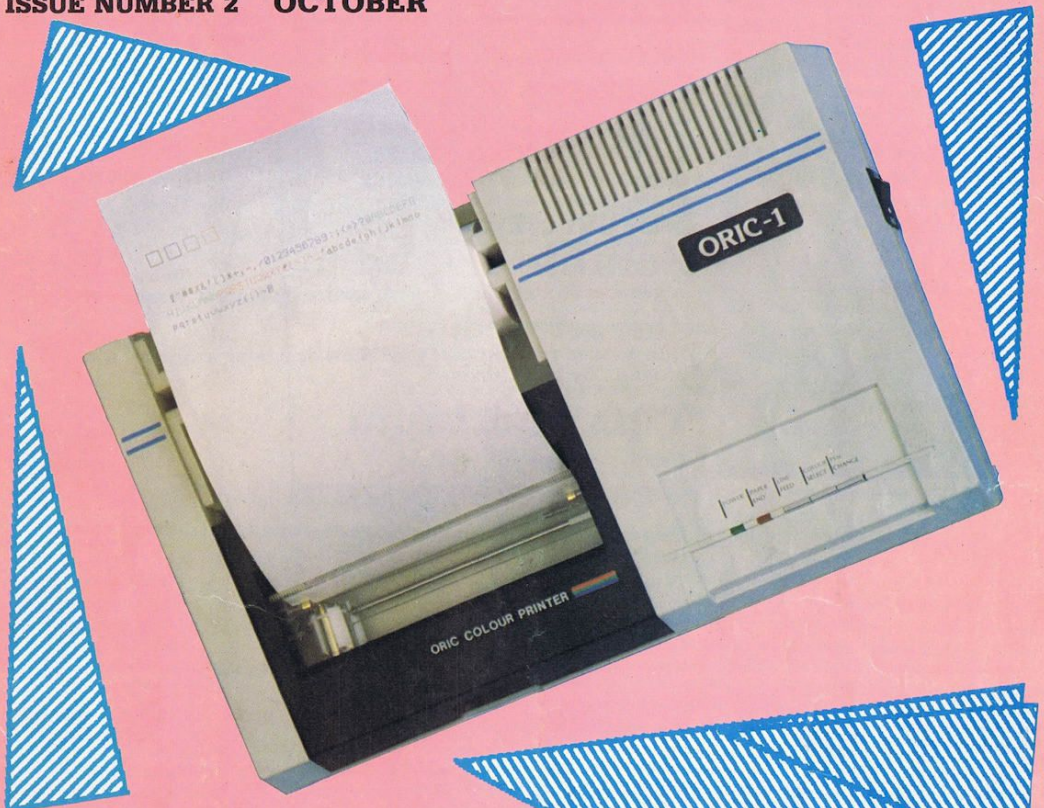
ORIC USER

ISSUE NUMBER 2 OCTOBER

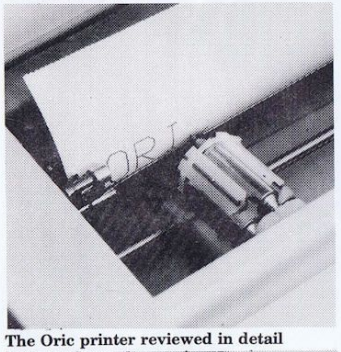


INSIDE:

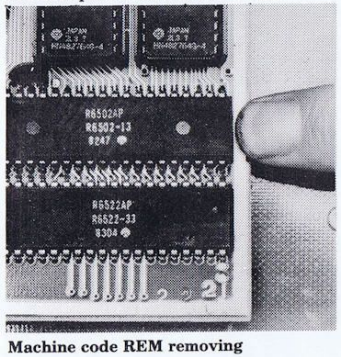
PRICE 85p

Contents:

A real-time game set in the far reaches of the solar system



The Oric printer reviewed in detail



Machine code REM removing

Photographs by Andrew Wiard (Report)

ORIC USER

THE TEAM

Editorial:
Ian Williams
Frank Brecht

Software editorial:
Fin Fahey
Tarquin Binary

Graphics and Advertising:
Maggie McDonald

Typesetting and Printing by:
Calverts Press
55 Mount Pleasant
London WC1
01-278 7177

Published by:
Co-operative Computers &
Communications Ltd
20 Wynford House
Wynford Road
London N1

BY NOW YOU HAVE pulled your micro from its box. You will probably have had fun and games tuning it in so that a moderately reasonable picture appears. You may have dabbled with programming and even got something to run.

But face it, the chances are that you are a little disappointed. Don't worry, this condition is perfectly normal. It is a condition known as post-micro depression. Don't let it put you off: you are lucky, you are the owner of a good little micro. True, it might not look much, and the keys might stay down when you press them, and the screen might wobble when any sound is output.

Look on the bright side. The Oric is capable of bright colours, high resolution graphics and excellent sound. It is also going to grow into an enviable system. The first six months of any micro's life are fraught with trauma. That is behind us now.

The Oric printer is upon us, and is reviewed elsewhere in this issue. It is a truly wonderful little device and demonstrates the support that is beginning to build up for the Oric owner. Oric will soon follow this up with a microdrive and modem. Both these items look like arriving soon, yet Sinclair Spectrum owners have had to wait for over a year for these treats, and will have to wait longer yet.

Software is beginning to appear for the Oric. So far the majority of this has been crude and elementary, but it takes six to twelve months for professional programmers to explore a micro's capabilities fully and consequently we cannot expect to see the best yet. Hang on, it will be with us soon.

But this brings us back to the question: What do you do with your Oric? Is it just a toy or can it earn its living? For it to be useful, it requires software just like any other computer system—but so far no serious applications software has emerged. For the moment, you will have to content yourself either with games or with learning to program it yourself.

A word-processor is on the way and with that and the Oric printer we can all begin our promising careers as the next Doris Lessing or George Orwell. Such a system might not be quite as good as, say, as Olivetti or IBM word-processor, but with the current price of the Oric at £130, it will cost about a twentieth as much. What's more, those machines don't have high resolution colour graphics and ear-blasting sound.

When financial software comes on stream, we will all be able to balance our budgets, get our bills paid on time and pay off the crippling debts incurred by buying Orics, printers and software. The money has to come from somewhere.

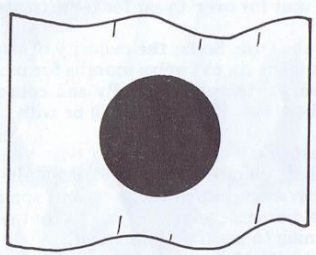
Here's to the next few years of Oric Using. Games, practical applications and communications are going to keep us all busy for quite a while. Better still, we can keep our Orics busy.

Cutprice Oric

THE ORIC 1 is now firmly established as one of the better selling micros in Britain's high street computer outlets. The company claims that sales in the UK and Europe will exceed 350,000 before the end of 1983. To help this along, especially in the face of price cutting by all the other micro manufacturers the prices have been cut.

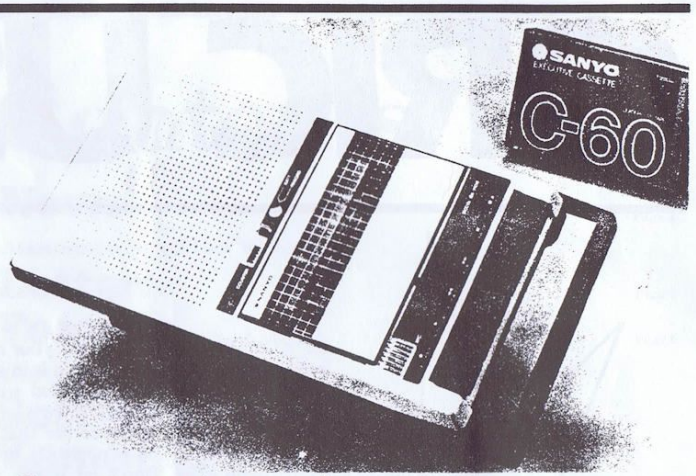
A 16K Oric will now set you back £100 — which is the same as the original launch price, while the 48K version clocks-in at £130. The prices compare favourably with the Sinclair Spectrum, but then that machine cannot Zap, Ping or Shoot.

A yen for computers



Not content with being able to sell heaps of machines to us and our European cousins — where incidentally the Oric is hot news, especially in France, Oric Products International are to sell micros to the Japanese. This is not really a case of shipping coals to Newcastle, because at the bottom end of the computer market it is the British who are more advanced. To date there has not been a single successful Japanese home computer for less than £150.

Oric Japan, the company responsible for all this aim to part with 120,000 Orics in the first year of selling to the Japanese market. Good luck to them.



Sanyo get it taped

THE SANYO DRIOI is a special tape recorder aimed at the home computer user. It can use standard C90/60 cassettes as well as the shorter tapes preferred by many users. There is a three-way switch which allows the user to choose between a normal tape recorder mode — which means you can play music, a special data mode, and a second data mode which disables the loudspeaker. This is especially useful as an Oric program sounds like nothing on earth if you are unfortunate enough to ever listen to one.

The DRIOI can be powered from the mains or four HP11 batteries. There is a battery indicator and a counter which can be reset, a useful function when searching tapes for programs. The tape recorder will cost around £39, further details from Sanyo Marubeni (UK) Ltd, 8 Greycaine Road, Creycaine Estate, Watford, Herts, WD2 4QU.

PASE joystick interface

WHAT THE ORIC lacks is a joystick port. With one games become so much more fun, and what is more you can get higher scores. Unfortunately, short of redesigning the Oric there is not much chance of there being a joystick port without some extra hardware and, of course, the software to drive it.

PASE security and computer peripherals of Hyde in Cheshire has developed just such a joystick port. It is capable of coping with two joysticks, so you can play games with a friend. The unit plugs into the Oric's printer port, and provides it with the sockets used by Atari and most major brandsof joystick. It comes supplied with software and documentation.

For further details contact PASE, 213 Market Street, Hyde, Cheshire. Telephone 061–366 5935.

Star role for programmers

MOGUL SOFTWARE is a new company that plans to market cassette based software for the Oric. The company is already involved in the video and record industry and can boast names such as Bob Marley, Chuck Berry and Telly "Kojack" Savalas among its clients. The plan is that the best and most innovative programmers will also become stars.

Software on the Mogul label will be marketed worldwide through an existing distribution network. Des Dolan, Managing director of Mogul claims that the games produced will contain good play value, distinctive graphics and sound together with a "random play facility" — whatever that might be. There will also be multiple levels of difficulty.

THE INFORMATION AND TECHNOLOGY ROAD SHOW

Dear **Oric User**
I AM WRITING to you on behalf of "The Information and Technology Road Show". The ITRS is a small group of student nurses who write programs to help mentally handicapped people who live at a hospital in Derbyshire.

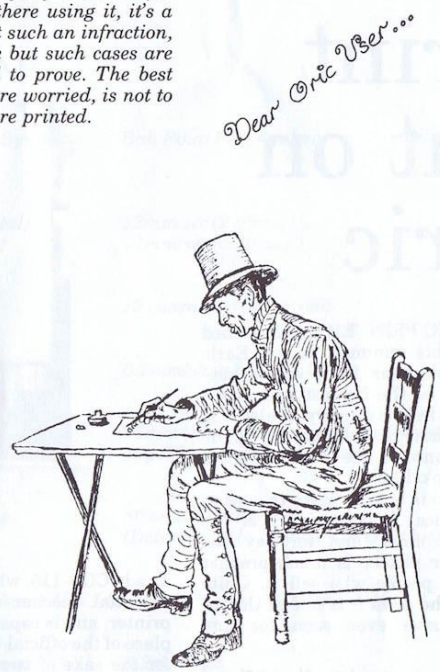
Apart from these programs being used as learning tools by these people we have now just started to build up a small reference library of medical programs (mostly Anatomy and Physiology) for reference by the nursing staff in the School of Nursing.

Most of these programs are written for the Oric and one or two for the Newbrain and the BBC micros.

If "The Information and Technology Road Show" can be of any use to you or anybody you know, please let us know at the address below.
Yours faithfully
ALAN BEARDSMORE
The Information and Technology Road Show
28 Crown Street
Derby DE 3 3UR

Dear Oric User
While I would be quite happy to share one or two of the programs I have produced with my fellow ORIC users by having them published in ORIC USER, I am a bit worried that they may possibly be sold for profit by someone else (not necessarily you). What safeguards do I have against this?
A.R. Peterson
Wimbledon

Ian Williams writes: *well, first of all, we wouldn't do a thing like that. If you send us a feature sized program and we decide to publish it in the magazine, you'll receive £5. If we think it might be good enough to be sold as a cassette package we'll pass it on to our software section who may make you an offer. The last I heard they were muttering about paying 25-30% royalty rates. As for anyone else out there using it, it's a risk. If you detect such an infraction, then you can sue but such cases are notoriously hard to prove. The best solution, if you are worried, is not to have your software printed.*



Letters.
If you have something to say, we want to hear from you. You might want to moan about something, or praise something else — providing it isn't libelous we will try and print your grievance. If enough people moan together, they can often get things done.

Criticism, of software, products, even of Oric User is welcome. By listening to your ideas, manufacturers, publishers and even ourselves can provide you with a better service. You may want to express an opinion, about the square root of minus one, or the colour of your Oric, or answer someone else's letter. You can even use our letters page to get in touch with other Oric users in your area.

Ideally letters should be typed – but as long as your handwriting is clear we may still be able to use it. It is important to leave double spaces between the lines, and to only write on one side of uniform-sized sheets of paper. Don't forget to include your name and address.

Questions
These can be on any relevant matter and we will answer as many as possible. Please only send one question on each sheet of paper and put your name and address on each sheet.

Print out on Oric

THE ORIC PRINTER was launched earlier this summer at the Earls Court Computer Fair, which happened to be the biggest such show ever held in this country. It also was the first time that the paying public got a chance to see what the Oric people look like!

But here is not the place to discuss the physical characteristics of the Oric personnel, and anyway the printer or plotter is much prettier than the people who sell it. Quite frankly the printer is one of the nicest I have ever seen for any machine.

Decked out smartly in the blue and grey livery of the Oric computer and labelled with the Oric badge, the printer sits proudly beside your faithful Oric ready to obey your every command — well, almost.

The unit costs £170, but can be purchased for less if you happen to be one of the lucky new Oric owners. These chosen few will find a voucher sitting inside the packaging of their micros entitling them to a hefty £40 off the purchase price. Making the price of a 48k Oric and a printer a competitive £270, less than the price of many other micros.

What do you get for your money? First of all is the printer/plotter itself, the necessary lead to connect it to the host Oric, a roll of paper about the same size and shape as certain types of toilet rolls but of a much higher quality and minus the little perforations. A starter set of four ball-points and a manual which is about ten times as good as the rather grotty original Oric manual, this together with a cassette showing off the printer's capabilities to the full make up the total package.
This compares with £150 for the Tandy CGP-115, which has the same essential mechanism as the Oric printer, and is capable of working in place of the official Oric product. But for the sake of twenty quid it is almost not worth buying a Tandy when you take into account the fact that a lead, which is not supplied with the Tandy device, will empty your pockets of around another ten pounds.

