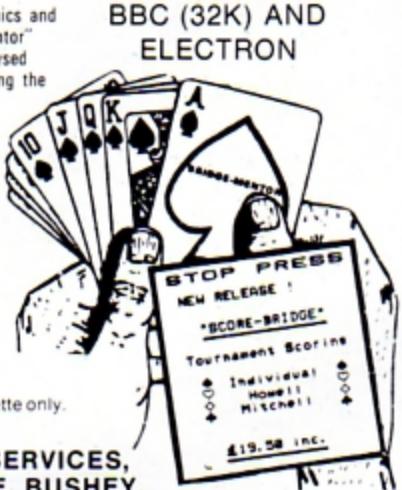
Compile special teaching material.

£9.50 inc.

Cheques to: - * All cassette only.

DEPT: EU12

MINIC BUSINESS SERVICES,
12 WOBURN CLOSE, BUSHEY,
HERTFORDSHIRE. WD2 3XA.



Easy Programming for the Electron Eric Deeson £5.95

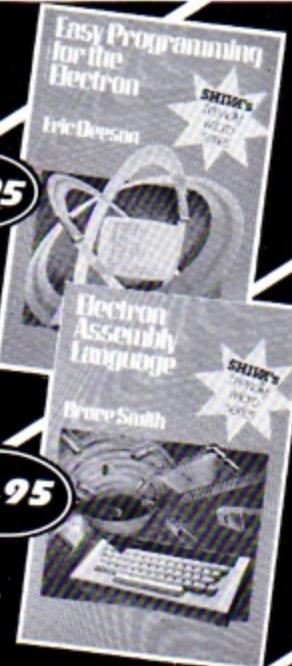
For newcomers to computing, the author starts at basics and instructs the beginner in a lighthearted but informative manner in the intricacies of the micro-Electron User.

£5.95

Electron Assembly Language Bruce Smith £7.95

This book will open a whole new world to the Basic programmer - Electron User, by gently introducing assembly language programming. The hidden depths of your Electron can be tapped when you learn all about the use of the machine's built-in assembler, and full explanations of the operating system. You will be amazed by the results!

£7.95



Shiva titles are available from branches of Boots, Menzies, W.H.Smith, bookshops, computer shops and department stores. In case of difficulty, please write to:
SHIVA PUBLISHING LTD, FREEPOST, 64 Welsh Row, Nantwich, Cheshire CW5 5BR.
Telephone orders welcome: 0270 628272 (24 hrs/7 days) Telex: 367258 (GASEOP)

ORDER FORM

Qty	Title	Price	Enclose a cheque for £.....
—	Easy Programming for the Electron	£5.95	made payable to Shiva Publishing Ltd.
—	Electron Assembly Language	£7.95	For payment by Access/Barclaycard/American Express:
Signature		Card No.	
Please send full catalogue of computer books and software		<input type="checkbox"/>	
Name			
Address			

EU/12

What the Acorn Electron has been waiting for!

Latest version of Forth for the Electron (Not re-hashed Forth 79 Code)

Unique Stack Display Utility

16k Eprom type 27128



Available as a 'bare' ROM or a ROM Cartridge for the 'Plus I' interface

Multi-tasking operating system for Real Time use.

Here's another first from Skywave Software. A Forth Eprom for the Acorn Electron which can Multi-task. It's called Multi-Forth.

It's the same Forth that has already revolutionised the BBC Micro and, since it follows hot on the heels of the ZX81-Forth ROM and Spectrum Forth-I/O Cartridge, you can probably guess that David Husband is the genius behind it.

Multi-Forth 83 is a 16k Eprom type 27128 which sits sideways in the ROM area along with any other ROMs in use. It then allows a number of Forth programs to run simultaneously and transparently of each other, placing each task in a queue, up to a maximum of twenty-eight!

Multi-Forth 83 is also compatible with the MOS and specially vectored to enable a system to be reconfigured. It contains a Standard 6502 Assembler, a Standard Screen Editor and a unique Stack Display Utility, too.

At a later date a Cartridge version for the Acorn 'Plus I' will be available, but for now Multi-Forth 83 is sold as a 'Bare' ROM which means an interface is needed for the Standard Acorn Electron.

This unique Eprom comes with an extensive Manual and, at £45+VAT it is superb value. Order it using the coupon, adding £2.30 p&p (£5 for Europe, £10 outside) or, for more information, simply tick that box instead. Either way, you'll be one step ahead of the competition.

Please send me Multi-Forth 83 for Acorn Electron. £45+VAT. Cheques to Skywave Software.		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Name		<input type="checkbox"/>		
Address		<input type="checkbox"/>		
Post code		<input type="checkbox"/>		
SUBJECT TO AVAILABILITY FOR 1000 ORDERS				
Send to Skywave Software, 73 Curzon Road, Bournemouth, BH1 4PW, Dorset, England. Tel: (0202) 302385				
Please send me more information:				
<input type="checkbox"/> Multi-Forth 83 (Electron)				
<input type="checkbox"/> ZX81-Forth ROM				
<input type="checkbox"/> Spectrum Forth-I/O Cartridge				

Skywave SOFTWARE

MULTI-FORTH 83 FOR THE ACORN ELECTRON

Xmas Carol listing

From Page 52

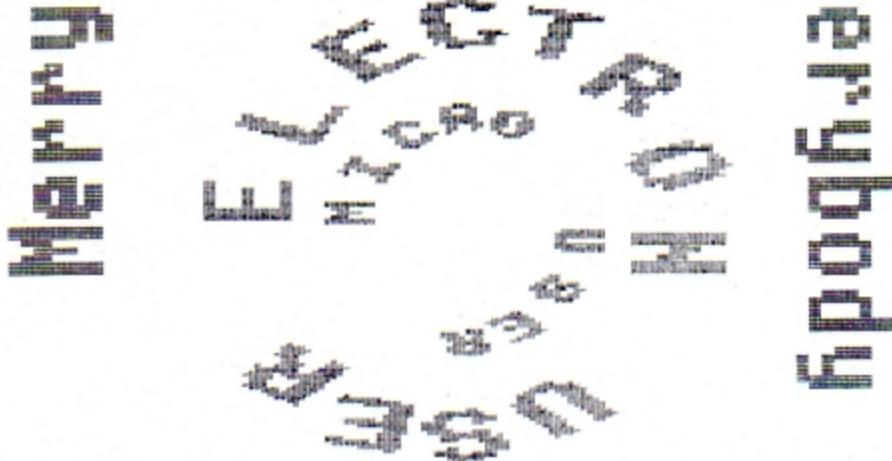
```

330
340REM **Silent Night**
350DATA 80,15, 88,5, 80,1
0, 68,30, 80,15, 88,5, 80,1
0, 68,30, 108,30, 96,30, 10
0,30, 80,30, 88,30, 100,15,
96,5, 88,10, 80,15, 88,5,8
0,10, 68,30
360
370REM **We Three Kings**
380DATA 96,18, 88,9, 80,1
8, 68,9, 76,9, 80,9, 76,9,
68,27, 96,18, 88,9, 80,18,
68,9, 76,9, 80,9, 76,9, 68,
27, 80,18, 0,0, 80,9, 88,18,
0,0, 88,9, 96,18, 0,0, 96
,9, 108,9, 100,9, 96,9, 88,
9, 96,9, 88,9, 80,18, 76,9,
68,40
390
400REM **Jingle Bells**
410DATA 96,5, 0,0, 96,5,
0,0, 96,10, 0,0, 96,5, 0,0,
96,5, 0,0, 96,10, 0,0, 96,
5, 108,5, 80,7, 88,3, 96,20
, 100,5, 0,0, 100,5, 0,0, 1
00,7, 0,0, 100,5, 96,5, 0,0
, 96,5, 0,0, 96,3, 0,0, 96,
3, 108,5, 0,0, 108,5, 100,5
, 88,5, 80,40
420
430REM **Away In A Manger
**
440DATA 52,10, 72,10, 0,0
, 72,10, 80,5, 88,5, 72,10,
0,0, 72,10, 88,5, 92,5, 10
0,10, 0,0, 100,10, 108,10,
92,20, 80,5, 88,5, 92,10, 0
,0, 92,10, 100,10, 88,10, 0
,0, 88,10, 72,5, 88,5, 80,
10, 60,10, 68,10,72,40
450
460DEF PROCassemble
470pointer=&70
480sound=&71
490!sound=1
500sound!2=-15
510osword=&FFF1
520osbyte=&FFF4
530?&220=&00?&221=&09
540FOR pass=0 TO 2 STEP 2
550P%=&900
560! OPT pass
570PHP:PHA \save regi
sters

```

Christmas

From



```

580TXA:PHA
590TYA:PHA
600LDA #&80
610LDX #&FA
620LDY #&FF
630JSR osbyte \space in
sound buffer?
640CPX #10
650BMI end \if not en
ough
660LDY pointer
670LDA &A00,Y \get pitch
680STA sound+4
690DEY
700LDA &A00,Y \get lengt
h
710STA sound+6
720LDA #7
730LDX #sound MOD256
740LDY #sound DIV256
750JSR osword \play note
760DEC pointer
770DEC pointer
780BNE end
790LDA #244:STA pointer
800.end
810PLA:TAY \restore r
egisters
820PLA:TAX
830PLA:PLP
840RTS
850
860NEXT
870+FX14,4
880ENDPROC
890
900DEF PROCletters
910VDU 23,1,0;0;0;0;
920VDU 19,3,0;0;
930GCOL 0,2
940a$="Merry"
950FOR IX=1 TO 5
960PROCprint(MID$(a$,IX,1
),2,PI/2,128,IX*64+330)
970NEXT
980a$="Christmas"
990b$="Everybody"
1000FOR IX=1 TO 9
1010PROCprint(MID$(a$,IX,1
),2,0,IX*128-80,900)
1020PROCprint(MID$(b$,IX,1
),2,-PI/2,1144,900-IX*64)
1030NEXT
1040GCOL 0,1
1050a$="ELECTRON USER"
1060b$="MICRO USER "
1070FOR IX=1 TO 13
1080theta=RAD(IX*(360/14)-
(90+360/14))
1090x$=639+200*SIN(theta)
1100y$=452+200*COS(theta)
1110PROCprint(MID$(a$,IX,1
),2,-theta,x$,y$)
1120x$=639+110*SIN(theta)
1130y$=452+110*COS(theta)
1140PROCprint(MID$(b$,IX,1
),1,-theta,x$,y$)
1150NEXT
1160VDU 19,3,4;0;
1170PRINT TAB(8,7);"From"
1180ENDPROC
1190
1200DEF PROCprint(letter$,
size,angle,x$,y$)
1210LOCAL IX,JX
1220PRINT TAB(0,31);letter
$;
1230cos=size*COS(angle)
1240sin=2*size*SIN(angle)
1250FOR IX=0 TO 64 STEP 4
1260FOR JX=0 TO 32
1270IF POINT(IX,JX) PLOT 6
9,IX+IX*cos-JX*sin,YX+(IX*s
in/4)+JX*cos
1280NEXT
1290NEXT
1300ENDPROC
1310
1320REM ** error **
1330MODE 6:VDU 19,0,4;0;14
1340REPORT:PRINT" at line
";ERL

```

This listing is included in this month's cassette tape offer. See order form on Page 47.