Also worth comparing is the ZX printer that was built for the Sinclair micros, but can be interfaced to be used with other machines. It is tatty, and so is the printout which soon becomes unreadable, because the beast uses paper that is heat sensitive. Unlike the Oric product, the ZX printer only allows two colours, grey and darker grey. The only real thing in its favour is that you could buy one and an interface for under £100.

A separate power supply for the printer means that you will need to go out and buy a plug to be able to print anything. I cannibalised one from my record player because it was late at night and I could not wait to get started. You must remember to get one when you purchase the printer.

One drawback of the printer is that having a separate, internal, power supply also requires an extra power-point. With one for the computer, a second for the television which is used as a monitor and a third for the tape-recorder used for loading and saving programs, the printer power-point becomes the fourth.

Not many people have houses that have been specially wired for computer installation, so the extra power requirement will probably require an extension socket with multiple points. These can be purchased in Woolworths and similar shops for around £5. In my home, this situation resulted in a tangle of cables that has not yet caused any accidents, but would not be a good insurance risk.

It doesn't matter if you only have a 16k Oric, though not many people do. The printer looks fine with both versions of the machine. The driver cable does tie up the expansion port, though.

Four little ball-point pens coloured black, red, green and blue actually draw on the four and a half inch wide paper. The pens are held on a carousel which holds all four at once, rotating to the position of the pen currently in use. The pen positions are referred to as 1, 2, 3 and 4, but it is up to you the user to keep track of which is which.

If you happen to swap two of the pens around, the printer doesn't actually know, so the drawings will be in a different colour. The pens don't last very long, there is enough ink in them to write for 250 metres, but that is not many program listings. We found that the black pen kept running out first, so consequently any program listings we made that were not crucial were done in any other colour.

One thing that would be really nice to see is sets of refills in one colour only. At present you have to buy all four at once. The refills should be stocked at the shop where you bought the printer, but both paper and pen refills can also be found in Tandy stores, where the items sell for and respectively.

Strictly speaking the device is a plotter and not a printer, but there is a full character set stored in ROM. If you need extra characters it should be possible to produce them yourself. The characters have a funny look to them, but are more than perfectly acceptable and of course there are both upper and lower case sets.

Movement along the X-axis is by the pen carriage or carousel moving back and forth under control of some logic. There is a possible width of around four inches — 96mm to be exact. And step resolution is about 0.2mm or some tiny fraction of an inch if you, like me, don't have the foggiest idea what 0.2mm is.

This width can accommodate a number of characters, though usually 40 or 80, set by a command from software. There is a choice of 15 different character sizes varying from the enormous to the minuscule.

Y-axis movement is achieved by the paper feed mechanism winding its weary way back and forth. Some amazingly long distances can be achieved and in theory the maximum possible length of a line drawn on the paper using the plotter is infinite. In practice the sun will explode, the earth will be destroyed and the universe will have collapsed by then — more likely, though, the pen will have run out as it only writes for 250 metres.

One thing about the plotter is very impressive. If you try drawing a picture and returning the carousel to the start position and then drawing again, you will get an idea of how accurately the plotter works. I am pleased to report that this test worked perfectly, even with the tiniest detail imaginable. However we did experience a number of "glitches", where a particular character was misread.

This glitching problem resulted in an undefined character, incidentally it was the same one each time, being printed when in text mode, and weird things happening when in graphics mode. These two modes can be thought of as printer and plotter modes. The glitch is due to the keyboard interrupt. Disconnect it using CALL*E6CA and restart it using CALL*E804.

CONCLUSION

The Oric printer makes the Oric look like a much better machine. It is smart and works well despite the glitch problem. It is also a very reasonable price.

ORIC COLOUR PRINTER

Specifications

Printing/Plotting System	*Ball Point Pen, 4 colour*
Plotting Speed	
(Horizontal)	*52mm/sec (2.05ips)*
(Vertical)	*73mm/sec (3.80ips)*
Printing Speed	*12 characters per second*
Resolution	*0.2mm/step (0.00787 inch)*
Effective Plotting Range	*96mm (3.804 inch) x axis Divided into 480 steps (No limit in y direction)*
Characters per Line	*80 or 40 (Text Mode)* *(Determined by Software in Graphics Mode)*
*Characters per Line = INT (480/(n+1) *6) for 0< = n< = 15*	
Accuracy (repetition)	*0.2mm max*
(Movement)	*0.3mm max*
(Distance)	*0.5% max (X-axis)* *1% (Y-axis)*
Pen Life	*250 metres (825 feet)*
Parallel Interface	*8-bit parallel. Uses BUSY handshaking. STROBE, and ACKNOWLEDGE*
Selectable Modes	
Self Test	*Prints 96 ASCII character set in 4 colours*
Text Mode	*Normal Serial and Parallel Printing*
Graphics Mode	*Image Plotting using the Various commands*

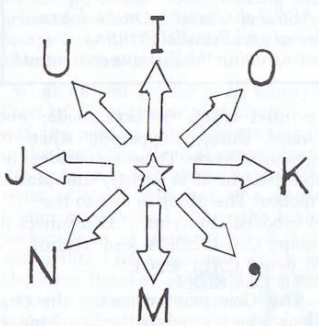
ZOMBIE ISLAND

It seems innocent enough—a haven even—a small tropical island basking in the warm Caribbean sun. But soon after you stumble ashore you realise that something is stirring. The Zombies are on the move!

A program by Margot Tomlinson of Chingford

Here's a de-luxe version of an old favourite. You are a shipwrecked sailor (designated on the screen by *) seeking to evade the hideous undead zombies. The zombies (designated by z) continually home on your position but as they are creatures of very little brain, they only travel in straight lines and should they walk into a puddle (shown as O) will dissolve. So all you have to do is to keep the puddles between you and the zombies; but beware lest you fall in one or in the ocean yourself!

Keys are used to move according to the following diagram:



Any other key will leave you in the same spot.

Important variables: The program generates a random number of zombies, the X,Y co-ordinates of which are stored in arrays ZX and ZY. Should a zombie fall in a puddle, its X co-ordinate is set to 99, ruling it out of the game. A count (LZ) is kept of the number of live zombies and when this equals zero, you've won. Note the use of white noise to produce the splashing sounds!

ZOMBIE ISLAND: DOCUMENTATION

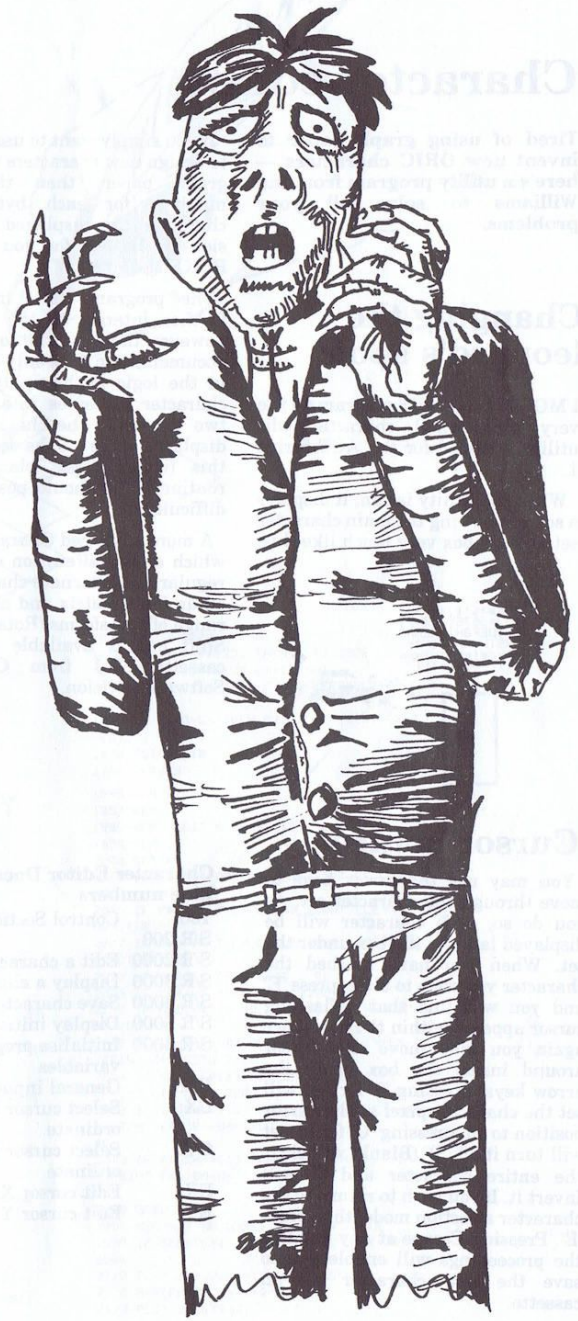
3000 Initialisation
Arrays: ZX: Zombie X co-ordinate
ZY: Zombie Y co-ordinate
A$: Graphics string
YS: Your Score
ZS: Zombies' Scores
50-60: Randomise/Initialise
100-150: Set up screen frame, display score
N: General purpose variable
R: Random number maximum
RM: Random number minimum
S/R 5: Random number generated between limits
NP: Number of puddles
PX: Puddle X co-ordinate
PY. Puddle Y co-ordinate
200-250: Plot puddles
NZ: Number of zombies
RN: Random number
YX: Player's X co-ordinate
YY: Player's Y co-ordinate
LZ: Number of live zombies
290-380: Plot zombies
1000-1075: Get and process player's move
D$: Input variable for direction
OX: Player's last X co-ordinate
OY: Player's last Y co-ordinate
1110-1170: Checks whether player walks into an obstacle
1200-1280: Move a zombie
1300-1400: Check if zombie hits anything
M: General purpose variable
1440-1450: Check if any zombies left
1500-1510: Music for zombies' win
1600-1630: Music for your win
2000-2040: Check if another game wanted

```
1 GOTO 70
5 RN=INT(RND(1)*RM) : IF RN<RM THEN GOTO
7 RETURN                                  5
10 REM ZOMBIE ISLAND
20 REM MARGOT TOMLINSON
30 REM JULY 1983
40 REM
70 GOSUB 3000
96 REM
97 REM
98 REM START NEW GAME
99 REM
100 CLS
110 PLOT 3,0,"YOUR":PLOT 9,0,A$:PLOT 32,
0,"ZOMBIE"
120 PLOT 3,1,"SCORE":PLOT 32,1,"SCORE"
130 PLOT 2,2,STR$(YS):PLOT 31,2,STR$(ZS)
140 FOR N=1 TO 20:PLOT 9,N,CHR$(126):PLO
T 30,N,CHR$(126):NEXT N
150 PLOT 9,21,A$
170 REM
180 REM PLOT PUDDLES
190 REM
200 R=40 : RM=20 : GOSUB 5 : NP=RN
210 FOR N=1 TO NP
220 R=30 : RM=10 : GOSUB 5 : PX=RN
230 R=21 : RM=1 : GOSUB 5 : PY=RN
240 PLOT PX,PY,"O"
250 NEXT N
260 REM
270 REM  PLOT & STORE ZOMBIES
280 REM
290 R=20 : RM=5 : GOSUB 5 : NZ=RN
300 FOR N=0 TO RN-1
310 R=30:RM=10 : GOSUB 5 : ZX(N)=RN
320 R=21 : RM=1 : GOSUB 5 : ZY(N)=RN
330 PLOT ZX(N),ZY(N),"Z"
340 NEXT N
350 R=30 : RM=10 : GOSUB 5 : YX=RN
360 R=21 : RM=1 : GOSUB 5 : YY=RN
370 PLOT YX,YY,"*"
380 LZ=NZ
996 REM
997 REM
998 REM MAIN LOOP
999 REM
1000 PLOT 9,24,"YOUR MOVE
     "
1010 OX=YX : OY=YY
1020 GET D$
1030 IF D$="O" OR D$="K" OR D$="," THEN
YX=YX+1
1040 IF D$="N" OR D$="J" OR D$="U" THEN
YX=YX-1
1050 IF D$="U" OR D$="I" OR D$="O" THEN
YY=YY-1
1060 IF D$="N" OR D$="M" OR D$="," THEN
YY=YY+1
1070 PLOT OX,OY," "
1075 PLOT 9,25,"          "
1080 REM
1090 REM CHECK FOR LETHAL MOVE
1100 REM
1110 N=SCRN(YX,YY)
1120 IF N=79 THEN PLOT 9,23,"RIGHT INTO
A PUDDLE":PLAY 0,1,1,5000
1125 IF N=79 THEN PLOT 9,24,"I'M AFRAID
YOU'VE DROWNED":GOTO 1510
1130 IF N=126 THEN PLOT 9,23,"THE LIMPID
 WATERS OF THE":PLAY 0,1,1,
5000
1140 IF N=126 THEN PLOT 9,24,"CARRIBEAN
CLAIM ANOTHER VICTIM!":GOTO
 1510
1150 IF N=90 THEN PLOT 9,23,"STRAIGHT IN
TO THE ARMS OF A"
```



```
1160 IF N=90 THEN PLOT 9,24,"WAITING ZOM
BIE...":GOTO 1500
1170 PLOT YX,YY,"*"
1180 REM
1190 REM  ZOMBIES MOVE
1200 REM
1205 PLOT 9,24,"HERE COME THE ZOMBIES!!!"
1210 FOR N=0 TO NZ-1
1215 PLOT 9,23,"
 "
1220 IF ZX(N)=99 THEN GOTO 1400
1230 OX=ZX(N):OY=ZY(N)
1240 PLOT OX,OY," "
1250 IF OX<YX THEN ZX(N)=ZX(N)+1
1260 IF OX>YX THEN ZX(N)=ZX(N)-1
1270 IF OY<YY THEN ZY(N)=ZY(N)+1
1280 IF OY>YY THEN ZY(N)=ZY(N)-1
1290 REM
1300 REM CHECK FOR ZOMBIE DROWNED
1310 REM
1320 M=SCRN(ZX(N),ZY(N))
1330 IF M=79 THEN PLOT 9,23,"SPLASH GOES
 A ZOMBIE!!!"
1335 IF M=79 THEN ZX(N)=99 : LZ=LZ-1 :PL
AY 0,1,1,3000:WAIT 75:GOTO
1400
1340 IF M=42 THEN PLOT 9,24,"A ZOMBIE DU
N GOTCHA!!       ":GOTO 1500

1345 IF M=90 THEN ZX(N)=OX : ZY(N)=OY
1350 PLOT ZX(N),ZY(N),"Z"
1360 PLAY 1,0,1,1000
1400 NEXT N
1410 REM
1420 REM CHECK IF ZOMBIES LEFT
1430 REM
1440 IF LZ=0 THEN GOTO 1600
1450 GOTO 1000
1460 REM
1470 REM   THE ZOMBIES WIN
1480 REM
1500 PLAY 4,0,0,0:MUSIC 3,0,6,6:WAIT 75:
MUSIC 3,0,4,6:WAIT 75:MUSIC
 3,0,3,6
1502 WAIT 100:PLAY 0,0,0,0
1510 ZS=ZS+1 : GOTO 2000
1520 REM
1530 REM   YOU WIN
1540 REM
1600 PLOT 9,25,"
 "
1605 PLOT 9,23,"
 "
1610 YS=YS+1 : PLOT 9,24,"YOU SHO' WHUPP
ED DEM ZOMBIES!!!"
1615 PLAY 4,0,0,0
1620 MUSIC 3,4,3,4:WAIT 20:MUSIC 3,4,5,5
:WAIT 30:MUSIC 3,4,8,6:WAIT
 60
1630 PLAY 0,0,0,0
1640 REM
1650 REM CHECK FOR ANOTHER GAME
2000 REM
2005 PLOT 9,26,"ANOTHER GAME(Y/N)?"
2010 GET D$
2020 IF D$="Y" THEN GOTO 100
2030 IF D$="N" THEN CLS : STOP
2040 GOTO 2010
2996 REM
2997 REM
2998 REM **** INITIALISATION ****
2999 REM
3000 DIM ZX(50),ZY(50)
3010 A$="":FOR N=1 TO 22:A$=A$+CHR$(126)
:NEXT N
3020 YS=0:ZS=0
3030 PRINT CHR$(17);
3040 MUSIC 1,0,9,0
3050 MUSIC 2,0,1,0
3060 DOKE #FB,DEEK(#276):DOKE #FD,DEEK(#
276)
3900 RETURN
```



Character editor

Tired of using graph paper to invent new ORIC characters — here's a utility program from Ian Williams to solve all your problems.

Changing the leopard's spots

I MODELLED THIS program on the very professional character edit utility provided for the ACT Sirius 1.

When the utility is run, it displays a screen showing the main character set above a box very much like this:

```
 !"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^
 abcdefghijklmno
pqrstuvwxyz{|}
```

If you simply want to use the editor to design new characters rather like graph paper, then the POKE numbers for each byte of the character are displayed down the side of the box for you to use in POKE statements.

The program itself may be of further interest to the reader — however, there is little to add to the documentation. The only complexity in the logic is the analysis of the character set bytes into powers of two to enable the character to be displayed large on the screen. Even this (which takes place in sub-routine 3000) should pose no great difficulty.

A more advanced Character Editor which allows alteration of both the regular and alternate character sets on an 8 x 8 matrix and also a wider range of transforms (Rotate, Mirror, Shift etc) is available on Utility cassette No.1 from CCC Ltd's Software Division.

Cursor control

You may use the arrow keys to move through the character set. As you do so, each character will be displayed large in the box under the set. When you have reached the character you wish to alter, press 'E' and you will find that a flashing cursor appears within the box. Once again you may move this cursor around inside the box using the arrow keys. Pressing 'S' for Set will set the character pixel at the cursor position to on, pressing 'U' for Unset will turn it off. 'B' (Blank) will clear the entire character and 'I' will Invert it. If you wish to return to the character selection mode, then press 'E'. Pressing Escape at any point in the proceedings will enable you to save the new character set on cassette.

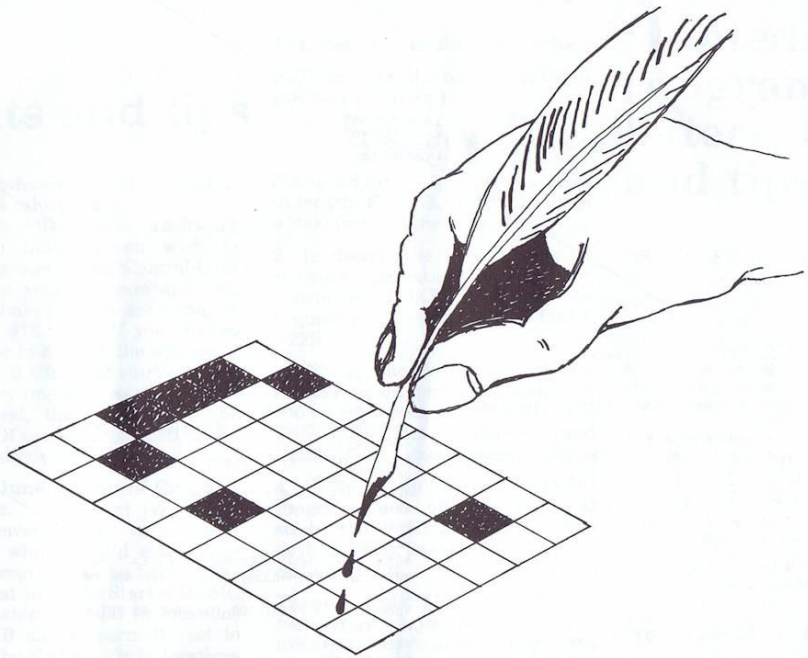
Character Editor Documentation

Line numbers

100	Control Section
S/R 200	
S/R 2000	Edit a character
S/R 3000	Display a character
S/R 4000	Save character set to tape
S/R 5000	Display initial screen
S/R 6000	Initialise program variables
K$:	General input field
CX:	Select cursor X co-ordinate
CY:	Select cursor Y co-ordinate
EX:	Edit cursor X co-ordinate
EY:	Edit cursor Y co-ordinate
D$, U$, R$, L$:	Cursor control strings
AD	Current character start address
BA	Edit mode address field
CA	Edit character set start address

This is set for the 48K model. For 16K ORICs, line 6000 must be altered. CA actually points to ASCII 32 — the first printable character.

M,N,X,Y,Z	General purpose variables
F$	File name
C (M,N)	Character storage array (not used on this version).



```
10 REM  CHARACTER SET EDIT
20 REM  IAN WILLIAMS
30 REM  COPYRIGHT CCC LTD 1983
40 REM
50 REM
96 REM
97 REM
98 REM   CONTROL SECTION
99 REM
100 GOSUB 6000' INITIALISE
110 GOSUB 5000' SET UP INITIAL SCREEN
120 GOSUB 1000' SELECT A CHARACTER
130 GOSUB 4000' TAPE SAVE
140 IF K$="Y" THEN GOTO 110
150 STOP
996 REM
997 REM
998 REM  SELECT A CHARACTER
999 REM
1000 GOSUB 3000'PLOT LATEST CHAR
1010 GET K$
1020 IF K$=CHR$(27) THEN GOTO 1900
1030 IF K$="E" THEN GOTO 1800
1040 IF ASC(K$)<8 OR ASC(K$)>11 THEN GOT
O 1010
1050 IF K$=CHR$(8) AND CX>0 THEN AD=AD-8
:CX=CX-1:PRINT K$;:GOTO 100
0
1060 IF K$=CHR$(9) AND CX<11THEN AD=AD+8
:CX=CX+1:PRINT K$;:GOTO 100
0
1070 IF K$=CHR$(10) AND CY<7 THEN AD=AD+
96:CY=CY+1:PRINT K$;:GOTO 1
000
1080 IF K$=CHR$(11) AND CY>0 THEN AD=AD-
96:CY=CY-1:PRINT K$;:GOTO 1
000
1090 GOTO 1010
1800 PRINT LEFT$(D$,10-CY);
1810 IF CX>3 THEN PRINT LEFT$(L$,CX-3);
1813 IF CX<3 THEN PRINT LEFT$(R$,3-CX);
1815 EX=0:EY=0:BA=AD
1820 GOSUB 2000
1830 IF K$=CHR$(27) THEN GOTO 1900
1840 PRINT LEFT$(U$,EY-CY+10);
1850 IF CX<3+EX THEN PRINT LEFT$(L$,EX-C
X+3);
1860 IF CX>3+EX THEN PRINT LEFT$(R$,CX-3
-EX);
1870 GOTO 1010
1900 RETURN
1996 REM
1997 REM
1998 REM  EDIT A CHARACTER
1999 REM
2000 GET K$
2010 IF K$="E" OR K$=CHR$(27) THEN GOTO
2300
2015 IF K$="B" THEN GOSUB 2500' CLEAR CH
ARACTER
2017 IF K$="I" THEN GOSUB 2600'INVERT CH
ARACTER
2022 IF K$="S" THEN GOTO 2100
2025 IF K$="U" THEN GOTO 2200
2030 IF ASC(K$)<8 OR ASC(K$)>11 THEN GOT
O 2000
2040 IF K$=CHR$(8) AND EX>0 THEN EX=EX-1
:PRINT K$;:GOTO 2000
2050 IF K$=CHR$(9) AND EX<5 THEN EX=EX+1
:PRINT K$;:GOTO 2000
2060 IF K$=CHR$(10) AND EY<7 THEN EY=EY+
1:PRINT K$;:BA=BA+1:GOTO 20
00
2070 IF K$=CHR$(11) AND EY>0 THEN EY=EY-
1:PRINT K$,:BA=BA-1:GOTO 20
00
2080 GOTO 2000
2090 REM SET OR UNSET SCREEN POINTS
2100 IF SCRN(EX+17,EY+13)=127 THEN GOTO
2000
2110 PLOT EX+17,EY+13,CHR$(127)
2120 X=PEEK(BA):X=X+2^(5-EX)
2123 PLOT 25,EY+13,"     "
```

```
2125 PLOT 25,EY+13,STR$(X)
2130 POKE BA,X
2140 GOTO 2000
2200 IF SCRN(EX+17,EY+13)<>127 THEN GOTO
 2000
2210 PLOT EX+17,EY+13," "
2220 X=INT(PEEK(BA)):Y=INT(2^(5-EX)):X=X
-Y
2223 PLOT 25,EY+13,"    "
2225 PLOT 25,EY+13,STR$(X)
2230 POKE BA,X
2240 GOTO 2000
2497 REM
2498 REM  CANCEL WHOLE CHARACTER
2499 REM
2500 FOR N=0 TO 7
2510 POKE (AD+N),0:PLOT 17,13+N,"      "
:PLOT 25,N+13,"      ":PLOT 2
5,N+13,"0"
2520 NEXT N
2540 RETURN
2597 REM
2598 REM  INVERT WHOLE CHARACTER
2599 REM
2600 FOR N=0 TO 7
2610 X=PEEK(AD+N):X=63-X:POKE AD+N,X
2625 NEXT N
2630 GOSUB 3000
2640 RETURN
2900 RETURN
2996 REM
2997 REM
2998 REM   PLOT LATEST CHARACTER
2999 REM
3000 FOR N=13 TO 20:PLOT 17,N,"      ":N
EXT N
3005 FOR N=0 TO 7:Y=32
3007 PLOT 25,13+N,"    "
3010 X=PEEK(AD+N):PLOT 25,13+N,STR$(X)
3020 FOR M=5 TO 0 STEP-1
3030 Z=INT(X/Y):C(M,N)=Z
3040 IF Z=1 THEN PLOT 22-M,13+N,CHR$(127
)
3050 X=Y*(X/Y-Z):Y=Y/2
3060 NEXT M:NEXT N
3900 RETURN
3996 REM
3997 REM
3998 REM STORE CHARACTERS ON TAPE
3999 REM
4000 CLS:PRINT:PRINT:PRINT
4010 PRINT "DO YOU WISH TO STORE THE CHA
RACTER"
4020 PRINT "SET ON TAPE(Y/N)?"
4030 GET K$:IF K$="N" THEN GOTO 4800
4040 PRINT : PRINT
4043 INPUT "PLEASE ENTER A FILE NAME";F$
4045 PRINT
4050 PRINT : PRINT "PLEASE ENSURE THAT T
HE TAPE DECK"
4060 PRINT "IS SET TO RECORD"
4070 PRINT : PRINT "PRESS ANY KEY TO SAV
E":GET K$
4073 PRINT
4075 PRINT "SAVING CHARACTER FILE: ";F$
4080 CSAVE F$,A46336,E47103 : REM 48K OR
IC
4800 PRINT : PRINT "DO YOU WANT TO RETUR
N TO EDIT(Y/N)"
4810 GET K$
4900 RETURN
4996 REM
4997 REM
4998 REM  SET UP INITIAL SCREEN
4999 REM
5000 CLS:N=32
5005 PLOT 3,1,"ORIC USER Character Set E
ditor"
5010 FOR Y=3 TO 10
5020 FOR X=14 TO 25
5030 PLOT X,Y,CHR$(N)
5040 N=N+1
5050 NEXT X
5060 NEXT Y
5070 REM  PLOT BORDER
5075 B$=CHR$(126)
5080 FOR X=16 TO 23:PLOT X,12,B$:PLOT X,
21,B$:NEXT X
5090 FOR Y=13 TO 20:PLOT 16,Y,B$:PLOT 23
,Y,B$:NEXT Y
5100 CX=0:CY=0:EX=0:EY=0:AD=CA
5110 PRINT LEFT$(D$,3);LEFT$(R$,15);
5900 RETURN
5996 REM
5997 REM
5998 REM  PROGRAM INITIALISE
5999 REM
6000 CA=46336 : REM 48K ORIC
6005 INK 7:PAPER 0
6010 DIM C(5,7):U$="":D$="":L$="":R$=""
6020 FOR N=1 TO 40:U$=U$+CHR$(11):D$=D$+
CHR$(10):L$=L$+CHR$(8):R$=R
$+CHR$(9)
6030 NEXT N
6900 RETURN
```


Hints and tips

C. Aprahamian of Stroud, Gloucestershire, says:
Readers may like to know a software protection hint. If you wish to prevent a user using Control-C to break into your program and list, you can change a software vector at locations #IB, #IC. If you change this vector to #EA59, the address of the ORIC BASIC cold start routine, then everytime Control-C or Reset are pressed, the memory will be cleared. Of course, this location can be set to any vector you want.

June Seaton of Croydon points out:
I've discovered two very useful locations which should enable two BASIC programs to be held in the machine at once. The Start of BASIC address which is held in locations #9A, #9B and is normally set to #0501. The End-hold is in locations #9C, #9D. Providing that you don't overwrite one program with variables, you can flip between two programs by DOKEing these locations.

If you've got any tips or advice you'd like to share — and you'd like to see your name in print — just send them in, folks.

A. Desai of Skelmersdale points out:
To get better random numbers out of ORIC BASIC, the 'seed' for the random number generator can be found in the four bytes from locations #FB to #FE. If you POKE a number into these locations at the start of processing it will determine the random sequence which follows, otherwise you'll get the same sequence every time you run the program.

Tarquin adds: *I suggest that the seed locations be DOKEd with the contents of the ORIC timer. This is two locations, #275 and #277, which are incremented every hundredth of a second. Thus DOKE #FB, DEEK (#276): DOKE #FD, DEEK (#276) will do the trick.*

J.M. Bell of Rotherham writes:

1. To print on the top line of ORIC 48K

```
10 A$="The Title"
20 FOR N=#BB83 TO #BB83+36:
   POKE N,32:NEXT N
30 FOR N=1 TO LEN(A$):POKE #BB83+N,
   ASC(MID$(A$,N,1)):NEXT N
```

String A$ can be up to 36 characters in length. For 16K ORICS, subtract #8000 from all the addresses.