Christmas Snap listing

From Page 16

```

10 REM ***CHRISTMAS**SNA
***  

20 REM ***BY P.TAYLER***  

22 REM**ELECTRON USER***  

23 REM**CHRISTMAS 1984**  

25 PROCinstructions  

30 MODE2  

32 REPEAT  

35 VDU23,1,0;0;0;0;  

40 count=FALSE  

50 COLOUR135:CLS  

60 PROCproper_jack  

70 REPEAT:PROCrandomcolour:  

r:PROCjack_in_the_box(11,30)  

:PROCcheck:UNTIL count=TRUE  

E  

90 count=FALSE  

100 PROCproper_tree  

110 REPEAT:PROCrandomcolour:  

r:PROCchristmas_tree(11,30)  

:PROCcheck:UNTIL count=TRUE  

130 count=FALSE  

140 PROCproper_santa  

150 REPEAT:PROCrandomcolour:  

r:PROCsanta(11,30):PROCcheck:  

UNTIL count=TRUE  

170 UNTIL FALSE  

180 DEFPROCjack_in_the_box:  

=(X%,Y%)  

190 VDU28,X%,Y%,X%+7,Y%-  

9,17,135:CLS  

200 COLOURA:PRINTSPC(0);  

210 COLOURA:PRINTSPC(1)::C  

OLOURB:PRINTSPC(7);  

220 COLOURA:PRINTSPC(1)::C  

OLOURB:PRINTSPC(2)::COLOURC  

:PRINTSPC(3)::COLOURB:PRINT  

SPC(2);  

230 COLOURA:PRINTSPC(1)::C  

OLOURB:PRINTSPC(1)::COLOURC  

:PRINTSPC(1)::COLOURD:PRINT  

SPC(1)::COLOURC:PRINTSPC(1)  

::COLOURD:PRINTSPC(1)::COLO  

URC:PRINTSPC(1)::COLOURB:PR  

INTSPC(1);  

240 COLOURA:PRINTSPC(1)::C  

OLOURB:PRINTSPC(1)::COLOURC  

:PRINTSPC(1)::COLOURD:PRINT  

SPC(1)::COLOURC:PRINTSPC(1)  

::COLOURD:PRINTSPC(1)::COLO  

URC:PRINTSPC(1)::COLOURB:PR  

INTSPC(1);  

250 COLOURA:PRINTSPC(2)::C  

OLOURB:PRINTSPC(1)::COLOURC  

:PRINTSPC(1)::COLOURB:PRINTSP  

C(1)::COLOURC:PRINTSPC(1)::C  

OLOURB:PRINTSPC(1);

```

PROCEDURES

PROCinstructions

PROCrandomcolour

PROCproper_jack
PROCproper_santa
PROCproper_tree

PROCchristmas_tree
PROCjack_in_the_box
PROCsanta

PROCtune
PROCgoodwenceslas
PROCjinglebells
PROCdeckthehalls

Sets up the introductory and instruction screens in Mode 6.

Gives values to the variables *B-H* (used in PROC's to set up the pictures). It decides a random value for *B*, and the others take their values from that.

These set up a text window on the left of the screen, then the variables *A-H* are given their correct values, then it jumps into the corresponding general PROC (for example, PROCchristmas_tree).

These accept the randomly generated values for the variables *B-H*, set up a text window on the right of the screen, and then draw the corresponding picture.

PROCtune uses a random feature to decide which of the four carols is to be played. It then calls up the appropriate PROC.

VARIABLES

count

Used to determine the point at which we jump out of the REPEAT...UNTIL loop. Initially count is FALSE, but if a correct match is made, it becomes TRUE.

Used in PROCchristmas_tree and so on to set up the text window.

X% and Y%

Used for the random colouring of the right-hand pictures. Actually, they are not truly random, as a variable *random%* becomes *B*, and then the other variables take their values in order from this one.

The variable set during the initialisation to decide how long should be allowed to the child to respond.

B to H
random%

Also used as a keyboard variable using *G=GET*.

Used as a counter in the loops playing the carols. *A*, *P* and *D* (as appropriate) are used to read amplitude, pitch and duration.

timeallowed

G

note

270 COLOURA:PRINTSPC(2)::COLOURA:PRINTSPC(3)::COLOURB:PRINTSPC(1)::COLOURC:PRINTSPC(1)::COLOURD:PRINTSPC(1)::COLOURE:PRINTSPC(2)::COLOURF:PRINTSPC(1)::COLOURG:PRINTSPC(1)::COLOURH:PRINTSPC(1)::COLOURI:PRINTSPC(1)::COLOURJ:PRINTSPC(1)::COLOURK:PRINTSPC(1)::COLOURL:PRINTSPC(1)::COLOURM:PRINTSPC(1)::COLOURN:PRINTSPC(1)::COLOURO:PRINTSPC(1)::COLOURP:PRINTSPC(1)::COLOURQ:PRINTSPC(1)::COLOURR:PRINTSPC(1)::COLOURS:PRINTSPC(1)::COLOURT:PRINTSPC(1)::COLOURU:PRINTSPC(1)::COLOURV:PRINTSPC(1)::COLOURW:PRINTSPC(1)::COLOURX:PRINTSPC(1)::COLOURY:PRINTSPC(1)::COLOURZ:PRINTSPC(1)::COLOURA:PRINTSPC(2)::COLOURB:PRINTSPC(3)::COLOURC:PRINTSPC(4)::COLOURD:PRINTSPC(5)::COLOURE:PRINTSPC(6)::COLOURF:PRINTSPC(7)::COLOURG:PRINTSPC(8)::COLOURH:PRINTSPC(9)::COLOURI:PRINTSPC(10)::COLOURJ:PRINTSPC(11)::COLOURK:PRINTSPC(12)::COLOURL:PRINTSPC(13)::COLOURM:PRINTSPC(14)::COLOURN:PRINTSPC(15)::COLOURO:PRINTSPC(16)::COLOURP:PRINTSPC(17)::COLOURQ:PRINTSPC(18)::COLOURR:PRINTSPC(19)::COLOURS:PRINTSPC(20)::COLOURT:PRINTSPC(21)::COLOURU:PRINTSPC(22)::COLOURV:PRINTSPC(23)::COLOURW:PRINTSPC(24)::COLOURX:PRINTSPC(25)::COLOURY:PRINTSPC(26)::COLOURZ:PRINTSPC(27)::COLOURA:PRINTSPC(28)::COLOURB:PRINTSPC(29)::COLOURC:PRINTSPC(30)::COLOURD:PRINTSPC(31)::COLOURE:PRINTSPC(32)::COLOURF:PRINTSPC(33)::COLOURG:PRINTSPC(34)::COLOURH:PRINTSPC(35)::COLOURI:PRINTSPC(36)::COLOURJ:PRINTSPC(37)::COLOURK:PRINTSPC(38)::COLOURL:PRINTSPC(39)::COLOURM:PRINTSPC(40)::COLOURN:PRINTSPC(41)::COLOURO:PRINTSPC(42)::COLOURP:PRINTSPC(43)::COLOURQ:PRINTSPC(44)::COLOURR:PRINTSPC(45)::COLOURS:PRINTSPC(46)::COLOURT:PRINTSPC(47)::COLOURU:PRINTSPC(48)::COLOURV:PRINTSPC(49)::COLOURW:PRINTSPC(50)::COLOURX:PRINTSPC(51)::COLOURY:PRINTSPC(52)::COLOURZ:PRINTSPC(53)::COLOURA:PRINTSPC(54)::COLOURB:PRINTSPC(55)::COLOURC:PRINTSPC(56)::COLOURD:PRINTSPC(57)::COLOURE:PRINTSPC(58)::COLOURF:PRINTSPC(59)::COLOURG:PRINTSPC(60)::COLOURH:PRINTSPC(61)::COLOURI:PRINTSPC(62)::COLOURJ:PRINTSPC(63)::COLOURK:PRINTSPC(64)::COLOURL:PRINTSPC(65)::COLOURM:PRINTSPC(66)::COLOURN:PRINTSPC(67)::COLOURO:PRINTSPC(68)::COLOURP:PRINTSPC(69)::COLOURQ:PRINTSPC(70)::COLOURR:PRINTSPC(71)::COLOURS:PRINTSPC(72)::COLOURT:PRINTSPC(73)::COLOURU:PRINTSPC(74)::COLOURV:PRINTSPC(75)::COLOURW:PRINTSPC(76)::COLOURX:PRINTSPC(77)::COLOURY:PRINTSPC(78)::COLOURZ:PRINTSPC(79)::COLOURA:PRINTSPC(80)::COLOURB:PRINTSPC(81)::COLOURC:PRINTSPC(82)::COLOURD:PRINTSPC(83)::COLOURE:PRINTSPC(84)::COLOURF:PRINTSPC(85)::COLOURG:PRINTSPC(86)::COLOURH:PRINTSPC(87)::COLOURI:PRINTSPC(88)::COLOURJ:PRINTSPC(89)::COLOURK:PRINTSPC(90)::COLOURL:PRINTSPC(91)::COLOURM:PRINTSPC(92)::COLOURN:PRINTSPC(93)::COLOURO:PRINTSPC(94)::COLOURP:PRINTSPC(95)::COLOURQ:PRINTSPC(96)::COLOURR:PRINTSPC(97)::COLOURS:PRINTSPC(98)::COLOURT:PRINTSPC(99)::COLOURU:PRINTSPC(100)::COLOURV:PRINTSPC(101)::COLOURW:PRINTSPC(102)::COLOURX:PRINTSPC(103)::COLOURY:PRINTSPC(104)::COLOURZ:PRINTSPC(105)::COLOURA:PRINTSPC(106)::COLOURB:PRINTSPC(107)::COLOURC:PRINTSPC(108)::COLOURD:PRINTSPC(109)::COLOURE:PRINTSPC(110)::COLOURF:PRINTSPC(111)::COLOURG:PRINTSPC(112)::COLOURH:PRINTSPC(113)::COLOURI:PRINTSPC(114)::COLOURJ:PRINTSPC(115)::COLOURK:PRINTSPC(116)::COLOURL:PRINTSPC(117)::COLOURM:PRINTSPC(118)::COLOURN:PRINTSPC(119)::COLOURO:PRINTSPC(120)::COLOURP:PRINTSPC(121)::COLOURQ:PRINTSPC(122)::COLOURR:PRINTSPC(123)::COLOURS:PRINTSPC(124)::COLOURT:PRINTSPC(125)::COLOURU:PRINTSPC(126)::COLOURV:PRINTSPC(127)::COLOURW:PRINTSPC(128)::COLOURX:PRINTSPC(129)::COLOURY:PRINTSPC(130)::COLOURZ:PRINTSPC(131)::COLOURA:PRINTSPC(132)::COLOURB:PRINTSPC(133)::COLOURC:PRINTSPC(134)::COLOURD:PRINTSPC(135)::COLOURE:PRINTSPC(136)::COLOURF:PRINTSPC(137)::COLOURG:PRINTSPC(138)::COLOURH:PRINTSPC(139)::COLOURI:PRINTSPC(140)::COLOURJ:PRINTSPC(141)::COLOURK:PRINTSPC(142)::COLOURL:PRINTSPC(143)::COLOURM:PRINTSPC(144)::COLOURN:PRINTSPC(145)::COLOURO:PRINTSPC(146)::COLOURP:PRINTSPC(147)::COLOURQ:PRINTSPC(148)::COLOURR:PRINTSPC(149)::COLOURS:PRINTSPC(150)::COLOURT:PRINTSPC(151)::COLOURU:PRINTSPC(152)::COLOURV:PRINTSPC(153)::COLOURW:PRINTSPC(154)::COLOURX:PRINTSPC(155)::COLOURY:PRINTSPC(156)::COLOURZ:PRINTSPC(157)::COLOURA:PRINTSPC(158)::COLOURB:PRINTSPC(159)::COLOURC:PRINTSPC(160)::COLOURD:PRINTSPC(161)::COLOURE:PRINTSPC(162)::COLOURF:PRINTSPC(163)::COLOURG:PRINTSPC(164)::COLOURH:PRINTSPC(165)::COLOURI:PRINTSPC(166)::COLOURJ:PRINTSPC(167)::COLOURK:PRINTSPC(168)::COLOURL:PRINTSPC(169)::COLOURM:PRINTSPC(170)::COLOURN:PRINTSPC(171)::COLOURO:PRINTSPC(172)::COLOURP:PRINTSPC(173)::COLOURQ:PRINTSPC(174)::COLOURR:PRINTSPC(175)::COLOURS:PRINTSPC(176)::COLOURT:PRINTSPC(177)::COLOURU:PRINTSPC(178)::COLOURV:PRINTSPC(179)::COLOURW:PRINTSPC(180)::COLOURX:PRINTSPC(181)::COLOURY:PRINTSPC(182)::COLOURZ:PRINTSPC(183)::COLOURA:PRINTSPC(184)::COLOURB:PRINTSPC(185)::COLOURC:PRINTSPC(186)::COLOURD:PRINTSPC(187)::COLOURE:PRINTSPC(188)::COLOURF:PRINTSPC(189)::COLOURG:PRINTSPC(190)::COLOURH:PRINTSPC(191)::COLOURI:PRINTSPC(192)::COLOURJ:PRINTSPC(193)::COLOURK:PRINTSPC(194)::COLOURL:PRINTSPC(195)::COLOURM:PRINTSPC(196)::COLOURN:PRINTSPC(197)::COLOURO:PRINTSPC(198)::COLOURP:PRINTSPC(199)::COLOURQ:PRINTSPC(200)::COLOURR:PRINTSPC(201)::COLOURS:PRINTSPC(202)::COLOURT:PRINTSPC(203)::COLOURU:PRINTSPC(204)::COLOURV:PRINTSPC(205)::COLOURW:PRINTSPC(206)::COLOURX:PRINTSPC(207)::COLOURY:PRINTSPC(208)::COLOURZ:PRINTSPC(209)::COLOURA:PRINTSPC(210)::COLOURB:PRINTSPC(211)::COLOURC:PRINTSPC(212)::COLOURD:PRINTSPC(213)::COLOURE:PRINTSPC(214)::COLOURF:PRINTSPC(215)::COLOURG:PRINTSPC(216)::COLOURH:PRINTSPC(217)::COLOURI:PRINTSPC(218)::COLOURJ:PRINTSPC(219)::COLOURK:PRINTSPC(220)::COLOURL:PRINTSPC(221)::COLOURM:PRINTSPC(222)::COLOURN:PRINTSPC(223)::COLOURO:PRINTSPC(224)::COLOURP:PRINTSPC(225)::COLOURQ:PRINTSPC(226)::COLOURR:PRINTSPC(227)::COLOURS:PRINTSPC(228)::COLOURT:PRINTSPC(229)::COLOURU:PRINTSPC(230)::COLOURV:PRINTSPC(231)::COLOURW:PRINTSPC(232)::COLOURX:PRINTSPC(233)::COLOURY:PRINTSPC(234)::COLOURZ:PRINTSPC(235)::COLOURA:PRINTSPC(236)::COLOURB:PRINTSPC(237)::COLOURC:PRINTSPC(238)::COLOURD:PRINTSPC(239)::COLOURE:PRINTSPC(240)::COLOURF:PRINTSPC(241)::COLOURG:PRINTSPC(242)::COLOURH:PRINTSPC(243)::COLOURI:PRINTSPC(244)::COLOURJ:PRINTSPC(245)::COLOURK:PRINTSPC(246)::COLOURL:PRINTSPC(247)::COLOURM:PRINTSPC(248)::COLOURN:PRINTSPC(249)::COLOURO:PRINTSPC(250)::COLOURP:PRINTSPC(251)::COLOURQ:PRINTSPC(252)::COLOURR:PRINTSPC(253)::COLOURS:PRINTSPC(254)::COLOURT:PRINTSPC(255)::COLOURU:PRINTSPC(256)::COLOURV:PRINTSPC(257)::COLOURW:PRINTSPC(258)::COLOURX:PRINTSPC(259)::COLOURY:PRINTSPC(260)::COLOURZ:PRINTSPC(261)::COLOURA:PRINTSPC(262)::COLOURB:PRINTSPC(263)::COLOURC:PRINTSPC(264)::COLOURD:PRINTSPC(265)::COLOURE:PRINTSPC(266)::COLOURF:PRINTSPC(267)::COLOURG:PRINTSPC(268)::COLOURH:PRINTSPC(269)::COLOURI:PRINTSPC(270)::COLOURJ:PRINTSPC(271)::COLOURK:PRINTSPC(272)::COLOURL:PRINTSPC(273)::COLOURM:PRINTSPC(274)::COLOURN:PRINTSPC(275)::COLOURO:PRINTSPC(276)::COLOURP:PRINTSPC(277)::COLOURQ:PRINTSPC(278)::COLOURR:PRINTSPC(279)::COLOURS:PRINTSPC(280)::COLOURT:PRINTSPC(281)::COLOURU:PRINTSPC(282)::COLOURV:PRINTSPC(283)::COLOURW:PRINTSPC(284)::COLOURX:PRINTSPC(285)::COLOURY:PRINTSPC(286)::COLOURZ:PRINTSPC(287)::COLOURA:PRINTSPC(288)::COLOURB:PRINTSPC(289)::COLOURC:PRINTSPC(290)::COLOURD:PRINTSPC(291)::COLOURE:PRINTSPC(292)::COLOURF:PRINTSPC(293)::COLOURG:PRINTSPC(294)::COLOURH:PRINTSPC(295)::COLOURI:PRINTSPC(296)::COLOURJ:PRINTSPC(297)::COLOURK:PRINTSPC(298)::COLOURL:PRINTSPC(299)::COLOURM:PRINTSPC(300)::COLOURN:PRINTSPC(301)::COLOURO:PRINTSPC(302)::COLOURP:PRINTSPC(303)::COLOURQ:PRINTSPC(304)::COLOURR:PRINTSPC(305)::COLOURS:PRINTSPC(306)::COLOURT:PRINTSPC(307)::COLOURU:PRINTSPC(308)::COLOURV:PRINTSPC(309)::COLOURW:PRINTSPC(310)::COLOURX:PRINTSPC(311)::COLOURY:PRINTSPC(312)::COLOURZ:PRINTSPC(313)::COLOURA:PRINTSPC(314)::COLOURB:PRINTSPC(315)::COLOURC:PRINTSPC(316)::COLOURD:PRINTSPC(317)::COLOURE:PRINTSPC(318)::COLOURF:PRINTSPC(319)::COLOURG:PRINTSPC(320)::COLOURH:PRINTSPC(321)::COLOURI:PRINTSPC(322)::COLOURJ:PRINTSPC(323)::COLOURK:PRINTSPC(324)::COLOURL:PRINTSPC(325)::COLOURM:PRINTSPC(326)::COLOURN:PRINTSPC(327)::COLOURO:PRINTSPC(328)::COLOURP:PRINTSPC(329)::COLOURQ:PRINTSPC(330)::COLOURR:PRINTSPC(331)::COLOURS:PRINTSPC(332)::COLOURT:PRINTSPC(333)::COLOURU:PRINTSPC(334)::COLOURV:PRINTSPC(335)::COLOURW:PRINTSPC(336)::COLOURX:PRINTSPC(337)::COLOURY:PRINTSPC(338)::COLOURZ:PRINTSPC(339)::COLOURA:PRINTSPC(340)::COLOURB:PRINTSPC(341)::COLOURC:PRINTSPC(342)::COLOURD:PRINTSPC(343)::COLOURE:PRINTSPC(344)::COLOURF:PRINTSPC(345)::COLOURG:PRINTSPC(346)::COLOURH:PRINTSPC(347)::COLOURI:PRINTSPC(348)::COLOURJ:PRINTSPC(349)::COLOURK:PRINTSPC(350)::COLOURL:PRINTSPC(351)::COLOURM:PRINTSPC(352)::COLOURN:PRINTSPC(353)::COLOURO:PRINTSPC(354)::COLOURP:PRINTSPC(355)::COLOURQ:PRINTSPC(356)::COLOURR:PRINTSPC(357)::COLOURS:PRINTSPC(358)::COLOURT:PRINTSPC(359)::COLOURU:PRINTSPC(360)::COLOURV:PRINTSPC(361)::COLOURW:PRINTSPC(362)::COLOURX:PRINTSPC(363)::COLOURY:PRINTSPC(364)::COLOURZ:PRINTSPC(365)::COLOURA:PRINTSPC(366)::COLOURB:PRINTSPC(367)::COLOURC:PRINTSPC(368)::COLOURD:PRINTSPC(369)::COLOURE:PRINTSPC(370)::COLOURF:PRINTSPC(371)::COLOURG:PRINTSPC(372)::COLOURH:PRINTSPC(373)::COLOURI:PRINTSPC(374)::COLOURJ:PRINTSPC(375)::COLOURK:PRINTSPC(376)::COLOURL:PRINTSPC(377)::COLOURM:PRINTSPC(378)::COLOURN:PRINTSPC(379)::COLOURO:PRINTSPC(380)::COLOURP:PRINTSPC(381)::COLOURQ:PRINTSPC(382)::COLOURR:PRINTSPC(383)::COLOURS:PRINTSPC(384)::COLOURT:PRINTSPC(385)::COLOURU:PRINTSPC(386)::COLOURV:PRINTSPC(387)::COLOURW:PRINTSPC(388)::COLOURX:PRINTSPC(389)::COLOURY:PRINTSPC(390)::COLOURZ:PRINTSPC(391)::COLOURA:PRINTSPC(392)::COLOURB:PRINTSPC(393)::COLOURC:PRINTSPC(394)::COLOURD:PRINTSPC(395)::COLOURE:PRINTSPC(396)::COLOURF:PRINTSPC(397)::COLOURG:PRINTSPC(398)::COLOURH:PRINTSPC(399)::COLOURI:PRINTSPC(400)::COLOURJ:PRINTSPC(401)::COLOURK:PRINTSPC(402)::COLOURL:PRINTSPC(403)::COLOURM:PRINTSPC(404)::COLOURN:PRINTSPC(405)::COLOURO:PRINTSPC(406)::COLOURP:PRINTSPC(407)::COLOURQ:PRINTSPC(408)::COLOURR:PRINTSPC(409)::COLOURS:PRINTSPC(410)::COLOURT:PRINTSPC(411)::COLOURU:PRINTSPC(412)::COLOURV:PRINTSPC(413)::COLOURW:PRINTSPC(414)::COLOURX:PRINTSPC(415)::COLOURY:PRINTSPC(416)::COLOURZ:PRINTSPC(417)::COLOURA:PRINTSPC(418)::COLOURB:PRINTSPC(419)::COLOURC:PRINTSPC(420)::COLOURD:PRINTSPC(421)::COLOURE:PRINTSPC(422)::COLOURF:PRINTSPC(423)::COLOURG:PRINTSPC(424)::COLOURH:PRINTSPC(425)::COLOURI:PRINTSPC(426)::COLOURJ:PRINTSPC(427)::COLOURK:PRINTSPC(428)::COLOURL:PRINTSPC(429)::COLOURM:PRINTSPC(430)::COLOURN:PRINTSPC(431)::COLOURO:PRINTSPC(432)::COLOURP:PRINTSPC(433)::COLOURQ:PRINTSPC(434)::COLOURR:PRINTSPC(435)::COLOURS:PRINTSPC(436)::COLOURT:PRINTSPC(437)::COLOURU:PRINTSPC(438)::COLOURV:PRINTSPC(439)::COLOURW:PRINTSPC(440)::COLOURX:PRINTSPC(441)::COLOURY:PRINTSPC(442)::COLOURZ:PRINTSPC(443)::COLOURA:PRINTSPC(444)::COLOURB:PRINTSPC(445)::COLOURC:PRINTSPC(446)::COLOURD:PRINTSPC(447)::COLOURE:PRINTSPC(448)::COLOURF:PRINTSPC(449)::COLOURG:PRINTSPC(450)::COLOURH:PRINTSPC(451)::COLOURI:PRINTSPC(452)::COLOURJ:PRINTSPC(453)::COLOURK:PRINTSPC(454)::COLOURL:PRINTSPC(455)::COLOURM:PRINTSPC(456)::COLOURN:PRINTSPC(457)::COLOURO:PRINTSPC(458)::COLOURP:PRINTSPC(459)::COLOURQ:PRINTSPC(460)::COLOURR:PRINTSPC(461)::COLOURS:PRINTSPC(462)::COLOURT:PRINTSPC(463)::COLOURU:PRINTSPC(464)::COLOURV:PRINTSPC(465)::COLOURW:PRINTSPC(466)::COLOURX:PRINTSPC(467)::COLOURY:PRINTSPC(468)::COLOURZ:PRINTSPC(469)::COLOURA:PRINTSPC(470)::COLOURB:PRINTSPC(471)::COLOURC:PRINTSPC(472)::COLOURD:PRINTSPC(473)::COLOURE:PRINTSPC(474)::COLOURF:PRINTSPC(475)::COLOURG:PRINTSPC(476)::COLOURH:PRINTSPC(477)::COLOURI:PRINTSPC(478)::COLOURJ:PRINTSPC(479)::COLOURK:PRINTSPC(480)::COLOURL:PRINTSPC(481)::COLOURM:PRINTSPC(482)::COLOURN:PRINTSPC(483)::COLOURO:PRINTSPC(484)::COLOURP:PRINTSPC(485)::COLOURQ:PRINTSPC(486)::COLOURR:PRINTSPC(487)::COLOURS:PRINTSPC(488)::COLOURT:PRINTSPC(489)::COLOURU:PRINTSPC(490)::COLOURV:PRINTSPC(491)::COLOURW:PRINTSPC(492)::COLOURX:PRINTSPC(493)::COLOURY:PRINTSPC(494)::COLOURZ:PRINTSPC(495)::COLOURA:PRINTSPC(496)::COLOURB:PRINTSPC(497)::COLOURC:PRINTSPC(498)::COLOURD:PRINTSPC(499)::COLOURE:PRINTSPC(500)::COLOURF:PRINTSPC(501)::COLOURG:PRINTSPC(502)::COLOURH:PRINTSPC(503)::COLOURI:PRINTSPC(504)::COLOURJ:PRINTSPC(505)::COLOURK:PRINTSPC(506)::COLOURL:PRINTSPC(507)::COLOURM:PRINTSPC(508)::COLOURN:PRINTSPC(509)::COLOURO:PRINTSPC(510)::COLOURP:PRINTSPC(511)::COLOURQ:PRINTSPC(512)::COLOURR:PRINTSPC(513)::COLOURS:PRINTSPC(514)::COLOURT:PRINTSPC(515)::COLOURU:PRINTSPC(516)::COLOURV:PRINTSPC(517)::COLOURW:PRINTSPC(518)::COLOURX:PRINTSPC(519)::COLOURY:PRINTSPC(520)::COLOURZ:PRINTSPC(521)::COLOURA:PRINTSPC(522)::COLOURB:PRINTSPC(523)::COLOURC:PRINTSPC(524)::COLOURD:PRINTSPC(525)::COLOURE:PRINTSPC(526)::COLOURF:PRINTSPC(527)::COLOURG:PRINTSPC(528)::COLOURH:PRINTSPC(529)::COLOURI:PRINTSPC(530)::COLOURJ:PRINTSPC(531)::COLOURK:PRINTSPC(532)::COLOURL:PRINTSPC(533)::COLOURM:PRINTSPC(534)::COLOURN:PRINTSPC(535)::COLOURO:PRINTSPC(536)::COLOURP:PRINTSPC(537)::COLOURQ:PRINTSPC(538)::COLOURR:PRINTSPC(539)::COLOURS:PRINTSPC(540)::COLOURT:PRINTSPC(541)::COLOURU:PRINTSPC(542)::COLOURV:PRINTSPC(543)::COLOURW:PRINTSPC(544)::COLOURX:PRINTSPC(545)::COLOURY:PRINTSPC(546)::COLOURZ:PRINTSPC(547)::COLOURA:PRINTSPC(548)::COLOURB:PRINTSPC(549)::COLOURC:PRINTSPC(550)::COLOURD:PRINTSPC(551)::COLOURE:PRINTSPC(552)::COLOURF:PRINTSPC(553)::COLOURG:PRINTSPC(554)::COLOURH:PRINTSPC(555)::COLOURI:PRINTSPC(556)::COLOURJ:PRINTSPC(557)::COLOURK:PRINTSPC(558)::COLOURL:PRINTSPC(559)::COLOURM:PRINTSPC(560)::COLOURN:PRINTSPC(561)::COLOURO:PRINTSPC(562)::COLOURP:PRINTSPC(563)::COLOURQ:PRINTSPC(564)::COLOURR:PRINTSPC(565)::COLOURS:PRINTSPC(566)::COLOURT:PRINTSPC(567)::COLOURU:PRINTSPC(568)::COLOURV:PRINTSPC(569)::COLOURW:PRINTSPC(570)::COLOURX:PRINTSPC(571)::COLOURY:PRINTSPC(572)::COLOURZ:PRINTSPC(573)::COLOURA:PRINTSPC(574)::COLOURB:PRINTSPC(575)::COLOURC:PRINTSPC(576)::COLOURD:PRINTSPC(577)::COLOURE:PRINTSPC(578)::COLOURF:PRINTSPC(579)::COLOURG:PRINTSPC(580)::COLOURH:PRINTSPC(581)::COLOURI:PRINTSPC(582)::COLOURJ:PRINTSPC(583)::COLOURK:PRINTSPC(584)::COLOURL:PRINTSPC(585)::COLOURM:PRINTSPC(586)::COLOURN:PRINTSPC(587)::COLOURO:PRINTSPC(588)::COLOURP:PRINTSPC(589)::COLOURQ:PRINTSPC(590)::COLOURR:PRINTSPC(591)::COLOURS:PRINTSPC(592)::COLOURT:PRINTSPC(593)::COLOURU:PRINTSPC(594)::COLOURV:PRINTSPC(595)::COLOURW:PRINTSPC(596)::COLOURX:PRINTSPC(597)::COLOURY:PRINTSPC(598)::COLOURZ:PRINTSPC(599)::COLOURA:PRINTSPC(600)::COLOURB:PRINTSPC(601)::COLOURC:PRINTSPC(602)::COLOURD:PRINTSPC(603)::COLOURE:PRINTSPC(604)::COLOURF:PRINTSPC(605)::COLOURG:PRINTSPC(606)::COLOURH:PRINTSPC(607)::COLOURI:PRINTSPC(608)::COLOURJ:PRINTSPC(609)::COLOURK:PRINTSPC(610)::COLOURL:PRINTSPC(611)::COLOURM:PRINTSPC(612)::COLOURN:PRINTSPC(613)::COLOURO:PRINTSPC(614)::COLOURP:PRINTSPC(615)::COLOURQ:PRINTSPC(616)::COLOURR:PRINTSPC(617)::COLOURS:PRINTSPC(618)::COLOURT:PRINTSPC(619)::COLOURU:PRINTSPC(620)::COLOURV:PRINTSPC(621)::COLOURW:PRINTSPC(622)::COLOURX:PRINTSPC(623)::COLOURY:PRINTSPC(624)::COLOURZ:PRINTSPC(625)::COLOURA:PRINTSPC(626)::COLOURB:PRINTSPC(627)::COLOURC:PRINTSPC(628)::COLOURD:PRINTSPC(629)::COLOURE:PRINTSPC(630)::COLOURF:PRINTSPC(631)::COLOURG:PRINTSPC(632)::COLOURH:PRINTSPC(633)::COLOURI:PRINTSPC(634)::COLOURJ:PRINTSPC(635)::COLOURK:PRINTSPC(636)::COLOURL:PRINTSPC(637)::COLOURM:PRINTSPC(638)::COLOURN:PRINTSPC(639)::COLOURO:PRINTSPC(640)::COLOURP:PRINTSPC(641)::COLOURQ:PRINTSPC(642)::COLOURR:PRINTSPC(643)::COLOURS:PRINTSPC(644)::COLOURT:PRINTSPC(645)::COLOURU:PRINTSPC(646)::COLOURV:PRINTSPC(647)::COLOURW:PRINTSPC(648)::COLOURX:PRINTSPC(649)::COLOURY:PRINTSPC(650)::COLOURZ:PRINTSPC(651)::COLOURA:PRINTSPC(652)::COLOURB:PRINTSPC(653)::COLOURC:PRINTSPC(654)::COLOURD:PRINTSPC(655)::COLOURE:PRINTSPC(656)::COLOURF:PRINTSPC(657)::COLOURG:PRINTSPC(658)::COLOURH:PRINTSPC(659)::COLOURI:PRINTSPC(660)::COLOURJ:PRINTSPC(661)::COLOURK:PRINTSPC(662)::COLOURL:PRINTSPC(663)::COLOURM:PRINTSPC(664)::COLOURN:PRINTSPC(665)::COLOURO:PRINTSPC(666)::COLOURP:PRINTSPC(667)::COLOURQ:PRINTSPC(668)::COLOURR:PRINTSPC(669)::COLOURS:PRINTSPC(670)::COLOURT:PRINTSPC(671)::COLOURU:PRINTSPC(672)::COLOURV:PRINTSPC(673)::COLOURW:PRINTSPC(674)::COLOURX:PRINTSPC(675)::COLOURY:PRINTSPC(676)::COLOURZ:PRINTSPC(677)::COLOURA:PRINTSPC(678)::COLOURB:PRINTSPC(679)::COLOURC:PRINTSPC(680)::COLOURD:PRINTSPC(681)::COLOURE:PRINTSPC(682)::COLOURF:PRINTSPC(683)::COLOURG:PRINTSPC(684)::COLOURH:PRINTSPC(685)::COLOURI:PRINTSPC(686)::COLOURJ:PRINTSPC(687)::COLOURK:PRINTSPC(688)::COLOURL:PRINTSPC(689)::COLOURM:PRINTSPC(690)::COLOURN:PRINTSPC(691)::COLOURO:PRINTSPC(692)::COLOURP:PRINTSPC(693)::COLOURQ:PRINTSPC(694)::COLOURR:PRINTSPC(695)::COLOURS:PRINTSPC(696)::COLOURT:PRINTSPC(697)::COLOURU:PRINTSPC(698)::COLOURV:PRINTSPC(699)::COLOURW:PRINTSPC(700)::COLOURX:PRINTSPC(701)::COLOURY:PRINTSPC(702)::COLOURZ:PRINTSPC(703)::COLOURA:PRINTSPC(704)::COLOURB:PRINTSPC(705)::COLOURC:PRINTSPC(706)::COLOURD:PRINTSPC(707)::COLOURE:PRINTSPC(708)::COLOURF:PRINTSPC(709)::COLOURG:PRINTSPC(710)::COLOURH:PRINTSPC(711)::COLOURI:PRINTSPC(712)::COLOURJ:PRINTSPC(713)::COLOURK:PRINTSPC(714)::COLOURL:PRINTSPC(715)::COLOURM:PRINTSPC(716)::COLOURN:PRINTSPC(717)::COLOURO:PRINTSPC(718)::COLOURP:PRINTSPC(719)::COLOURQ:PRINTSPC(720)::COLOURR:PRINTSPC(721)::COLOURS:PRINTSPC(722)::COLOURT:PRINTSPC(723)::COLOURU:PRINTSPC(724)::COLOURV:PRINTSPC(725)::COLOURW:PRINTSPC(726)::COLOURX:PRINTSPC(727)::COLOURY:PRINTSPC(728)::COLOURZ:PRINTSPC(729)::COLOURA:PRINTSPC(730)::COLOURB:PRINTSPC(731)::COLOURC:PRINTSPC(732)::COLOURD:PRINTSPC(733)::COLOURE:PRINTSPC(734)::COLOURF:PRINTSPC(735)::COLOURG:PRINTSPC(736)::COLOURH:PRINTSPC(737)::COLOURI:PRINTSPC(738)::COLOURJ:PRINTSPC(739)::COLOURK:PRINTSPC(740)::COLOURL:PRINTSPC(741)::COLOURM:PRINTSPC(742)::COLOURN:PRINTSPC(743)::COLOURO:PRINTSPC(744)::COLOURP:PRINTSPC(745)::COLOURQ:PRINTSPC(746)::COLOURR:PRINTSPC(747)::COLOURS:PRINTSPC(748)::COLOURT:PRINTSPC(749)::COLOURU:PRINTSPC(750)::COLOUR