2. To disable the Reset button for software protection include the instruction POKE 555,64 at the beginning of the program or CALL #22B.

3. To find the start and finish of a machine code program, assuming you've just loaded it, DEEK (95) gives the start address and DEEK(97) the end address.

4. To speed up an ORIC BASIC Program turn off the keyboard strobe by calling #E6CA. The keyboard may be turned on again by calling #E804.

Tarquin adds: *Always remember to turn off the keyboard in this way when printing to avoid random bad characters.*

Wanted: your programs, letters, hints and tips

ORIC USER is your magazine. We are interested to see any software you have. No matter how simple, it will be of interest to some of our readers. Remember that it is an offence to copy programs out of other magazines and you may be prosecuted for doing this.

If you don't have an Oric printer, we will accept handwritten listings but would prefer software on tape. Remember to tell us what speed you used to record the program. Handwritten programs are more liable to error so please check these carefully before sending them.

We will pay £5 for a feature program

TARQUIN IN A RELAXED MOOD...

Can you survive the journey through 7000 km of deadly methane to rescue an astronaut from the frozen hell of Neptune's surface? Only your skill and quick wits can rescue her from a terrible death... Another Real-Time game from Ian Williams.

NEPTUNE—EIGHTH PLANET OF THE SOLAR SYSTEM

Mean distance from sun: 4496.7 million km
Sidereal period: 164.8 years
Rotation period: 22 hours
Axial inclination: 28° 48'
Diameter: 49500 km
Density: (water =1): 1.77
Mass (earth =1): 17.2
Escape velocity: 23.9 km/sec
Surface gravity: 1.2g
Surface temperature: —220 °C
Atmosphere: Mostly methane, mixed with hydrogen. Water-ice clouds present
Discovered: 1846

RESCUE ON NEPTUNE

IN THIS GAME you have to guide a remotely controlled module to the surface of Neptune, rescue a trapped astronaut and return to orbit. Your vessel is moved from side to side by using the J and K keys. The main drive is turned on by the B key and off by the C. But that's not all...

First of all, fuel. Attitude control uses one unit of fuel for each burn, whereas the main drive uses two. For weight reasons, you cannot carry all the fuel you need for take-off so you have to refuel the fusion drive constantly by passing through water-ice clouds which look like this:

Beware: as your fuel level increases, your vehicle's stability declines. If it passes below zero, you will crash out of control in the atmosphere. Audible warnings sound when fuel or stability get low.

Neptune's atmosphere produces vicious thunderstorms, depicted as:

Should you hit one of these, you'll be destroyed. You may save time and jump over obstacles by passing through hydrogen vortices, shown as:

You may play multiple games with the same scenario, but note that your individual stability level declines with each new game. Note that fuel probability declines as you go down, while storm probability increases.

The program works by setting up a map of the atmosphere. You'll notice that this takes some time! This is then scrolled up or down depending on whether you are landing or taking off. This map is set up in 150 x 40 character strings (RL$) by subroutine 100. Certain constant portions such as the planetary surface are defined in S/R 30000. This also redefines the character set.

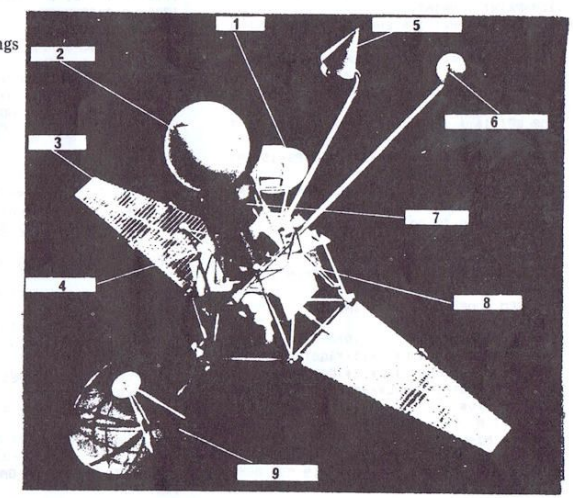
This program makes use of the Oric's bi-directional scrolling ability and trades speed in setting up the scenario for speed later when you are running the game. You'll find it's as much a real-time maze game as anything. Happy landings!

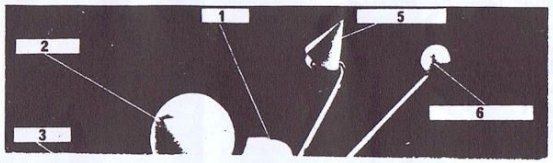
DOCUMENTATION

S/R 100: Set up a new sky scenario if wanted
S/R 400 Display top of screen data
1000-1130 Initial game screen
2000-2050: Attitude control
2055-2100: Check for impact
2110-2130: Process burn/no burn
3000-3030: Scroll scenario up or down
3035-3060: Check for crash or orbit
4000-4080: Vehicle crashed
5000-5250: Landing/rescue sequence
6000-6090: Orbit achieved

S/R 30000: Program initialise, change character set
S/R 40000: Data statement for changing character set
SA: Character set start address (46080 for 48k)
N,M,R,X: General purpose variables
A1$, A2$: Astronaut characters
C$: Thunder cloud character
B$: Fuel cloud character
V$: Vortex character
P$: Rocket character

Array:
RL$: Sky scenario strings
L$, Z$: Work string
UP$, DOWN$, U$: Cursor control strings
S$: Blank string
I$: Input character
PA: Fuel cloud probability
PB: Thunderstorm probability
TI: time so far
LL: Next string to scroll
LA: Landing indicator
AP: Astronaut rescue indicator
DE: Crash indicator
OX: Stored X control
NT: Move distance
PC: Vortex probability
PS, PV: Probability work-fields
RY: Rocket Y co-ordinate
RX: Rocket X co-ordinate
NY: Look-ahead Y co-ord
BT: Best time so far
SL: Stability limit (initial)
NL: Stability limit (running)
GN: Game number
FU: Fuel level (running)
FL: Fuel level (initial)
BI: Burn on switch



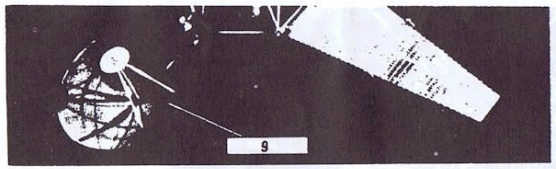



```
5 GOSUB 30000:GOTO 1000
10 REM NEPTUNE RESCUE
20 REM AUTHOR IAN WILLIAMS
30 REM COPYRIGHT
40 REM CCC LTD. 1983
96 REM
97 REM
98 REM SET UP MISSION SCENARIO
99 REM
100 CLS : PRINT : PRINT
102 PRINT "DO YOU WANT A NEW SCENARIO(Y/
N)?":GET I$
104 IF I$="N" THEN RETURN
106 IF I$<>"Y" THEN GOTO 102
108 CLS
110 PRINT TAB(12) "SCENARIO GENERATION"
120 PRINT : PRINT
130 PRINT TAB(12)"SCENARIOS ARE GRADED 1
-8"
140 PRINT TAB(12)"IN ORDER OF INCREASING
"
150 PRINT TAB(12)"DIFFICULTY - ENTER 1 T
O 8"
160 GET I$ : IF VAL(I$)>8 OR VAL(I$)<1 T
HEN GOTO 160
170 PA=10+VAL(I$)*5:PB=5:PC=10:PS=15+PA:
PV=PA+PB
180 PRINT : PRINT "THIS WILL TAKE A MINU
TE"
190 REM LOOP TO DO WHOLE SKY
195 CALL #E6CA
200 FOR N=27 TO 120
210 L$="":IF N > 50 AND PA>5 THEN PA=PA-
1:PB=PB+1
215 REM LOOP TO DO ONE LINE
220 FOR M=1 TO 40
230 RN=RND(1)*200
240 IF RN>PS THEN L$=L$+" ":GOTO 280
250 IF RN<PA THEN L$=L$+C$:GOTO 280
260 IF RN<PUTHEN L$=L$+B$:GOTO 280
270 L$=L$+U$
280 NEXT M
290 RL$(N)=L$
300 NEXT N
320 RY=6:BT=0:SL=9.1:GN=0:FL=19
330 CALL #E804
390 RETURN
396 REM
397 REM
398 REM DISPLAY MISSION DATA
399 REM
400 PLOT 0,0,S$
405 PLOT 1,0,"STAB.":PLOT 7,0,LEFT$(STR$
(NL),4)
410 PLOT 15,0,"FUEL":PLOT 19,0,STR$(FU)
420 PLOT 24,0,"TIME":PLOT 28,0,STR$(TI)
430 IF BI=1 THEN POKE 48073,12:PLOT 33,0
,"BURN"
433 REM
435 REM   AUDIBLE FUEL/STAB WARNING
437 REM
440 IF FU<5  THEN PLAY 7,0,3,2000 ELSE I
F NL <1 THEN PLAY 7,0,4,100
0
450 RETURN
996 REM
997 REM
998 START OF GAME
999 REM
1000 TI=0 : BI=0 : LL=27
1005 LA=0:AP=0:DE=0
1010 GOSUB 100
1020 CLS :SL=SL-.1 : GN=GN+1:NL=SL
1025 FL=FL+1 : FU=FL
1030 FOR N=1 TO 26:PRINT RL$(N);:NEXT
1040 RX=INT(RND(1)*39):RY=13
1050 PLAY 0,0,0,0 : GOSUB 400
1055 MUSIC 1,2,4,0 : MUSIC 2,4,7,0 : MUS
IC 3,6,9,0
1060 PLOT RX,RY,P$
1065 POKE 48120,12
1070 PLOT 1,2,"ALL SYSTEMS NOMINAL"
1075 IF BT>0 THEN PLOT 1,3,"BEST TIME TH
IS SCENARIO":PLOT 27,3,STR$
(BT)
1080 PLOT 1,4,"PRESS ANY KEY TO INITIATE
 DESCENT"
1085 PLOT 10,6,"GAME":PLOT 15,6,STR$(GN)
1090 GET I$
1095 PLOT 1,4,S$:PLOT 1,6,S$
1100 PLOT 1,2,"DESCENT BURN INITIATED"
1110 WAIT 120
1120 PLOT 1,2,"ENGINES OFF - DROGUE CHUT
ES DEPLOYED"
1130 WAIT 120
1996 REM
1997 REM

1998 REM   START OF GAME LOOP
1999 REM
2000 I$=KEY$ : OX=RX : NT=1
2003 IF I$="" THEN GOTO 2050
2010 IF I$<"J" THEN GOTO 2110
2012 REM
2013 REM      ATTITUDE CONTROL
2014 REM
2015 IF FU<1 THEN GOTO 3000
2020 IF I$="J" AND RX<1 THEN GOTO 2000
```


```
2025 IF I$="K" AND RX>37 THEN GOTO 2000
2030 IF I$="J" THEN RX=RX-1:FU=FU-1:NL=N
L+.1
2040 IF I$="K" THEN RX=RX+1:FU=FU-1:NL=N
L+.1
2050 IF BI=1 THEN NY=RY-1 ELSE NY=RY+1
2052 REM
2053 REM CHECK FOR OBJECT
2054 REM
2055 X=SCRN(RX,NY) : IF X=32 THEN GOTO 3
000
2056 REM
2057 REM      ICE CLOUD
2058 REM
2060 IFX=123THENFU=FU+10:NL=NL-1:GOSUB 4
00:PLAY 3,0,1,250:IF NL<.1T
HEN DE=1
2063 REM
2065 REM      DEADLY THUNDERSTORM
2067 REM
2070 IF X=122 THEN DE=1
2073 REM
2075 REM      VORTEX ACCELERATION
2077 REM

2080 IF X=124 AND BI=1 THEN NT=NT+2:NY=N
Y-2:PLAY 7,0,2,400:GOTO 205
5
2085 IF X=124 AND BI=0 THEN NT=NT+2:NY=N
Y+2:PLAY 7,5,2,500:GOTO 205
5
2086 REM
2088 REM      LANDING!!!!
2089 REM
2090 IF X=126 THEN GOTO 5000
2100 GOTO 3000
2103 REM
2105 REM CANCEL  -     - ITION
2107 REM
2110 IF I$="C" AND BI=1 THEN PRINT DOWN$
;:LL=LL+27:BI=0:GOTO 2000
2113 REM
2115 REM   INITIATE BURN CONDITION
2117 REM
2120 IF I$="B" AND FU>0 AND BI=0 THEN BI
=1:PRINT UP$;:LL=LL-27:LA=0
2130 GOTO 2000
2996 REM
2997 REM
2998 REM MOVE SCENARIO
2999 REM
3000 TI=TI+1:REM INCREMENT TIMER
3002 PLOT OX,RY," ":REM DEPLOT ROCKET
3005 REM
3007 REM THIS LOOP SCROLLS 'NT' TIMES
3009 REM
3010 FOR M=1 TO NT
3015 IF LL<1 THEN M=NT:GOTO 3030
3020 PRINT RL$(LL);
3023 IF BI=1 THENPOKE 48040,INT(LL/25):L
L=LL-1:PRINTU$;:FU=FU-2:NL=
NL+.2
3025 IF FU<0 THEN FU=0
3026 GOSUB 400
3027 IF BI=0 THEN POKE 49040,INT(LL/25):
LL=LL+1
3030 NEXT M
3031 REM
3032 REM CHECK FOR DESTRUCTION/ORBIT
3033 REM
3035 IF DE=1 THEN GOTO 4000:REM DESTROYE
D?
3040 PLOT RX,RY,P$:REM PLOT ROCKET
3050 IF FU<1 AND BI=1 THEN BI=0:PRINT DO
WN$;:LL=LL+27:REM FUEL'S RU
N OUT?
3055 IF LL<1 THEN GOTO 6000:REM IN ORBIT
?
3060 GOTO 2000: REM RETURN TO START OF L
OOP
3996 REM
3997 REM
3998 REM   LETHAL COLLISION
3999 REM
4000 REM
4005 WAIT 10 : EXPLODE
4010 FOR N=1 TO 10
4020 PLOT RX,RY,P$:WAIT 10
4030 P     RX,RY," ":WAIT 10
4035 NEXT N
4040 POKE 48640,12
4050 PLOT 1,RY+2,"VESSEL DESTROYED
                 "
4060 PLOT 1,RY+4,"DO TRY TO BE MORE CAR
EFUL IN FUTURE"
4070 PLOT 1,RY+6,"PRESS ANY KEY TO TRY A
GAIN"
4080 GET I$: GOTO 1000
4996 REM
4997 REM
4998 REM   LANDING SEQUENCE
4999 REM
5000 IF LA=1 THEN GOTO 2000
5005 POKE 48560,1
```



```
5010 IF AP=1 THEN GOTO 5100:REM RESCUE
5020 AP=1
5023 REM
5025 REM    RUNNING ASTRONAUT LOOP
5027 REM
5030 FOR N=1 TO RX-1
5040 PLOT N,RY,A1$ : WAIT 6
5045 PLAY 0,1,1,20
5050 PLOT N,RY,A2$ : WAIT 6
5055 PLAY 0,1,1,20
5060 PLOT N,RY," " : WAIT 6
5070 NEXT N
5075 REM
5080 PLOT 5,RY+2,"3 CHEERS FOR NASA!!!"
5090 WAIT 120
5100 LA=1
5110 IF FU<1 THEN GOTO 5200:REM ANY FUEL
?
5115 REM   FUEL LEFT IN TANKS
5120 POKE 48640,12
5125 PLOT 1,RY+2,S$
5130 PLOT 5,RY+2,"REACTOR PRIMED      "
5135 WAIT 30
5140 PLOT 5,RY+4,"YOU ARE GO FOR LIFT-OF
F"
5150 GOTO 2000
5160 REM
5170 REM NO FUEL - STRANDED
5180 REM
5200 POKE 48640,12
5205 PLOT 1,RY+2,S$
5210 PLOT 5,RY+2,"FUEL TANKS EMPTY     "
5220 WAIT 30
5230 PLOT 5,RY+4,"MISSION ABORT - YOU'RE
 STRANDED"
5240 PLOT 5,RY+6,"PRESS ANY KEY TO TRY A
GAIN"
5250 GET I$:GOTO 1000
5996 REM
5997 REM
5998 REM      IN ORBIT
5999 REM
6000 POKE 48080,12
6010 PLOT 1,1,"ORBITAL INSERTION ACHIEVE
D      "
6020 IF AP=0 THEN GOTO 6070:REM RESCUE?
6025 REM
6030 PLOT 0,3,"CONGRATULATIONS!!"
6035 IF BT=0 OR TI<BT THEN BT=TI : PLOT
0,5,"YOUR BEST TIME SO FAR"
6040 PLOT 0,7,"IF YOU WISH TO CONTINUE Y
OUR CAREER"
6050 PLOT 0,9,"PRESS ANY KEY"
6060 GET I$ : GOTO 1000
6063 REM
6065 REM  ASTRONAUT NOT RESCUED
6067 REM
6070 PLOT 0,3,"BUT NO RESCUE?"
6080 PLOT 0,5,"I'M AFRAID YOU'LL HAVE TO
 GO BACK"
6085 WAIT 180
6090 BI=0 : PRINT DOWN$;:LL=LL+27:GOTO 2
000
29997 REM
29998 REM  **** PROGRAM INITIALISE ***
29999 REM
30000 SA=46080 :REM *** 48K ORIC ***
30010 SA=SA+960
30013 REM
30015 REM   CHANGE CHARACTER SET
30017 REM
30020 FOR N=SA TO SA+47
30030 READ M
30040 POKE N,M
30050 NEXT N
30065 A1$=CHR$(120):A2$=CHR$(121)
30070 C$=CHR$(122):B$=CHR$(123)
30075 U$=CHR$(124):P$=CHR$(125)
30080 PRINT CHR$(17);CHR$(29)
30090 DIM RL$(150)
30100 Z$="":FOR N=1 TO 40:Z$=Z$+CHR$(126
):NEXT
30103 REM
30105 REM   CONSTANT SCENE STRINGS
30107 REM
30110 FOR N=0 TO 26 : RL$(N)="

   ":NEXT
30120 FOR N=121TO125 : RL$(N)="

    ":NEXT
30130 FOR N=126TO 150: RL$(N)=Z$ :NEXT
30133 REM
30135 REM  CURSOR CONTROL STRINGS
30137 REM
30140 UP$="":DOWN$="":FOR N=1 TO 26:UP$=
UP$+CHR$(11):DOWN$=DOWN$+CH
R$(10):NEXT
30150 U$=CHR$(11)+CHR$(11)
30160 S$="

30170 DOKE #FB,DEEK(#276):DOKE #FD,DEEK(
#276)
30900 RETURN
39996 REM
39997 REM
39998 REM    DATA STATEMENTS
39999 REM
40000 DATA 24,25,18,28,24,20,50,33
40010 DATA 24,24,16,31,24,20,20,20
40020 DATA 12,63,12,30,12,63,12,30
40030 DATA 42,42,42,63,63,21,21,21
40040 DATA 63,30,12,12,12,12,30,63
40050 DATA 12,12,12,30,63,63,51,51
```