Christmas Snap listing

From Page 55

```

300 COLOUR:PRINTSPC(4)::COLOUR:PRINTSPC(1)::COLOUR
A:PRINTSPC(3)::COLOUR:PRINTSPC(1)
)::COLOUR:PRINTSPC(4)::COLOUR:PRINTSPC(2)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(5)::COLOUR:PRINTSPC(3)::COLOUR:PRINTSPC(1);
310 COLOUR:PRINTSPC(4)::COLOUR:PRINTSPC(4)::COLOUR
E:PRINTSPC(1)::COLOUR:PRINTSPC(3)::COLOUR:PRINTSPC(4)
)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(3);
320 COLOUR:PRINTSPC(8)::COLOUR:PRINTSPC(1)::COLOUR
:PRINTSPC(2)::COLOUR:PRINTSPC(3)::COLOUR:PRINTSPC(1)
)::COLOUR:PRINTSPC(1);
330 COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(3)::COLOUR
:PRINTSPC(1)::COLOUR:PRINTSPC(2)::COLOUR:PRINTSPC(1)
)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(3)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(2)::COLOUR:PRINTSPC(1);
340 COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(3)::COLOUR
:PRINTSPC(1)::COLOUR:PRINTSPC(2)::COLOUR:PRINTSPC(1)
;
350 COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(2)::COLOUR
:PRINTSPC(2)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(1)
)::COLOUR:PRINTSPC(1)::COLOUR:PRINTSPC(6)::COLOUR:PRINTSPC(1)::PRINT"Jack in ";
355 PRINT"the Box ";
360 ENDPROC
370 DEFPROCcheck
375 *FX21,0
380 G=INKEY(100*ZX):IF G=32 THEN GOTO 400
390 ENDPROC
400 IF B=129 AND C=130 THEN count=TRUE:PROCtune
405 VDU7
409 *FX21,0
410 ENDPROC
420 DEFPROCproper_jack
430 A=135:B=129:C=130:D=1
31:E=132:F=133:G=134:VDU28,
1,30,8,1:COLOUR135:CLS:GOTO

```

```

200
440 ENDPROC
450 DEFPROCrandomcolour
460 random% = RND(7)+128
470 A=135:B=random%:IF B>
134 THEN B=8-7
480 C=B+1:IF C>134 THEN C
=C-7
490 D=B+2:IF D>134 THEN D
=D-7
500 E=B+3:IF E>134 THEN E
=E-7
510 F=B+4:IF F>134 THEN F
=F-7
520 G=B+5:IF G>134 THEN G
=G-7
530 H=B+6:IF H>134 THEN H
=H-7
540 ENDPROC
550 DEFPROCproper_santa
560 A=135:B=129:C=138:D=1
31:E=132:F=133:G=134:H=128:
VDU 28,1,30,8,1:COLOUR134:C
LS:GOTO 590
570 DEFPROCSanta(X%,Y%)
580 VDU28,XX,YX,XX+7,YZ-2
%:COLOUR135:CLS
590 COLOURG:PRINTSPC(3):::
COLOURA:PRINTSPC(2)::COLOUR
G:PRINTSPC(3)::COLOURG:PRIN
TSPC(2)::COLOURA:PRINTSPC(4
)::COLOURG:PRINTSPC(2)::COL
OURG:PRINTSPC(1)::COLOURA:P
RINTSPC(1)::COLOURB:PRINTSP
C(4)::COLOURA:PRINTSPC(1):::
COLOURG:PRINTSPC(1);
600 COLOURG:PRINTSPC(1):::
COLOURA:PRINTSPC(1)::COLOUR
E:PRINTSPC(1)::COLOURB:PRIN
TSPC(2)::COLOURE:PRINTSPC(1
)::COLOURA:PRINTSPC(1)::COL
OURG:PRINTSPC(1);
610 COLOURG:PRINTSPC(1):::
COLOURA:PRINTSPC(1)::COLOUR
B:PRINTSPC(1)::COLOURH:PRIN
TSPC(2)::COLOURB:PRINTSPC(1
)::COLOURA:PRINTSPC(1)::COL
OURG:PRINTSPC(1);
620 COLOURG:PRINTSPC(1):::
COLOURA:PRINTSPC(1)::COLOUR
B:PRINTSPC(4)::COLOURA:PRIN
TSPC(1)::COLOURG:PRINTSPC(1
)::COLOURG:PRINTSPC(2)::COL
OURA:PRINTSPC(4)::COLOURG:P
RINTSPC(2);
630 COLOURG:PRINTSPC(3):::
COLOURA:PRINTSPC(2)::COLOUR
G:PRINTSPC(3)::COLOURG:PRIN
TSPC(2)::COLOURG:PRINTSPC(2
):::
700 COLOURG:PRINTSPC(1):::
COLOURA:PRINTSPC(2)::COLOUR
G:PRINTSPC(3)::COLOURB:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
)::COLOURB:PRINTSPC(3)::COL
OURH:PRINTSPC(1)::COLOURG:P
RINTSPC(2)::COLOURH:PRINTSP
C(1)::COLOURG:PRINTSPC(2);
710 COLOURG:PRINTSPC(2):::
COLOURB:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURB:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
):::
720 COLOURG:PRINTSPC(2):::
COLOURB:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURB:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
):::
730 COLOURG:PRINTSPC(2):::
COLOURB:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURB:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
)::COLOURG:PRINTSPC(2)::COL
OURB:PRINTSPC(1)::COLOURG:P
RINTSPC(2)::COLOURB:PRINTSP
C(1)::COLOURG:PRINTSPC(2);
740 COLOURG:PRINTSPC(2):::
COLOURH:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURH:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
)::COLOURG:PRINTSPC(2)::COL
OURH:PRINTSPC(1)::COLOURG:P
RINTSPC(2)::COLOURH:PRINTSP
C(1)::COLOURG:PRINTSPC(2);
750 COLOURG:PRINTSPC(2):::
COLOURH:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURH:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
):::
760 COLOURG:PRINTSPC(2):::
COLOURH:PRINTSPC(1)::COLOUR
G:PRINTSPC(2)::COLOURH:PRIN
TSPC(1)::COLOURG:PRINTSPC(2
)::COLOUR G:PRINTSPC(1)::CO
LOURH:PRINTSPC(2)::COLOURG:
PRINTSPC(1)::COLOURH:PRINTSP
C(2)::COLOURG:PRINTSPC(2);
765 COLOURG:PRINTSPC(8);
767 PRINT" Santa ";
770 ENDPROC
780 DEFPROCproper_tree
790 A=135:B=129:C=132:D=1
31:E=132:F=133:G=134:H=129:
VDU28,1,30,8,1:COLOUR135:CL
S:GOTO 820
800 DEFPROCchristmas_tree
(X%,Y%)
810 VDU28,XX,YX,XX+7,YZ-2
%:COLOUR135:CLS
820 COLOURA:PRINTSPC(8)::C
OLOURA:PRINTSPC(4)::COLOURD
:PRINTSPC(1)::COLOURA:PRINT
SPC(3)::COLOURA:PRINTSPC(3)
::COLOURD:PRINTSPC(3)::COLO
URA:PRINTSPC(2);
830 COLOURA:PRINTSPC(4)::C
OLOURD:PRINTSPC(1)::COLOURA
:PRINTSPC(3)::COLOURA:PRINT
SPC(4)::COLOURC:PRINTSPC(1)
::COLOURA:PRINTSPC(3);
840 COLOURA:PRINTSPC(3)::C
OLOURB:PRINTSPC(1)::COLOURC
:PRINTSPC(1)::COLOURB:PRINT
SPC(1)::COLOURA:PRINTSPC(2)
:::
850 COLOURA:PRINTSPC(3)::C
OLOURB:PRINTSPC(1)::COLOURC
:PRINTSPC(1)::COLOURB:PRINT
SPC(1)::COLOURA:PRINTSPC(2)
:::

```

This listing is included in this month's cassette tape offer. See order form on Page 47.