Ask a silly question...

Well, pretty silly anyway. Here's a couple of short programs sent in by 13 year old Pete Zbkowitz of Watford. The first program tends toward bad grammar (depending on what you feed it), the second is certainly an exercise in bad manners: you've heard of user-friendliness—here's some user-hostility!

SILLY RIDDLES: This generates random riddles in a simplified format somewhat along the lines of

Q: What's yellow and black and very dangerous?
A: Shark-infested custard!

The program rearranges all the riddles in its memory to give new combinations. It's a very simple program but can be made as sophisticated as you wish. Should you guess the riddle, the program will ask you to teach it a new one. Try it and see.

It's best to stick to a standard formula: say, noun clause followed by adjectival clause (or vice versa) and all in the singular.

```
10 REM   SILLY JOKES
15 REM  PETER LOBKOWITZ
20 REM  JUNE 1983
25 REM
30 REM      RANDOMISE
35 REM
40 R=DEEK(#276):DOKE #FB,R
50 R=DEEK(#276):DOKE #FD,R
60 REM
70 REM
100 REM PROGRAM START
110 DIM A$(100),B$(100),C$(100),D$(100)
120 N1=0
140 READ A$(N1)
150 IF A$(N1)="ENDS" THEN GOTO 200
160 READ C$(N1)
170 N1=N1+1 : GOTO 140
200 N2=0
210 READ B$(N2)
220 IF B$(N2)="ENDS" THEN GOTO 250
230 READ D$(N2)
240 N2=N2+1 : GOTO 210
250 REM
260 REM  MAIN LOOP
270 REM
300 CLS:PRINT:PRINT "SO YOU WANT TO TRY
A RIDDLE EH?":PRINT
310 REM INVENT RIDDLE
320 X=INT(RND(1)*N1) : Y=INT(RND(1)*N2)
330 PRINT "O.K. - WHAT ";A$(X);" AND ";B
$(Y);"?":PRINT
340 INPUT "1ST PART: ";I$:A1$=I$
350 PRINT
360 INPUT "2ND PART: ";I$:A2$=I$
370 IF A1$=C$(X) AND A2$=D$(Y) THEN GOTO
 500
380 REM **** THE WRONG ANSWER ****
390 PRINT :PRINT:PRINT "NO, THE ANSWER I
S NOT ";A1$;" ";A2$:PRINT
400 PRINT "THE ANSWER IS ";C$(X);" ";D$(
Y)
410 PRINT : PRINT "WANT ANOTHER(Y/N)? ";
: GET I$
420 IF I$="Y" THEN GOTO 300
450 CLS:STOP
480 REM **** THE RIGHT ANSWER ****
500 PRINT "YES, YOU'RE RIGHT!!!"
510 PRINT : PRINT "IT'S ";A1$;" "A2$
520 PRINT : PRINT "NOW YOU CAN TEACH ME
ANOTHER RIDDLE"
530 PRINT "FILL IN THE GAPS...."
540 PRINT :INPUT "WHAT....";I$
550 A$(N1)=I$
560 PRINT : INPUT "AND...";I$
570 B$(N2)=I$
580 PRINT "ANSWER...."
590 PRINT : INPUT "1ST PART...";I$
600 C$(N1)=I$
610 PRINT : INPUT "2ND PART...";I$
620 D$(N2)=~$
630 N1=N1+1:N2=N2+1
640 PRINT "THANK YOU!!!"
650 GOTO 410
900 REM
999 REM **** 1ST PART DATA  ****
1000 DATA "IS GREEN","FROG"
1010 DATA "IS BLACK","SHARK-INFESTED"
1020 DATA "IS RED","CARROT-FLAVOURED"
1030 DATA "GOES BANG","EXPLOSIVE"
1040 DATA "KILLS","POISONOUS"
1300 DATA "ENDS"
1336 REM
1393 REM  **** 2ND PART DATA ****
2000 DATA "IS YELLOW","CUSTARD"
2010 DATA "FALLS OUT OF THE SKY","RAIN"
2020 DATA "LIVES AT THE BOTTOM OF THE SE
A","HALLOCK"
2030 DATA "RIDE HORSES","JOCKEYS"
2040 DATA "SWINGS THROUGH THE JUNGLE","T
ARZAN"
2050 DATA "TAKE OFF FROM CAPE CANAVERAL"
,"ASTRONAUTS"
2900 DATA "ENDS"
```

RIDDLES
DOCUMENTATION

Lines 40-50:	Randomise
100-250:	Initialise
300	Main loop
380-400:	Process wrong answer
410-450:	Check if another riddle wanted
500-650:	Process right answer, get new riddle
1000-1900:	First part of riddles and answers
2000-2900:	Second part of riddles and answers

...Get a silly answer

INSULTS: This simply booby-traps your Oric so that should anyone come up and press a key, it tells them in no uncertain terms what it thinks of them, using bad-tempered retorts, apalling adjectives and nasty nouns (nothing, however, unsuitable for a family magazine). The vocabulary of the machine may be increased indefinitely by adding DATA statements.

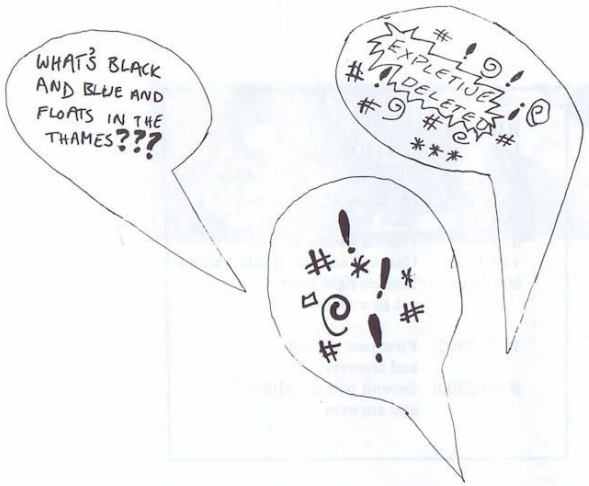
The main sophistication in this very simple program is the use of variable SN to show the number of the last adjective printed. Since up to three adjectives may be printed to each noun, this avoids printing the same adjective twice running. With a little more trouble, all three adjectives could be unique.

INSULT
DOCUMENTATION

Lines 50-60	Randomise
100-260	Program initialise
300-390:	Main loop
1000-1900:	Data: Retorts
2000-2900:	Data: Adjectives
3000-3900:	Data: Nouns

Arrays:
R$: Retorts (bad-tempered)
A$: Unpleasant adjectives

... CONTINUED OVER ...



CONTINUED FROM PREVIOUS PAGE

N$: Unpleasant nouns & noun clauses

N1: Number of retorts in array
N2: Number of adjectives in array
N3: Number of nouns in array
X: Loop counter
N,R: General purpose variables
I$: Input string
SN: Stored adjective number (line 300)
NA: Number of adjectives before noun

Insult a User

```
10 REM   BAD-TEMPERED ORIC
20 REM PETER LOBKOWITZ
30 REM JUNE 1983
35 REM
40 REM   RANDOMISE
45 REM
50 R=DEEK(#276):DOKE #FB,R
60 R=DEEK(#276):DOKE #FD,R
70 REM
90 REM    PROGRAM START
95 REM
100 DIM R$(100),A$(150),N$(100)
110 N1=0:N2=0:N3=0
120 READ R$(N1)
130 IF R$(N1)="ENDS" THEN GOTO 150
140 N1=N1+1 : GOTO 120
150 READ A$(N2)
160 IF A$(N2)="ENDS" THEN GOTO 200
170 N2=N2+1:GOTO 150
200 READ N$(N3)
210 IF N$(N3)="ENDS" THEN GOTO 250
220 N3=N3+1 : GOTO 200
250 CLS : PRINT :PRINT "YOU REALLY ASKED
 FOR IT RUNNING THIS PROGRA
M!!!!!"
255 PRINT "WHATEVER YOU DO, JUST DON'T T
OUCH MY KEYS - O.K.?"
260 PRINT
265 REM
270 REM LOOP
280 REM
290 WAIT 600
300 CLS:PRINT : PRINT :GET I$
305 N=INT(RND(1)*N1):PRINT:PRINT R$(N);"
 - YOU ";:SN=N2
310 NA=INT(RND(1)*3)+1'NO OF ADJS
320 FOR X=1 TO NA
330 N=INT(RND(1)*N2):IF N=SN THEN GOTO 3
340 SN=N:PRINT A$(N);" ";
350 NEXT X
360 REM NOUN
370 N=INT(RND(1)*N3):PRINT N$(N);"!!!"
390 PRINT : GOTO 290
800 REM
900 REM DATA STATEMENTS
950 REM
1000 DATA "GO PLAY WITH THE TRAFFIC"
1010 DATA "GO BOIL YOUR HEAD"
1020 DATA "YOU FILL ME WITH LOATHING"
1030 DATA "DROP DEAD"
1040 DATA "I HOPE YOU DEVELOP A LARGE BU
IL ON YOUR NOSE"
1050 DATA "MY PET WARTHOG IS BETTER LOOK
ING THAN YOU"
1900 DATA "ENDS"
1998 REM
1999 REM
2000 DATA "SMELLY","PUTRESCENT","HIDEOUS
","NAUSEATING","DISGUSTING"
2010 "PATHETIC","USELESS","REVOLTING","S
UB-STANDARD","MORONIC","IDI
OTIC"
2020 DATA "MONSTROUS","ABOMINABLE","FOUL
","FEEBLE-MINDED"
2900 DATA "ENDS"
2998 REM
2999 REM
3000 DATA "LUMP OF LARD","BLOB","IMBECIL
E","HYENA","RAT","TOAD","CR
ETIN"
3010 DATA "WASTE OF SPACE","JERK","DOPE"
,"ANIMAL"
3900 DATA "ENDS"
```

More from Kevin Smith of Hackney. Here's a small routine which displays the sort of rotating text you often see in shop windows. It sets double height and manipulates the text string by taking a character off the front and putting it back on the end using the LEFT$ and MID$ functions.

```
10 REM ROLLING TEXT
20 REM  KEVIN SMITH
30 REM
100 CLS:PRINT:PRINT:INPUT "ENTER YOUR ME
SSAGE";M$
110 CLS:PRINT CHR$(4)
120 FOR X=0 TO 7
130 PRINT
140 NEXT X
145 PRINT CHR$(27);"J"
150 IF LEN(M$)<38 THEN PLOT 1,9,M$:PLOT
1,10,M$:GOTO 170
160 REM   ROLLING LOOP
170 M$=MID$(M$,2)+LEFT$(M$,1)
180 IF LEN(M$)<38 THEN PLOT 1,9,M$:PLOT
1,10,M$:GOTO 210
190 PLOT 1,9,LEFT$(M$,38)
200 PLOT 1,10,LEFT$(M$,38)
210 WAIT 20:GOTO 170
```

Send in your programs ...