Silly Santa listing

From Page 45

```

10REM SILLYSANTA
20REM By Rob Frost
30REM With help from Sue
Frost
40REM (C) ELECTRON USER
1984
50MODE2
60VDU23;8202;0;0;0;
70PROCredits
80GCOL0,134:CLG
90GCOL0,2:MOVE0,0:MOVE12
79,0:PL0T85,0,400:PL0T85,12
79,400
100PROCcircle(600,800,240
,7)
110PROCcircle(600,360,360
,7)
120PROCcircle(600,800,220
,1)
130PROCcircle(600,360,350
,1)
140PROCcircle(600,760,200
,7)
150PROCcircle(600,800,200
,3)
160PROCcircle(520,850,40
,4)
170PROCcircle(680,850,40
,4)
180PROCcircle(600,760,50
,1)
190PROCcircle(600,800,70
,3)
200PROCcircle(600,800,20
,1)
210PROCcircle(600,100,10
,7)
220PROCcircle(600,200,10
,7)
230PROCcircle(600,300,10
,7)
240PROCcircle(600,400,10
,7)
250PROCcircle(600,500,10
,7)
260PRINTTAB(4,20)"SILLY S
ANTA"
270PRINTTAB(0,0)"
280VDU19,6,4,0,0,0
290GCOL0,3:MOVE1000,900:M
OVE1100,900:PL0T85,1050,800
300MOVE1000,850:MOVE1100,
850:PL0T85,1050,950
310PROCgoodtune:FORdelay%
=0TO5000:NEXT

```

```

320TX=10000:MX=10000
330MODE6
340VDU19,0,4,0,0,0
350PROCinstruct
360MODE2
3700NERROR CLEAR:GOT0330
380VDU19,1,4,0,0,0
390*FX11,0
400VDU23,8202;0;0;0;
410COLOUR1
420PROCchoose
430PROCsetup
440PROChouse(100,800):PRO
Chouse(500,800):PROChouse(9
00,800):PROChouse(100,400):
PROChouse(500,400):PROChous
e(900,400)
450PROCsanta(200,900,2):P
ROCsanta(600,900,3):PROCsan
ta(1000,900,4):PROCsanta(20
0,500,5):PROCsanta(600,500,
6):PROCsanta(1000,500,0)
460PROCsack
470VDU4
480PROCdecide
490CLEAR
500GOT0360
510END
520DEFFPROChouse(xpos%,ypo
s%)
530MOVE xpos%, ypos%
540MOVE xpos%+200, ypos%
550PL0T85, xpos%, ypos%+100
560PL0T85, xpos%+200, ypos%
+100
570MOVE xpos%+100, ypos%+1
80
580PL0T85, xpos%, ypos%+100
590MOVE xpos%+150, ypos%+1
80:MOVE xpos%+150, ypos%+100
:PL0T85, xpos%+180, ypos%+100
:MOVE xpos%+150, ypos%+100:F
LOT85, xpos%+180, ypos%+100
600 MOVE0,0
610ENDPROC
620DEFFPROC Santa(xpos%,ypo
s%, col%)
630VDU5
640GCOL 0,col%
650MOVE xpos%, ypos%
660PRINT;santa$
670ENDPROC
680DEFFPROCsack
690VDU19,14,3,0,0,0
700VDU23,240,15,7,3,1,3,7
,15,31
710VDU23,241,240,224,192,
128,192,224,240,248
720VDU23,242,63,63,63,63,
31,31,15,7
730VDU23,243,252,252,252,
252,248,248,248,224
740sack$=CHR$240+CHR$241+
CHR$8+CHR$10+CHR$8+CHR$242+
CHR$243
750GCOL0,14
760MOVE 80,890:PRINT;sack$:
MOVE 890,890:PRINT;sack$:
MOVE 890,890:PRINT;sack$:
MOVE 890,890:PRINT;sack$:
780GCOL4,1
790MOVE 115,860:PRINT;A$(1
):MOVE 515,860:PRINT;A$(2
):MOVE 915,860:PRINT;A$(3
):MOVE 115,460:PRINT;A$(4
):MOVE 515,460:PRINT;A$(5
):MOVE 915,460:PRINT;A$(6
):MOVE 810GCOL0,7
820MOVE 100,750:PRINT;H$:
I$:MOVE 500,750:PRINT;H$;J$:
MOVE 900,750:PRINT;H$;K$;M
OVE 100,380:PRINT;H$;L$;MOV
E 500,380:PRINT;H$;M$;MOVE
900,380:PRINT;H$;N$:
830MOVE300,850:DRAW500,85
0:MOVE700,850:DRAW900,850
840MOVE300,450:DRAW500,45
0:MOVE700,450:DRAW900,450
850MOVE600,800:DRAW200,55
0:MOVE600,800:DRAW1000,550
860ENDPROC
870DEFFPROCdecide
880COLOUR1
890TIME=0
900REPEAT
910VDU19,col%,7,0,0,0
920z%=newz%
930col% =z%+1:IFcol% =7 THEN
N col% =0
940VDU19,col%,1,0,0,0
950VDU28,0,24,19,22
960PRINTTAB(0,1)"You are
at house ";z%
970VDU28,0,31,19,25
980VDU23;8202;0;0;0;
990IF LEN(carry$)>2 THEN
GOTO 1080
1000IF A$(z%)="" THEN GOT01
080
1010REPEAT
1020INPUTTAB(0,1)"What will
I you take",takes$
1030PRINTTAB(18,1)" "
1040UNTIL takes$="" OR takes
$=A$(z%)
1050 IF takes$=A$(z%) PROCr
eaove
1060carry$=carry$+takes$
1070CLS
1080 IF A$(z%)<>"" THEN GO
TO 1190
1090REPEAT
1100INPUTTAB(0,1)"What wil
I you drop",leave$:
1110PRINTTAB(18,1)" "
1120FOR loop=1 TO LEN carr
y$:
1130 drop$=MID$(carry$,loo
p,1)
1140 IF drop$=leave$ GOT011
60
1150NEXT
1160UNTIL leave$="" OR lea
ve$=drop$:
1170IF drop$=leave$ PROCdro
p
1180CLS
1190REPEAT
1200INPUTTAB(0,1)"Where wi
ll you go",newz%
1210PRINTTAB(18,1)" "
1220IF newz% =0 AND A$(1) =
"1" AND A$(2) = "2" AND A$(3) =
"3" AND A$(4) = "4" AND A$(5) =
"5" AND A$(6) = "6" THEN GOT
0 1300
1230IF z% =1 UNTIL newz% =2
1240 IF z% =2 UNTIL newz% =1
OR newz% =4 OR newz% =3 OR n
ewz% =6
1250IF z% =3 UNTIL newz% =2
1260IF z% =4 UNTIL newz% =5
OR newz% =2
1270IF z% =5 UNTIL newz% =4
OR newz% =6
1280IF z% =6 UNTIL newz% =2
OR newz% =5
1290CLS
1300UNTIL TIME>T% OR A$(1)
="1" AND A$(2) = "2" AND A$(3) =
"3" AND A$(4) = "4" AND A$(5) =
"5" AND A$(6) = "6"
1310IF TIME>T% THEN PROCfa
il ELSE PROCsuccess
1320ENDPROC
1330DEFFPROCsetup
1340VDU24,0;350;1279;1023;
1350VDU28,0,31,19,22
1360GCOL0,129:CLG:COLOUR13

```

5:CLS
 1370VDU23,230,24,126,255,1
 26,126,60,24,24
 1380VDU23,231,255,189,189,
 60,60,102,102,231
 1390VDU23,255,255,255,255,
 255,255,255,255
 1400santa\$=CHR#230+CHR#8+C
 HR#10+CHR#231
 1410VDU19,2,7,0,0,0:VDU19,
 3,7,0,0,0:VDU19,4,7,0,0,0:V
 DU19,5,7,0,0,0:VDU19,6,7,0,
 0,0:VDU19,0,7,0,0,0
 1420H#="no."
 1430I\$="1":J\$="2":K\$="3":L
 \$="4":M\$="5":N\$="6"
 1440carry\$=""
 1450newz%1
 1460ENDPROC
 1470DEFPROCremove
 1480VDU5
 1490GCOL0,7
 1500IF z%1 THEN MOVE80,89
 0:PRINT;sack\$:A\$(1)=""
 1510IF z%2 THEN MOVE480,6
 90:PRINT;sack\$:A\$(2)=""
 1520IF z%3 THEN MOVE880,9
 90:PRINT;sack\$:A\$(3)=""
 1530IF z%4 THEN MOVE80,49
 0:PRINT;sack\$:A\$(4)=""
 1540IF z%5 THEN MOVE480,4
 90:PRINT;sack\$:A\$(5)=""
 1550IF z%6 THEN MOVE880,4
 90:PRINT;sack\$:A\$(6)=""
 1560VDU4
 1570ENDPROC
 1580DEFPROCdrop
 1590X=INSTR(carry\$,drop\$)
 1600Y=LEN(carry\$)
 1610carry\$=LEFT\$(carry\$,X-
 1)+MID\$(carry\$,X+1,Y)
 1620VDU5
 1630IF z%1 THEN GCOL0,14:
 MOVE 80,890:PRINT;sack\$:G0
 L4,1:MOVE 115,860:PRINTleav
 e\$:A\$(1)=leave\$
 1640IF z%2 THEN GCOL0,14:
 MOVE 480,890:PRINT;sack\$:G0
 L4,1:MOVE 515,860:PRINTleav
 e\$:A\$(2)=leave\$
 1650IF z%3 THEN GCOL0,14:
 MOVE 880,890:PRINT;sack\$:G0
 L4,1:MOVE 915,860:PRINTleav
 e\$:A\$(3)=leave\$
 1660IF z%4 THEN GCOL0,14:
 MOVE 80,490:PRINT;sack\$:G0
 L4,1:MOVE 115,460:PRINTleav

e\$:A\$(4)=leave\$
 1670IF z%5 THEN GCOL0,14:
 MOVE 480,490:PRINT;sack\$:G0
 L4,1:MOVE 515,460:PRINTleav
 e\$:A\$(5)=leave\$
 1680IF z%6 THEN GCOL0,14:
 MOVE 880,490:PRINT;sack\$:G0
 L4,1:MOVE 915,460:PRINTleav
 e\$:A\$(6)=leave\$
 1690VDU4
 1700ENDPROC
 1710DEFPROCfail
 1720TX=TIME
 1730VDU26
 1740VDU23;8202;0;0;0;
 1750COLOUR128:CLS
 1760VDU20:VDU19,14,3,0,0,0
 ,19,1,4,0,0,0
 1770COLOUR2
 1780PRINT ""You took too
 long!""
 1790PRINT ""Granny will be
 ""surprised to get a""ra
 ttle."
 1800PROCbadtune
 1810PROCfail
 1820TX=TX+2000
 1830PRINTTAB(0,11)"Space b
 ar for next""go."
 1840REPEAT UNTIL GET=32
 1850ENDPROC
 1860DEFPROCsuccess
 1870TX=TIME
 1880VDU20:VDU19,14,3,0,0,0
 ,19,1,4,0,0,0
 1890VDU4
 1900VDU28,0,31,19,22
 1910VDU23;8202;0;0;0;
 1920COLOUR129:CLS
 1930COLOUR3
 1940PRINT ""A happy person
 in""every house."
 1950PRINT ""You took ""TXDI
 V100;" seconds"
 1960PROCgoodtune
 1970IF MX>TX THEN MX=TX
 1980PROCfail
 1990PRINTTAB(0,11)"Space b
 ar for next""go."
 2000REPEAT UNTIL GET=32
 2010ENDPROC
 2020DEFPROCinstruct
 2030PRINT ""Santa has been
 very silly this year."""H
 e has left his presents at
 the wrong""houses."
 2040PRINT ""You have only a

limited time before all""
 the people wake up and find
 his mistake."
 2050PRINT ""You can move fr
 om house to house along""t
 he white paths, collecting
 presents""which can be see
 n numbered in each""house
 and leaving presents at the
 ""right house."
 2060PRINT ""Another problem
 is that your strength""is
 limited. You can only mana
 ge to""hold three sets of
 presents."
 2070PRINT ""Press the space
 bar to continue."
 2080REPEAT UNTIL GET=32
 2090CLS
 2100PRINT ""If you do not
 want to take or leave""an
 y presents just press RETUR
 N."""You must always press
 RETURN after ""you have en
 tered any number."
 2110PRINT ""When you have
 completed your task""press
 0 to take you home."""San
 ta will then send you on a
 new""mission but he will g
 ive you less time."
 2120PRINT ""Will you get
 a happy person in every""h
 ouse?"
 2130PRINT ""Press the spa
 ce bar to start the game."
 2140REPEAT UNTIL GET=32
 2150ENDPROC
 2160DEFPROCchoose
 2170DIMA\$(6)
 2180B\$="123456":C\$="":C=6
 2190REPEAT
 2200R=RND(C)
 2210C\$=C\$+MID\$(B\$,R,1)
 2220B\$=LEFT\$(B\$,R-1)+RIGHT
 \$(B\$,LEN(B\$)-R)
 2230C=C-1
 2240UNTILB\$=""
 2250FOR N=1 TO 6
 2260A\$(N)=MID\$(C\$,N,1)
 2270NEXTN
 2280ENDPROC
 2290DEFPROCgoodtune
 2300RESTORE2320
 2310FORtune%1=1TO30:READpit
 ch%,dur%:SOUND1,-15,pitch%+
 50,dur%-2:SOUND1,0,52,1:NEXT

This listing is included in
 this month's cassette
 tape offer. See order
 form on Page 47.

AT COMSOFT WE'VE GOT IT ALL TAPED UP FOR ONLY £4.95!

EXCLUSIVELY
AVAILABLE
MAIL ORDER!

SAS COMMANDER Shoot terrorists, save hostages, restore freedom. We are giving over 100 prizes away to the highest scorers! "Excellent. A very tempting game expertly produced and with all the refinements one could expect" — *Games Computing*.

CUSTARD PIE FIGHT An hilarious game for one or two players that will literally have you falling over with laughter! "Title of the week." *PCN*

SERPENTS LAIR A graphical geographical adventure. "The use of graphics impressed me. I can see the whole family getting involved. A pleasant first adventure in every way" *HCW*

PLAYBOX Three educational programs on one cassette for the whole family! Used in schools and homes nationwide! "Superb. Will be a very valuable addition to many schools and homes. I would certainly have spent my money on it with only two of the three games." *Electron User*.

We Guarantee:
Same Day Despatch
By First Class Post

TAPES: £4.95!

Please add 55p post & packing

COMSOFT
67 KENT ROAD
HARROGATE
NORTH YORKS
HG1 2NH.
TEL: 0532 665621
or 0423 57464.



If you already own the popular games of '**CHUCKIE EGG**'*, '**KILLER GORILLA**' or '**MONSTERS**' # then our UPGRADES:

"CHUCKEE"

will provide 3 Speeds, 8 New Screens, Variable Extended Jump and Select Start Level (1-41).

DESIGN YOUR OWN SCREENS!!!

"KILLA"

will provide up to 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 TWIDDLEERS

Dept, EU11, 158 Church End, Harlow, Essex CM19 5PF.

* & # are trademarks of A&F Software and Acornsoft

ADVERTISERS INDEX

Acornsoft	14, 15	M.P. Software	
Aggressive Software	60	New Horizon	60
Bec Sports	60	National Micro	
Bit Twiddlers	60	Centres	42, 43
Comsoft	60	Potter Programs	28
Epic	36	Qualsoft	11
First Byte Computers	17	Shards Software	35
Golem	28	Shiva	53
Icon Software	25	Skywave Software	53
Kay-Ess Software	11	Software Invasion	2
Kosmos	21	Software Liason	60
Microbyte	28	Squirrelsoft	21
Micropower	40, 64	Superior Software	63
Minic Business Services	53	Voltmace	36

STRIPPER II

The professional tape back-up program for the Electron. This is a totally new and updated version of the best-seller, Stripper I, which has been selling for over a year for use on the BBC Micro. Now, due to popular demand, available on the Electron also.

As Stripper II will duplicate all your commercial software we must insist on personal use only, please.

Just send £4 inc. cheque or P.O. to receive Stripper II by return post. Educational orders welcome.

Orders to:

**Aggressive Software, 14 Elmore Road,
Sheffield S10 1BY.**



FREE BALL! with 'YOUR TEAM'

a Bobby Charlton
Leather Match Ball
worth £6.99

A must
for all soccer
enthusiasts!

Yes, absolutely free for a limited period, with every copy of the program "Your Team", a real leather Match Ball autographed by Bobby Charlton (R.R.P. £6.99). "Your Team" is a database program to record and analyse your favourite teams match results, scores, attendances, statistics etc. All recorded in your clubs colours. Football league, Scottish league or your own team school or club can be kept season after season.

Available for Spectrum, BBC, Electron, CBM 64 — £9.95 plus £1.00 P&P. Order today enclosing cheques/P.O. Access/Visa To:

**Bec Sports, Dept EU, Bec House, Highlands Ave, Northampton.
Tel: 0604 499 246**

FOR
ELECTRON

MICROWORD Word-Processing

FOR
BBC

DO YOU HAVE: An ELECTRON PRINTER INTERFACE? A BBC without disc drives? If "the sky" is not the limit for you, and you need a Word-Processor, then MICROWORD is what you've been waiting for. MICROWORD is the ideal family/small business package. Invaluable for the writing of letters, manuscripts, minutes of meetings, labels, as a typing tester, and a multitude of other uses.

Modestly priced — MICROWORD — your introduction to a whole new world of writing. Send today for immediate dispatch to:

SOFTWARE LIASON, 8 Darwin Road, Welling, Kent DA16 2EG. Tel: 01-304 3411.

Dealer Enquiries Welcome.

Cheque/P.O. for £6.90 + 50p P&P

NEW HORIZON COMPUTER CENTRES

**BBC
CBM 64
SPECTRUM**



**ELECTRON
VIC 20
DRAGON**

FOR
EDUCATIONAL
SOFTWARE

FAST MAIL ORDER SERVICE
FREE P&P FOR ORDERS £25 OR MORE
SEND FOOLSCAP S.A.E. FOR CATALOGUE

**To: New Horizon Computer Centres
1 Goodall Street Walsall WS1 1QU**

Micro Messages

I READ with interest the letter in the October 1984 edition of the Electron User concerning moving the Electron screen down one line (p 62 "Moving down the line"). I would like to expand upon the ideas of Mr K. Sharkey in the following way.

I notice that a VDU 11 (Ctrl K) will in fact only move the screen down one line if the text cursor is positioned somewhere on the top line of the screen.

To overcome this I would suggest that a VDU 30 (Ctrl.) should be used immediately before the VDU 11. This has the effect of homing the text cursor to the top left hand corner of the screen, therefore ensuring that the following VDU 11 command has the desired effect.

Having carried out this operation you do however leave the text cursor at the top of the screen, which means that any subsequent text will be printed at that position.

To rectify this the following statement could be used:

**A=POS:B=VPOS-1:
VDU30,11:P.TAB(A,B)**

The variables POS and VPOS contain the X, Y coordinates of the text cursor's present screen position.

Having "remembered" your current screen position you can then use the VDU 30,11 command to move the screen display down one line. Having moved the image down the screen it is then possible, with the P.TAB(A,B) command, to return to your original position.

As you can see, you do in fact return to the same point on the X axis but one line up on the Y axis 'B=VPOS-1'.

This is to enable you to return to the "physical" point that you left, rather than the point in the text.

This is really only necessary if your original position was on the bottom line of the screen. To return to that position in the text you would need to move to a position off the bottom of the screen.

I would suggest that users

incorporate these commands into a procedure and then call the procedure when they wish to display a complete picture on the screen.

I hope that this additional information will be of value to your readers. The credit goes to K. Sharkey whose initial idea was the catalyst to the above line of thought. — **Martin Grantham, Acorn Customer Services Dept.**

● As ever, Acorn Customer Services Department produces the goods. Any more hints for us, Martin?

Short cut

IF you own Mr Wiz from Superior Software and just can't get onto the next screen then try pressing Caps lock, Q, 1, all at the same time. You should start at the beginning of the next screen. — **Richard R. Fairbrother, Stapleford, Notts.**

● Many thanks for the tip — but it sounds suspiciously like cheating.

Bug in solitaire

HAVING typed in and run your Solitaire program (in the July 1984 issue) I find there is a small bug.

If you wish to move a disc in coordinate position (3,1) up you are able to do so. Of course you would normally not wish to make this move, but if you do accidentally you will ruin the game.

There is an easy remedy — by inserting GCOL 0,0 in to line 200 and removing GCOL 0,0 from line 210.

This changes the X and Y labels to black so that the X label is not detected as a disc and so the move cannot be

made. — **Robert D. Snelling, Haslingfield, Cambridge.**

● Well spotted Robert. And many thanks for the remedy. It's always nice to hear from people who've probed into the workings of the games as well as playing them.

View into the ROM

Here's a short program for those of you interested in finding out what lies in the output area of your ROM. You'll probably be amazed, I was.

Enter the Basic program, making sure that nothing occupies the output area (Plus 1 or other add-on) then run the program.

```
10 FOR N=6FC00 TO 6FF00
20 IF ?N>31 AND ?N<127 THEN PRINT CHR$(?N);
30 NEXT
40 END
```

While I was playing around PEEKing into the ROM area I found some interesting words not listed in the manual, such as BOOT.

Also in the error message area I found "No Not Bad". Is this a compliment? — **E.T. (nothing to do with the**

film) Jones, Hillingdon, Middlesex.