DATA CAPTURE!

Our resident systems analyst, Fin Fahey, offers some hints and routines on how to give your software that professional look.

NOT EVERYONE WILL be familiar with the term Data Capture—it really just means getting the right information into the machine. Most people, when they start to program in BASIC, do this simply by using the INPUT command and allowing (horror of horrors!) the screen to scroll from one prompt to the next. They also tend to neglect the need to vet the data before it is used. I've seen quite expensive software allow you to enter alphabetic characters into a numeric field—pounds and pence! Here are some routines to abolish scrolling on your Oric for good.

How it works

The first thing these routines do is allow you to format a screen rather like a paper form. A field prompt known as a descriptor can be placed anywhere on the screen in any colour you wish and the program will ask you to input your data directly after it. Secondly, since we use the GET command, the information can be vetted character by character thus if we have specified that we want a numeric field, all alphabetic characters will be locked out from that entry. Once the field has been fully entered, it may be validated for size. Thirdly, by the use of control characters we can move freely around the screen from one field to the next, and even escape from the screen entirely.

STANDARDS In order to make these routines fairly independent of your main program (you don't have to understand them to use them, but all the better if you do!) most subroutine variables start with X, Y or Z.

Those starting with X are input to the subroutines. These are known as parameters. Variables output by the routines start with Y and those for use simply within the routines with Z.

CURSOR CONTROL The Oric does not provide many ready-made cursor handling routines in TEXT mode. There isn't even a HOME function, so we've had to invent one. We keep track of the current cursor position all the time: it's stored in variables XZ and XP. The cursor is moved by printing sections of strings of cursor control characters (CHR$ 8 thru 11).

Subroutine 1000 provides the missing HOME function. It moves the cursor from the current position XZ,XP to 0,0. This is simply done by printing XZ left cursor characters and XP up cursor characters.

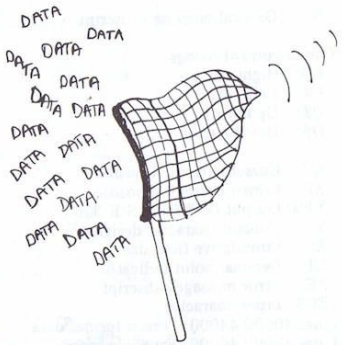
The other subroutine involves the screen format paramaters contained in DATA statements and transferred to arrays. These are as follows for each field:

Param No	Data Name	
1	XD$	Field Descriptor String
2	XX	Field Descriptor String X co-ord
3	XY	Field Descriptor Y co-ord
4	X0	Field Maximum Length
5	X1	Field Minimum Length
6	X2	Numeric Field Max Size
7	X3	Numeric Field Min Size
8	X4	Data Type. Values are: 1, String; 2, Numeric Real; 3, Numeric Real; 4, Y or N field.

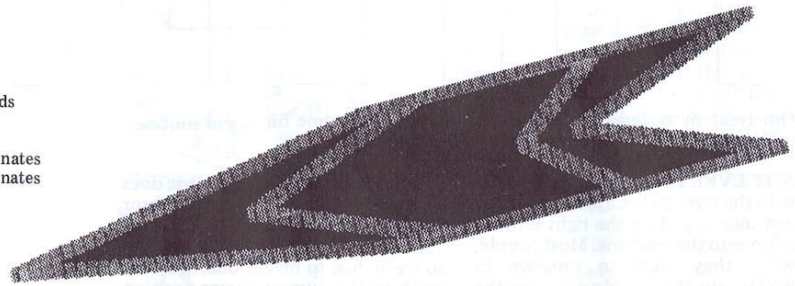
Other data types, of course, could be easily included—such as date (DD/MM/YY) for example.

The screen field for both subroutines 2000 and 3000 is designated by input parameter X. Thus S/R 2000 prints one descriptor XD$(X) at co-ordinate XX(X), XY(X). The program calls this from a loop so that all descriptors are shown at once. Note that the descriptor is preceded by an ESC (CHR$27)) character. Hence the first character of the descriptor determines its colour, flashing, etc. *See Appendix C of the Manual.*

Subroutine 3000 moves the cursor to just after the descriptor XD$(X) and accepts and validates the field. Once again the demonstration program calls this within a loop so that all fields are entered. The routine traps certain control characters when input and sets a variable Y to an appropriate value. This is then trapped by the main program so that, for example, the cursor up arrow flips the cursor to the preceding field. Finally, the subroutine outputs the captured data to field YF$. The main program places this as an element of array I$.



And when we come out of our loop, there we are—all our screen fields validated and placed neatly in array I$. Of course, much more could be done—multiple screens can be produced, other types of validation incorporated... but the central idea is the same. Perhaps in a future article we'll expand on the topic but in the meantime try it out—it'll make you feel like a professional!



NF: Number of screen fields
Arrays:
XD$: Descriptors
XX: Descriptors' X co-ordinates
XY: Descriptors' Y co-ordinates
X0: Field max length
X1: Field min length
X2: Number max size
X3: Number min size
X4: Type of field
ZM$: Error message
X: Field designator
I$: Storage of input fields for later use

N: General purpose subscript

Cursor control strings:
R$: Right cursor
L$: Left cursor
U$: Up cursor
D$: Down cursor

XZ: Cursor current X position
XP: Cursor current Y position
YF$: Output field from S/R 3000
Y: Control character designator
ZS: Cumulative field size
ZT: Decimal point indicator
ZE: Error message subscript
ZC$: Input character
Lines 40000-44000: Screen format data
Lines 45000-46000: Error messages
8000- : Initialisation for subroutines
100-300: Demonstration program
S/R 1000: Home cursor
S/R 2000: Display descriptor designated by X
S/R 3000: Input & validate field designated by X
3000-3040: Locate & clear variables
3050-3060: Generate one character—gross check
3080-3140: Handle control characters
3150-3180: Validate Y/N characters
3200-3260: Validate numeric characters
3300: Print 1 character
3310-3380: Field validation Ro times
3390-3919: Print error message if any, and exit.

```
10 REM SCREEN FORMAT S/RS
20 REM AUTHOR FIN FAHEY
30 REM COPYRIGHT CCC. LTD 1983
40 REM
50 REM
100 GOSUB 4000'INITIALISE
103 REM
105 REM PRINT DOUBLE SIZE HEADER
110 DIM I$(10)
120 PRINT CHR$(12);CHR$(4)
125 PRINTCHR$(27);"J      Personnel Recor
d";CHR$(4);:XZ=22
130 REM DISPLAY ALL FIELD
140 REM DESCRIPTORS AT ONCE
150 FOR X=1 TO NF
160 GOSUB 2000
170 NEXT X
173 REM
175 REM LOOP TO INPUT ALL FIELDS
177 Y=0
180 FOR X=1 TO NF
190 GOSUB 3000' INPUT ROUTINE
200 I$(X)=YF$
210 IF Y=98 AND X>1 THEN X=X-1:GOTO 190
215 IF Y=98 THEN GOTO 190
220 IF Y=99 AND X<NF-1THEN X=X+1:GOTO 19
0
230 IF Y=97 THEN X=NF
235 IF Y=96 THEN X=NF
240 NEXT X
243 REM
245 REM END OF LOOP
247 REM
250 IF Y=97 GOTO 177
260 IF Y=96 THEN CLS:PRINT:PRINT:PRINT "
YOU HAVE JUST ESCAPED"
265 PRINT CHR$(17)
270 PRINT:PRINT "END OF DEMONSTRATION"
300 GOTO 300
995 REM
996 REM
997 REM
998 REM  HOME CURSOR
999 REM
1000 REM
1010 PRINT LEFT$(L$,XZ);
1020 PRINT LEFT$(U$,XP);
1030 XP=0 : XZ=0
1040 RETURN
1995 REM
1996 REM
1997 REM
1998 REM DISPLAY A DESCRIPTOR
1999 REM
2000 GOSUB 1000'HOME
2010 PRINT LEFT$(D$,XY(X));
2020 PRINT LEFT$(R$BXX(X));
2030 PRINT CHR$(27);XD$(X);
2040 XP=XY(X) : XZ=XX(X)+LEN(XD$(X))
2050 RETURN
2995 REM
2996 REM
2997 REM
2998 REM  GET & VALIDATE DATA
2999 REM
3000 GOSUB 1000'HOME
3005 REM MOVE TO FIELD LOCATION
3010 PRINT LEFT$(D$,XY(X));
3020 PRINT LEFT$(R$,XX(X)+LEN(XD$(X)));
3030 XP=XY(X):XZ=XX(X)+LEN(XD$(X))
3035 REM INITIALISE ALL VARIABLES
3040 ZS=0 : ZT=0 : YF$="" : ZE=0 :Y=0
3045 PRINT CHR$(17);:REM CURSOR ON
3047 REM GET ONE CHAR OF FIELD
3050 GET ZC$
3055 IF ASC(ZC$)>125 THEN GOTO 3050
3060 IF ASC(ZC$)>31 THEN GOTO 3150
3065 REM
3070 REM TRAP CONTROL CHARS
3075 REM
3080 IF ZC$<>CHR$(8) THEN GOTO 3090' LEF
T CURSOR (=DELETE)
3085 IF ZS>0 THEN ZS=ZS-1:PRINT ZC$;" ";
ZC$;:YF$=LEFT$(YF$,ZS):XZ=X
Z-1
3090 IF ZC$=CHR$(10) THEN Y=99 : GOTO 38
00' DOWN CURSOR
3100 IF ZC$=CHR$(11) THEN Y=98 : GOTO 38
00' UP CURSOR
3110 IF ZC$=CHR$(12) THEN Y=97 : GOTO 38
00' HOME TO TOP FIELD
3120 IF ZC$=CHR$(13) AND ZS>X1(X) THEN G
OTO 3320' RETURN
3130 IF ZC$=CHR$(27) THEN Y=96:GOTO 3800
' ESCAPE
3140 GOTO 3050
3142 REM
3145 REM PRINTABLE CHARACTERS
3147 REM
3150 IF X4(X)<4 THEN GOTO 3200
3160 REM VALIDATE Y/N FIELD
3170 IF ZC$<>"N" AND ZC$<>"Y" THENGOTO 3
050
3180 PRINT ZC$;:YF$=ZC$:ZS=1:XZ=XZ+1:GOT
O 3900
3190 REM STRING FIELD?
3200 IF X4(X)=1 THEN GOTO 3300
3205 REM   NUMERIC FIELD
3210 IF X4(X)=2 AND ZC$="." THEN GOTO 30
50'WHOLE NUMBER
3220 IF ZS<>0 AND ZC$="-" THEN GOTO 3050
' LEADING -VE SIGN ONLY
3230 IF ZC$="." AND ZT=1 THEN GOTO 3050'
 ALREADY DECIMAL POINT
3240 IF ZC$="." THEN ZT=1 : GOTO 3300'1S
T DECIMAL POINT
3245 REM CHECK WITHIN NUMERIC CHAR SET
3250 IF ZC$="-" OR (ASC(ZC$)=>48 AND ASC
(ZC$)<=57) THEN GOTO 3300
3260 GOTO 3050
```



```
2995 REM
2996 REM
2997 REM
2998 REM  GET & VALIDATE DATA
2999 REM
3000 GOSUB 1000'HOME
3005 REM MOVE TO FIELD LOCATION
3010 PRINT LEFT$(D$,XY(X));
3020 PRINT LEFT$(R$,XX(X)+LEN(XD$(X)));
3030 XP=XY(X):XZ=XX(X)+LEN(XD$(X))
3035 REM INITIALISE ALL VARIABLES
3040 ZS=0 : ZT=0 : YF$="" : ZE=0 :Y=0
3045 PRINT CHR$(17);:REM CURSOR ON
3047 REM GET ONE CHAR OF FIELD
3050 GET ZC$
3055 IF ASC(ZC$)>125 THEN GOTO 3050
3060 IF ASC(ZC$)>31 THEN GOTO 3150
3065 REM
3070 REM TRAP CONTROL CHARS
3075 REM
3080 IF ZC$<>CHR$(8) THEN GOTO 3090' LEF
T CURSOR (=DELETE)
3085 IF ZS>0 THEN ZS=ZS-1:PRINT ZC$;" ";
ZC$;:YF$=LEFT$(YF$,ZS):XZ=X
Z-1
3090 IF ZC$=CHR$(10) THEN Y=99 : GOTO 38
00' DOWN CURSOR
3100 IF ZC$=CHR$(11) THEN Y=98 : GOTO 38
00' UP CURSOR
3110 IF ZC$=CHR$(12) THEN Y=97 : GOTO 38
00' HOME TO TOP FIELD
3120 IF ZC$=CHR$(13) AND ZS>X1(X) THEN G
OTO 3320' RETURN
3130 IF ZC$=CHR$(27) THEN Y=96:GOTO 3800
' ESCAPE
3140 GOTO 3050
3142 REM
3145 REM PRINTABLE CHARACTERS
3147 REM
3150 IF X4(X)<4 THEN GOTO 3200
3160 REM VALIDATE Y/N FIELD
3170 IF ZC$<>"N" AND ZC$<>"Y" THENGOTO 3
050
3180 PRINT ZC$;:YF$=ZC$:ZS=1:XZ=XZ+1:GOT
O 3900
3190 REM STRING FIELD?
3200 IF X4(X)=1 THEN GOTO 3300
3205 REM   NUMERIC FIELD
3210 IF X4(X)=2 AND ZC$="." THEN GOTO 30
50'WHOLE NUMBER
3220 IF ZS<>0 AND ZC$="-" THEN GOTO 3050
' LEADING -VE SIGN ONLY
3230 IF ZC$="." AND ZT=1 THEN GOTO 3050'
 ALREADY DECIMAL POINT
3240 IF ZC$="." THEN ZT=1 : GOTO 3300'1S
T DECIMAL POINT
3245 REM CHECK WITHIN NUMERIC CHAR SET
3250 IF ZC$="-" OR (ASC(ZC$)=>48 AND ASC
(ZC$)<=57) THEN GOTO 3300
3260 GOTO 3050
3300 PRINT ZC$;:YF$=YF$+ZC$:ZS=ZS+1:XZ=X
Z+1
3305 REM   CHECK FOR MAX LENGTH
3310 IF ZS<X0(X) THEN GOTO 3050
3315 REM   CHECK FOR MAX SIZE(NUM)
3320 IF X4(X)>1 AND VAL(YF$)>X2(X) THEN
ZE=1
3325 REM   CHECK FOR MIN SIZE(NUM)
3330 IF X4(X)>1 AND VAL(YF$)<X3(X) THEN
ZE=2
3340 REM
3350 REM   OTHER VALIDATION
~3360 REM       ROUTINES
3370 REM   CAN BE INSERTED HERE
3380 REM
3390 PRINT CHR$(17);:REM CURSOR OFF
3400 IF ZE=0 THEN GOTO 3420'NO ERRORS
3410 PRINT LEFT$(L$,ZS);SPC(ZS);LEFT$(L$
,ZS);:XZ=XZ-ZS
3415 REM PRINT ERROR MESSAGE
3420 GOSUB 1000'HOME
3430 PRINT LEFT$(D$,25);:PRINT CHR$(27);
"L";ZM$(ZE);:XP=25:XZ=LEN(Z
M$(ZE))+1
3440 GOSUB 1000'HOME
3450 IF ZE>0 THEN PING:GOTO 3010
3460 GOTO 3910
3800 REM
3900 PRINT CHR$(17);:REM CURSOR OFF
3910 RETURN
3995 REM
3996 REM
3997 REM
3998 REM   NECESSARY INITIALISATION ROUT
INES
3999 REM
4000 NF=8
4010 DIM XD$(NF),XX(NF),XY(NF),X0(NF),X1
(NF),X2(NF),X3(NF),X4(NF)
4020 FOR N=1 TO NF
4030 READ XD$(N),XX(N),XY(N),X0(N),X1(N)
,X2(N),X3(N),X4(N)
4040 NEXT N
4050 DIM ZM$(2)
4060 FOR N=1 TO 2 :READ ZM$(N):NEXT N
4070 REM
4080 REM SET UP CURSOR CONTROL STRINGS
4090 REM
4095 R$="":L$="":U$="":D$=""
4100 FOR N=1 TO 40:R$=R$+CHR$(9):L$=L$+C
HR$(8):NEXT N
4110 FOR N=1 TO 26:U$=U$+CHR$(11):D$=D$+
CHR$(10):NEXT N
4120 ZM$(0)="
      "
4130 PRINT CHR$(17);CHR$(29);
4140 INK 7:PAPER 0
4900 RETURN

39995 REM
39996 REM
39997 REM SCREEN CONTROL PARAMETERS
39998 REM
40000 DATA "AName     ",4,3,20,5,0,0,1
40010 DATA "BAddress ",4,5,20,0,0,0,1
40020 DATA "B         ",4,7,20,0,0,0,1
40030 DATA "B         ",4,9,20,0,0,0,1
40040 DATA "CAge(Yrs) ",4,11,2,1,99,18,2
40045 DATA "DTitle ",20,11,3,1,0,0,1
40050 DATA "ESalary ",4,13,8,3,15000,200
0,3
40060 DATA "Is this data correct? ",4,1
5,1,1,0,0,4
44996 REM
44997 REM
44998 REM ERROR MESSAGES
44999 REM
45000 DATA "Number too big"
45010 DATA "Number too small"
```

Fin Fahey leads off a monthly feature for the machine-code orientated.

IN THIS COLUMN we welcome your contributions. If you have any routines or programs, long or short, that you would like to share, please feel free to send them in. We'd prefer to receive listings in proper assembler format rather than as a series of POKES from BASIC although we haven't — so far — come across an adequate ORIC assembler. Starring programs will gain you £5.

A powerful legacy.

THE 6502 has been with us a long while now — the is only the latest in a long line of illustrious micro-computers based around it. We therefore have the advantage of all the publications and listings published over the years, often for other machines, but just as applicable to the ORIC. Studying a dissassembled Applesoft BASIC Listing can be very illuminating.

If you're completely new to machine code, can I recommend that you get hold of a good 6502 primer, such as Rodney Zak's manual published by Sybex. If you're serious, it's worth buying an assembler — I would not recommend programming in hex POKES to my worst enemy.

Kick out the REMS.

THE LISTING which follows is for a very simple REM remover. The program only removes REMS when they are at the start of the line. It won't take out REMs following a colon after a BASIC statement. Apart from being useful, it does serve to illustrate the structure of a BASIC Program.

You may not be aware of it but each line of an ORIC BASIC program has a constant overhead of 6 bytes. This is high by many machine standards but, given that most of us have 48K models, the waste of space shouldn't matter too much.

Do it byte by byte

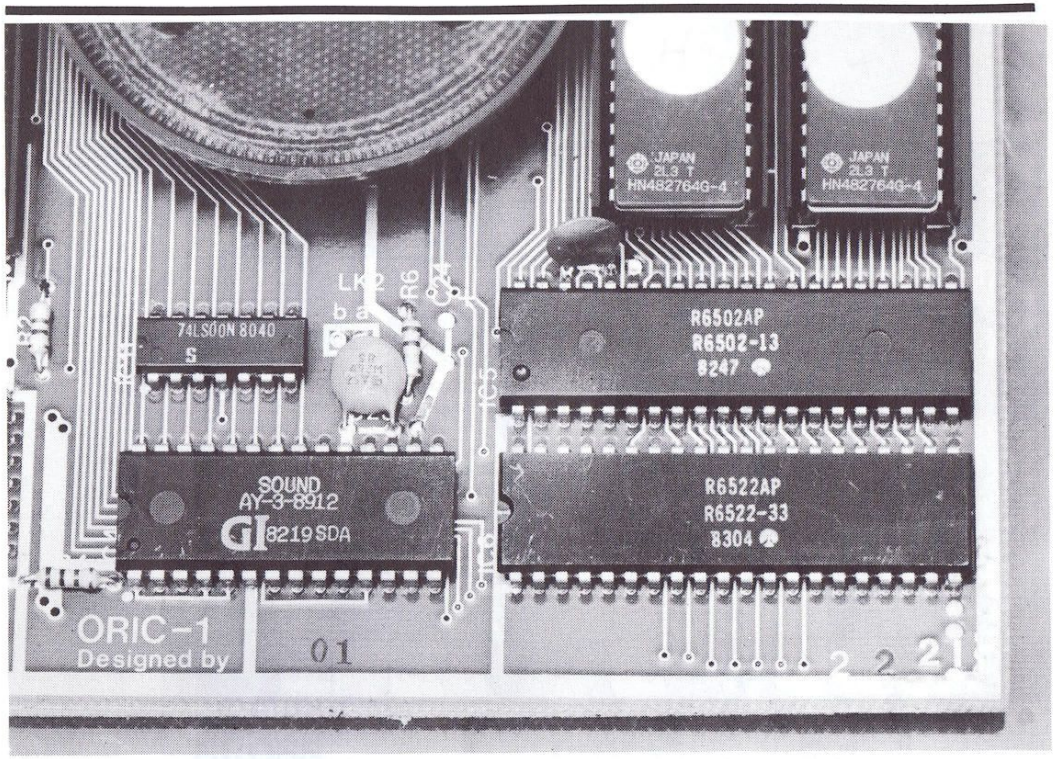
Each line starts with a two byte vector which points to the start of the next line in the program. This is followed by a two byte line number. After this comes the program line proper, which is terminated by a space and a null (zero) byte. The BASIC program itself is followed by a further two null bytes.

On page zero, the address of the start of BASIC is held in locations #9A and #9B, while the end address is held in #9C and #9D. These point to the first free location after the end of the BASIC program. The normal BASIC start address is, of course, #0501 but you may alter it to your liking.

The program which follows simply copies the BASIC program sequentially into its own space line by line. It used four free page zero locations of its own, #40 and #41 (the Source Line address,) and #42 and #43 (the Destination Line address). The Indirect Y addressing mode is used, with these addresses as base. REM Lines are removed by the simple expedient of not copying them over. At the end of the run, the End of BASIC address is updated. The assembly shown is at address 40000 but the routine is fully relocatable.

A final word of caution — Make sure that none of your GOTOs or GOSUBs reference REMs because this routine won't take the trouble to check.

See you next month.

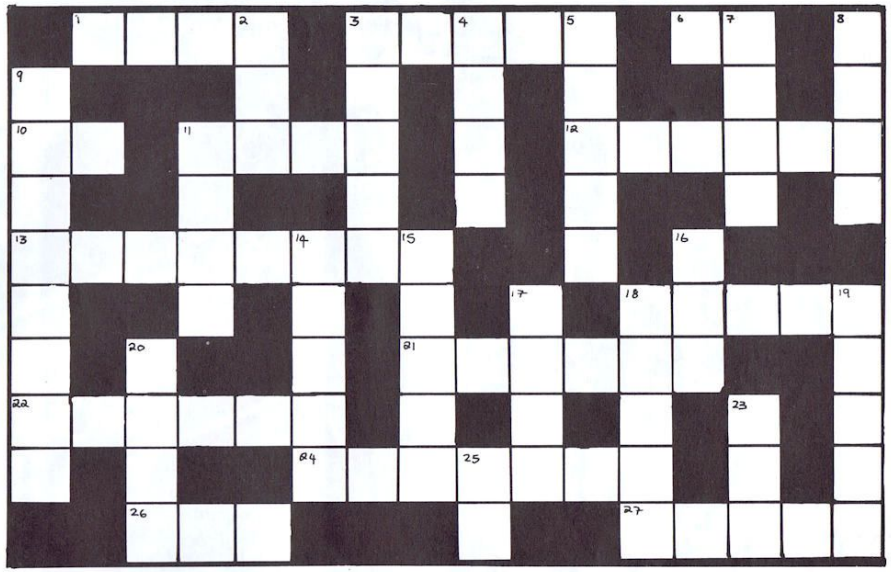



```
*********************************
*                               *
*   MACHINE CODE DE-REMMER      *
*   COPYRIGHT CCC LTD 1983      *
*   AUTHOR FIN FAHEY            *
*                               *
*********************************
***** SYMBOL TABLE *****
BAS   EQU #9A
BEND  EQU #9C
SRC   EQU #40
DEST  EQU #42
ORG 40000
***** DEREM CODE *****
```

```
9C40 A59B   @START  LDA BAS   ** INITIALISE
9C42 8541           STA SRC     SOURCE &
9C44 8543           STA DEST    DESTINATION
9C46 A59A           LDA BAS+1   ADDRESSES
9C48 8540           STA SRC+1   TO START OF
9C4A 8542           STA DEST+1  BASIC **
9C4C A001   @LOOP   LDY $1      MAIN LOOP
9C4E B140           LDA (SRC),Y
9C50 C900           CMP $0      CHECK FOR END
9C52 F049           BEQ @END
9C54 A000           LDY $0      ** COMPUTE
9C56 B140           LDA (SRC),Y    LINE
9C58 38             SEC            LENGTH
9C59 E540           SBC SRC        & STORE IN
```

```
9C5B AA              TAX            X REG **
9C5C 1001            BPL @REMCHK
9C5E CA              DEX
9C5F A004       @REMCHK LDY $4      ** CHECK FOR
9C61 B140            LDA (SRC),Y    REMARK
9C63 C99D            CMP $#9D       TOKEN **
9C65 F029            BEQ @REM
9C67 A000            LDY $0         ** COMPUTE
9C69 8A              TXA            LINE VECTOR
9C6A 18              CLC            LO BYTE  **
9C6B 6542            ADC DEST
9C6D 9142            STA (DEST),Y
9C6F A900            LDA $0         ** COMPUTE
9C71 C8              INY            LINE VECTOR
9C72 6543            ADC DEST+1  HI BYTE  **
9C74 9142            STA (DEST),Y
9C76 8A              TXA            ** SAVE X REG
9C77 48              PHA             INITIALISE
9C78 CA              DEX             COPY LOOP
9C79 A002            LDY $2          INDICES  **
9C7B B140       @COPY   LDA (SRC),Y ** COPY THE
9C7D 9142            STA (DEST),Y    WHOLE
9C7F C8              INY             BASIC
9C80 CA              DEX            LINE FROM
9C81 E001            CPX $1          SRC TO
9C83 D0F6            BNE @COPY       DEST **
9C85 68              PLA            ** RESTORE
9C86 AA              TAX             X REG **
9C87 18              CLC
9C88 6542            ADC DEST     ** UPDATE
9C8A 8542            STA DEST        DESTN
9C8C 9002            BCC @REM1       ADDRESS **
9C8E E643            INC DEST+1
9C90 18         @REM    CLC
9C91 8A              TXA            ** UPDATE
9C92 6540            ADC SRC         SOURCE
9C94 8540            STA SRC       ADDRESS **
9C96 90B4            BCC @LOOP
9C98 E641            INC SRC+1
9C9A 18              CLC            ** END OF
9C9B 90AF            BCC @LOOP       LOOP **
9C9D 88         @END    DEY
9C9E 9142            STA (DEST),Y ** PLACE
9CA0 C8              INY             NULL BYTES
9CA1 9142            STA (DEST),Y    AT END **
9CA3 A543            LDA DEST+1   ** UPDATE
9CA5 859D            STA BEND+1      END-OF-
9CA7 18              CLC             BASIC
9CA8 A542            LDA DEST        ADDRESS
9CAA 6902            ADC $2          ON PAGE
9CAC 859C            STA BEND        ZERO **
9CAE 9002            BCC @RETURN
9CB0 E69D            INC BEND+1
9CB2 60         @RETURN RTS
```