● Fascinating stuff Mr Jones. The BOOT command is for the disc filing system, but the error message is a mystery. We doubt that it's a compliment!

Station now closing down

REGARDING N. Wright's letter in Micro Messages about broadcasting Electrons, you said that you couldn't get yours to broadcast on your radio.

Well I have a very mysterious situation at my house. Both my sister and I got the same music centre for Christmas and my sister keeps on complaining about my Electron's sound effects on her radio in her room.

Yet mine is right next to my Electron and there's no interference at all. So we swapped machines — and it was still the same. Hers in her room suffered interference and mine in my room didn't. So this might mean distance from the Electron has something to do with it. — **Miles Touchard, Maidenhead, Berkshire.**

● We had a feeling of

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.**

Micro Messages

From Page 61

foreboding when we said last month that this correspondence was closed. Since then we've had lots of letters pointing out the distance effect, so we are publishing this FINAL letter.

Olympic records

I'D like to make a complaint about Micro Olympics.

After receiving a copy, I was most angry at the way the player runs in the running events: the action of pressing down two keys as fast as possible was causing vibrations which were felt throughout the house.

On account of this, and the fact that my parents weren't going to pay for another keyboard, I was banned from playing this otherwise very good game.

Thus I wish I'd never had the luck to get a copy.

Meanwhile on a less serious note, here are my records on the day before I was banned:

100m	9.07 secs
200m	18.93 secs
400m	42.41 secs
800m	1:47.68 secs
1500m	3:54.58 secs
Javelin	99.99m
Discus	71.11m
Hammer	84.79m
Long jump	8.98m
High jump	2.40m
Pole vault	5.60m

— C.J. Underhill, Whitton, Twickenham.

● Don't your fingers get very sore?

— and more

I THINK I've set the standards for all you athletes out there with my records on Micro Olympics:

100m	8.92 secs
200m	17.33 secs
Long jump	9.01m
High jump	2.39m
Javelin	100.93m
Discus	70.85m

— A. Ennis, Herne Hill, London.

● Congratulations on your

micro-athleticism! We have little doubt that others will be claiming better records.

Claim to fame

USING the command: ?&FE45 = 1, I have been able to slow down the BBC Micro. Is there any way of doing this on the Electron?

May I say that so far Electron User has been unfaultable. However, maybe I could suggest a couple more features which even The Micro User hasn't got.

□ A Hall of Fame, where readers could boast their high scores. Here are some of my own:

Cybertron	41630
Chuckie Egg	365790
Positron	41960
Snapper	23465
Croaker	14260
Cylon Attack	31970

Maybe a Micro Olympics table could be included.

□ An adventure solver page, where tips on how to solve popular adventures could be printed. I have solved two: Stranded, and Arrow of Death (pt. 1). Maybe Twin Kingdom Valley and Castle Frankenstein could be the first? — David Thompson, Sale, Cheshire.

● If there is a POKE to slow down the Electron, we don't

know about it. What we want to know is why you want to slow it down in the first place!

The Hall of Fame idea is nice, but how do we know that the scores are genuine? As for Micro Olympics, your wish is our command.

Finally we're trying to persuade Merlin to do a regular column, but every time we call round on him we're told that he's gone out for a spell.

Oh Brother

I RECENTLY bought a Brother HR5 printer on the understanding that it was compatible with the Electron.

However I am having great trouble producing graphics, as everything seems to be for the Epsom printers.

I understand a screen dump routine is required, but cannot find one for Brother to Electron. Can you help? — Ben Still, Bushey Heath, Herts.

● We haven't come across a screen dump routine for the Brother HR5. Could any of our readers help?

Exploring the OS

AFTER having endured Basic and eventually got round to learning machine code I am

now ready to risk exploring the operating system. Could you recommend a book? — Ian Woodruff, Garstang, Lancs.

● There are two books we can recommend. The first is the Basic ROM User Guide by Mark Plumley, published by Adder. This explains in depth how the Electron's Basic works.

The second is Acornsoft's Electron User Guide, by Mark Holmes and Adrian Dickens. This covers both the software and hardware aspects of the Electron.

Trill to victory

IN answer to Chris Jones' enquiry (Electron User November 1984) concerning the Micro Olympics, I am a music teacher and as a pianist I find no difficulty in beating the contestants (and world records!).

I just play the "left foot — right foot" keys as if they were a trill on the piano. I think readers might well discover that most pianists will equally be world champions! — David Forshaw, St. Helens, Lancs.

● So playing the piano helps you excel at Micro Olympics, does it? Is the reverse true? Does Micro Olympics help you with your piano playing?

Diagonal scrolling demonstration

I'VE written a short program demonstrating a diagonal scrolling technique. The string variable AS will take a message of any length but the space at the end is necessary.

— M.J. Rance, Broadstairs, Kent.

● Thanks for the first diagonal scroller we've received. Our original scrolling program seems to have struck a chord with Electron User readers and we've had all sorts of similar programs.

What a lot of little scrollers you are.

10 REM DIAGONAL SCROLL	140 REPEAT:Y=Y+1
20 REM MICHAEL RANCE	150 PRINT TAB(25-X+Y,19-X
30 MODE 6	+Y)MID\$(A\$,Y,1);
40 VDU 23,1,0;0;0;0;	160 UNTIL Y=X:NEXT
50 A\$="***ELECTRON USER*	170 REM SCROLL REST OF ME
**DIAGONAL SCROLL BY M.J.RA	SSAGE
NCE "	180 FOR a=2*LENA\$ TO 1 ST
60 REPEAT	EP-1
70 PROCscroll	190 FOR Y=1 TO 15
80 UNTIL FALSE	200 IF 2*LENA\$-a+Y>LENA\$
90 END	b=LENA\$-a+Y ELSE b=2*LENA\$-
100 DEF PROCscroll	a+Y
110 REM FILL FIRST DIAGONAL	210 PRINT TAB(10+Y,4+Y)M
120 FOR X=1 TO 14	I\$(A\$,b,1);
130 Y=	220 NEXT,:ENDPROC

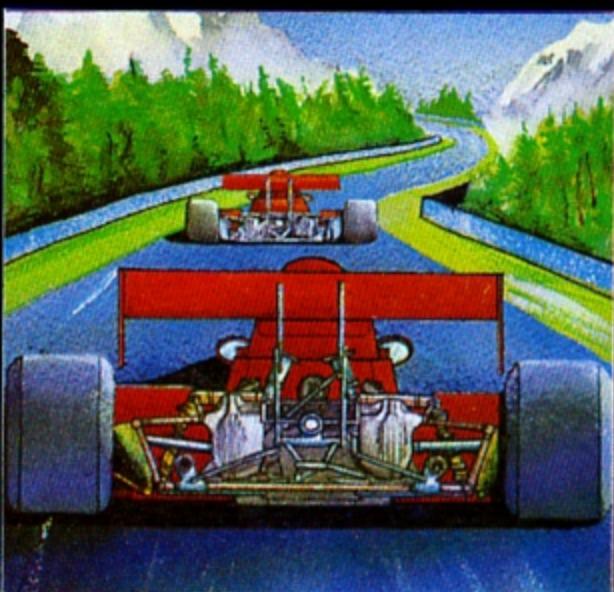
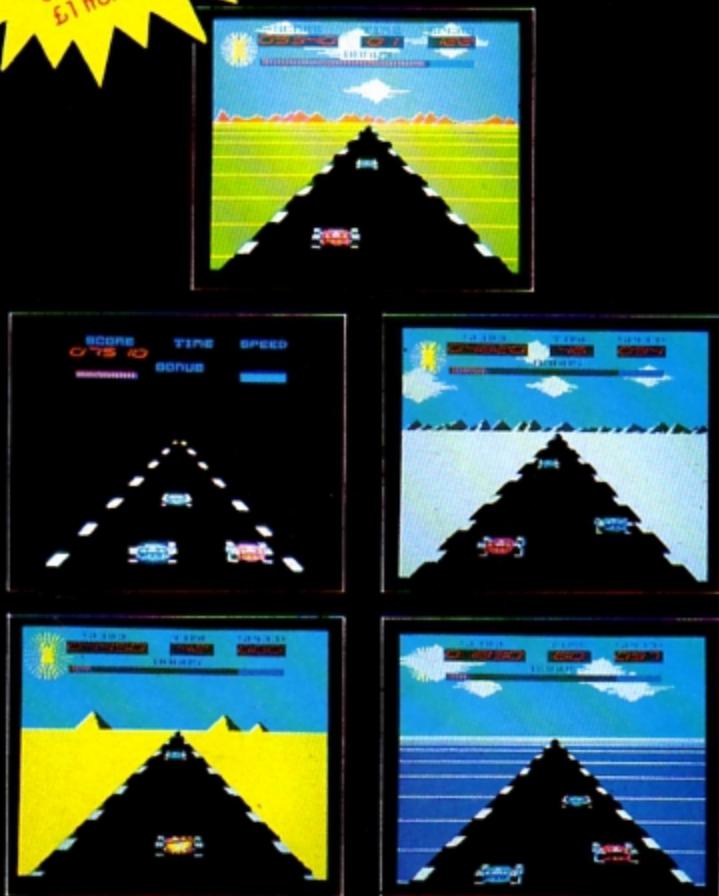
SPECIAL OFFER!
Order both
directly from
us and deduct
£1 from each.

THE BEST BBC MICRO SOFTWARE

* PRODUCED BY AN INDEPENDENT SOFTWARE HOUSE *

* TOP QUALITY MACHINE-CODE PROGRAMS *

ACORN
ELECTRON
BBC
MICRO

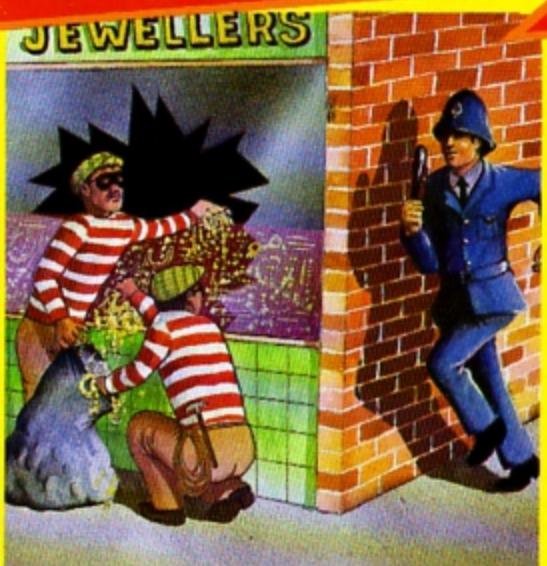


OVERDRIVE (32K)

£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.



**MICRO
POWER**

MICRO POWER LTD.,
NORTHWOOD HOUSE, NORTH STREET,
LEEDS LS7 2AA TEL: (0532) 458800
SELECTIVE BRANCHES OF BOOTS, CO-OP
MENZIES, W. H. SMITH, WOOLWORTHS AND ALL
GOOD DEALERS
AUTHORS! WE PAY 20% ROYALTIES!

WATCH OUT
FOR OUR NEW
PACKAGING AND
CATALOGUE