ISKRA

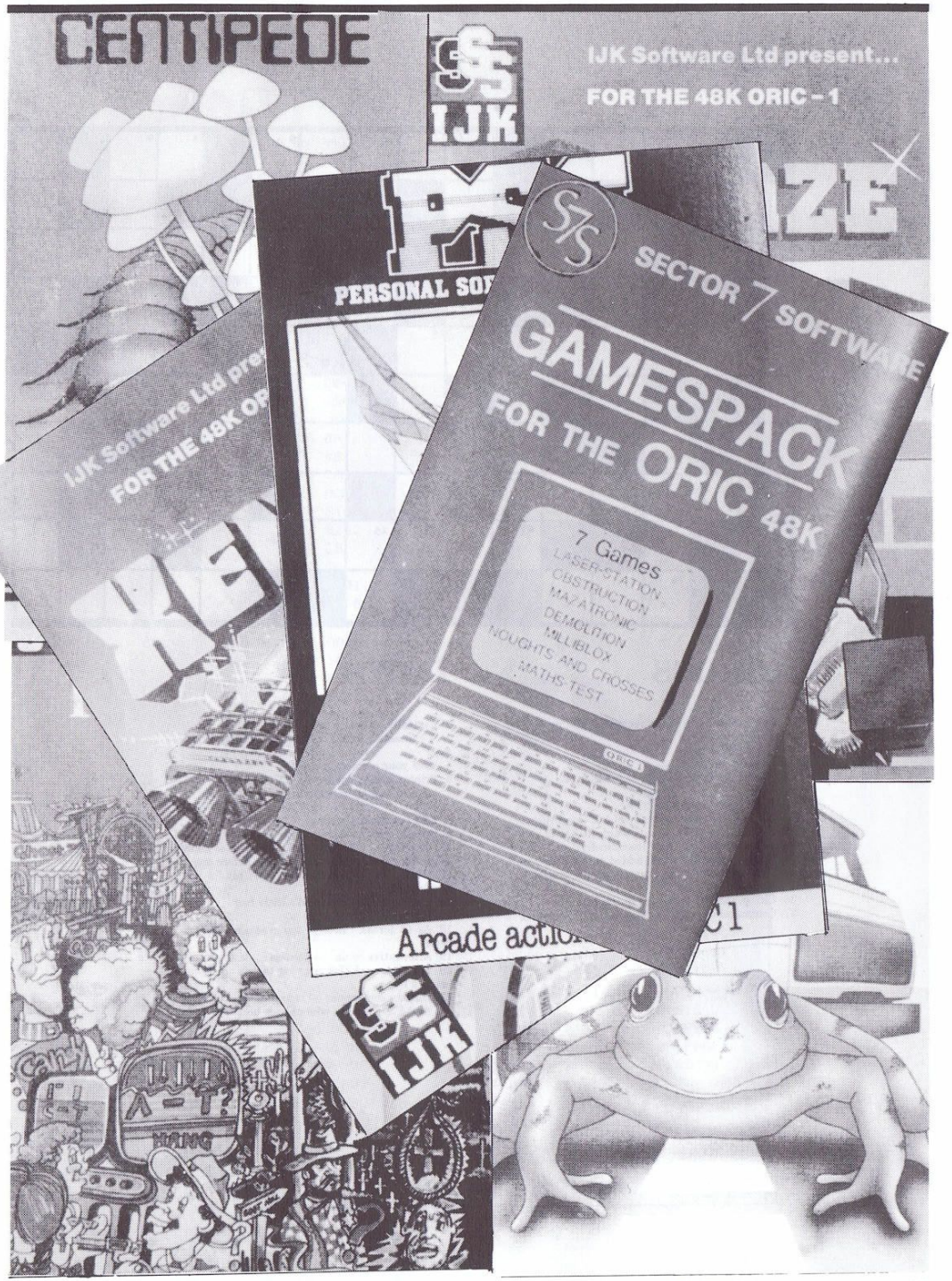


ACROSS

1. Ring enclosure to make records available
2. Fruit machine
6. — Y on 12 across
10. Alternative Function
11. Rico is confused enough to produce our machine
12. Vet . . . what we see on TV
13. Knight leads the company out of Cambridge
18. Scottish son of Ro — in a word, makes it big
20. Not a moving memory (technically)
21. Hugh soundalike — an added game attraction!
24. One could be a good example as one precedes . . . confused regent
26. Chi p handles messages to the printer
27. Command to put things in the right sequence

DOWN

2. Logically, opposite to 10 across
3. A character, sound like Arthur the funny man
4. Give your memory a prod?
5. Printer maker opens mix-up
7. Look in to your memory!
8. This commmand sounds . . . Chinese?
9. Develop game control: follow pleasure with rod
11. One leaves ORICS to make enemies of hobbits
14. In this company, a sailor comes before one
15. ORIC button you can't hit accidentally
16. Strategic Air Command's little bag
17. Big noise?
18. As 18 across, I become top grade, meaning little
19. Alternative to us — possess . . . the queen!
20. Turnover — I kick out ring to prevent it from being a flop
23. Display component — it sounds heavy!
25. And French alien who phones home?

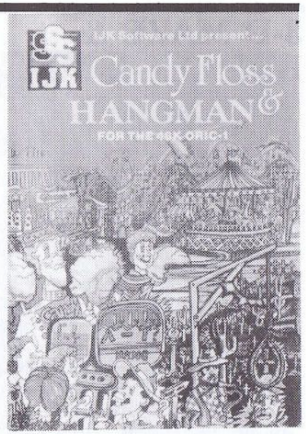


CANDYFLOSS & HANGMAN

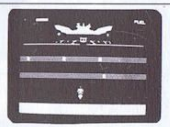
IJK Software
9 King Street
Blackpool
Lancashire

48K

ONCE AGAIN, I feel that IJK has overpriced its product. Candy Floss is a version of a game which we used to call Lemonade when run on the Apple II. It involves selling candy floss to realise a profit on Blackpool's Golden Mile. Although it does have an initial charm — particularly for kids — it does become repetitious very quickly. Hangman should be familiar to everyone from their schooldays and I think this is an enjoyable version: the hanged man changes expression depending on your guesses. It could do with a time limnit on the guesses, I think. Better value on the whole for younger children than the 3D Maze & Breakout package but still, in my opinion, not up to the asking price. [£7.50]



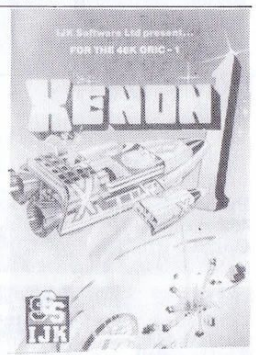
XENON-1



IJK Software
9 King Street
Blackpool
Lancashire

48K

THIS IS A GAME which has seen some service. I seem to remember playing it on an Apple II some years ago. Having said that, I must admit I rather like it even if I never get as far as the Battle Star. Basically, the game is several games in one — a couple of invadery shoot-em ups, dodge the falling meteorites and then the original bit. You have to laser down the falling Paratrons before they get beneath you. If they do, they crawl around underneath and get up to no good. Unfortunately, I got rather sick of going through Frames 1, 2 and 3 which are pretty simple to get to Frame 4 where the fun starts. On the whole, the game has a spark of crazed originality which saves it. [£8.50]



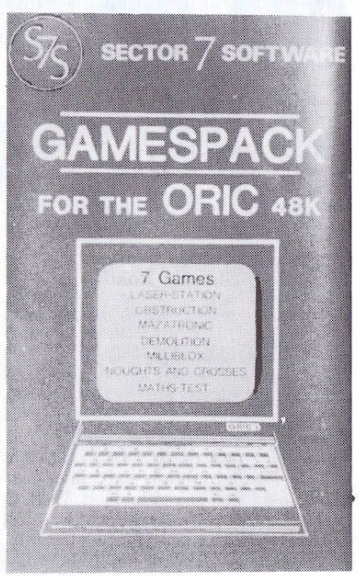
ORIC GAMES PACK

Sector 7 Software
PO Box 8
Newton Abbot
Devon
£7.00

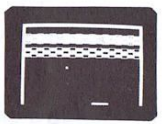
48K

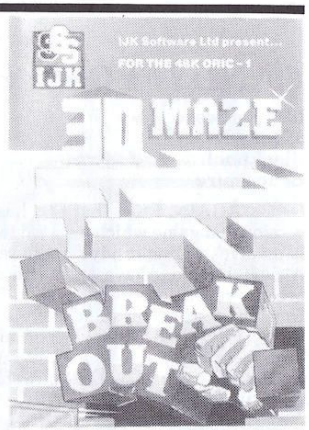
THIS IS PART of a very common genre of games packages. And to a buyer it can have its attractions as a sort of lucky dip: after all seven programs at a pound each is a very attractive proposition even if, for the most part, they are written in BASIC. The problem is that generally these packages are rapidly hacked out, boring, bug-ridden rip offs. I'm happy to say that Sector 7 have got it mostly right with this one. There are three eminently playable games which are addictive in spite of their simplicity: Obstruction, Milliblox and Demolition. The last is a version of the ever-popular Blitz and is my personal favourite. Milliblox is a very simplified Pacman derivative and Obstruction is a version of a game known variously as Worms, Light-Cycle etc. Laser Station I found pointless and limited while Mazatronic, a 3D maze program is well-written but only fun for one or two plays. Of the remaining two, Noughts and Crosses speaks for itself. The Maths Test game — an attempt at educational software — had a bug in it which stopped it running at all.

A mixed bag but on the whole good value. I'd recommend it to people who've just bought an ORIC and want to find out what can be done with BASIC.



3-D MAZE & BREAKOUT



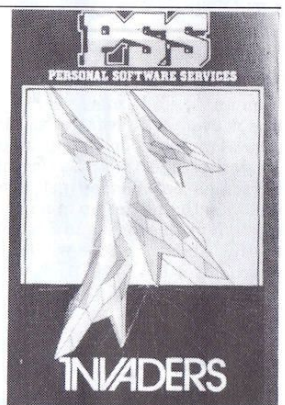


IJK Software
9 King Street
Blackpool
Lancashire

48K

NOT MUCH TO recommend this package. Breakout is really ancient history — even older than Space Invaders. If it's really what you want to play on your ORIC, this is a pretty reasonable version. As for the 3D Maze, this version is not as good as the Sector 7 offering in my opinion, which is one of seven programs in a package. Both of these programs would be quite acceptable as part of a bumper games package but IJK has grossly overpriced them here. [£7.50]

INVADERS



Arcade action for ORIC 1

Personal Software Services
452 Stoney Stanton Road
Coventry
£6.95

48K

OH GODS! The Original Video Game! What to say about this: it's like being asked to review Julius Caesar for a modern literary magazine. Well then, given that I probably haven't played this for two years, this looks like a good version. All the necessaries are there: the whirling saucer at the screen top, the sinister bass beat as the space invaders advance, the wriggly legs — but somehow, I couldn't work up that much excitement. Still, if you haven't played this game to death, you'll probably find this is worth a look. Fast and faultless but for originality, grade zero.

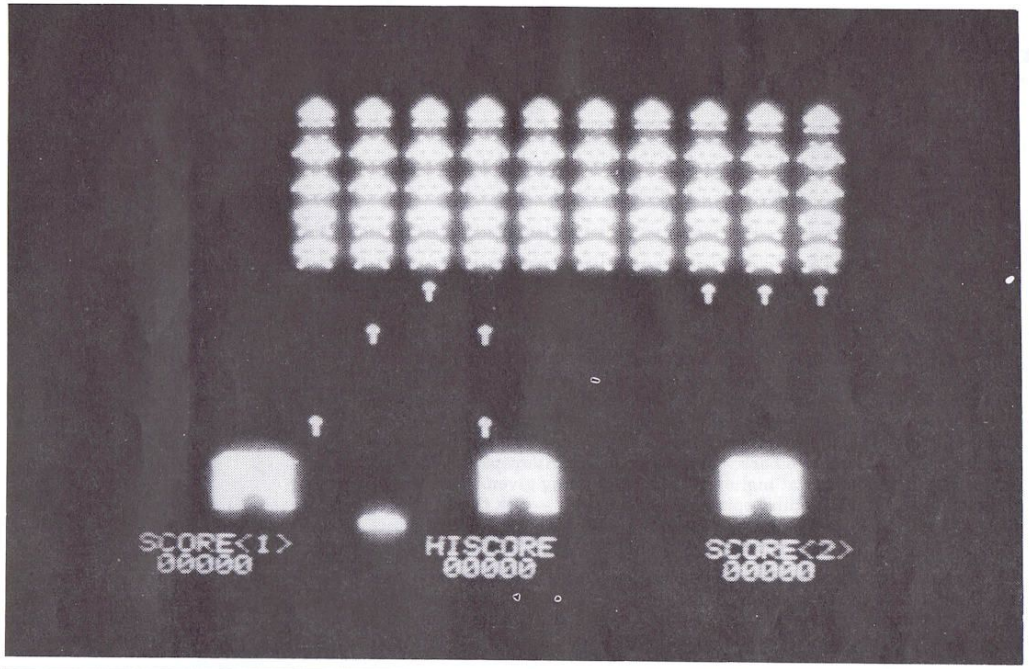
CENTIPEDE

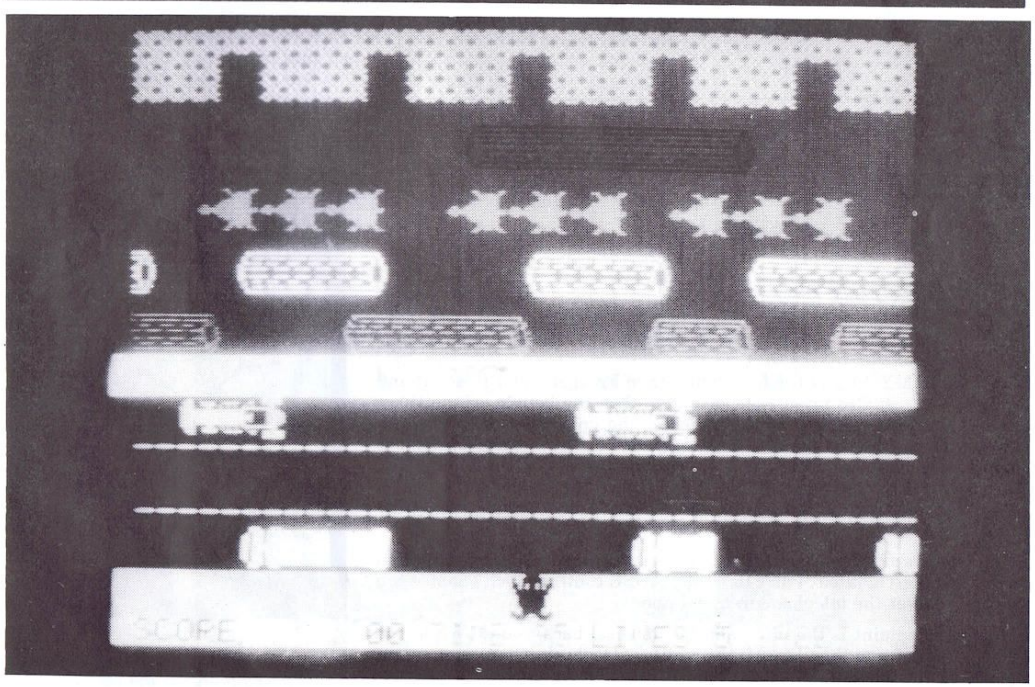
Arcade action for ORIC 1

Personal Software Services
452 Stoney Stanton Road
Coventry
£6.95

48K

ANOTHER COVER version here from PSS and up to their usual high standard. As you may know, Centipede is a popular arcade game involving zapping the eponymous creature as it races towards you through a field of mushrooms along with various other crawlies. Unfortunately, it tends to break into independent segments which are correspondingly hard to hit. It's difficult to fault this version except that, to this reviewer, it seems that Frame 1 is inordinately hard, once you're through it, life eases up and you're on Frame 5 before you know it. Well, that might just be the way I play. I felt this one could have done with the addition of a FREEZE button — if only to make screen photographs easy! highly competent, if unoriginal package on the whole, but good value.

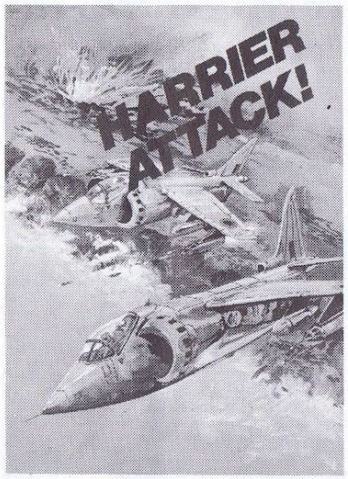




HARRIER ATTACK

Durrell Software
Castle Lodge
Castle Green
Taunton TA4 1AB

Price £6.95 **16/48K**

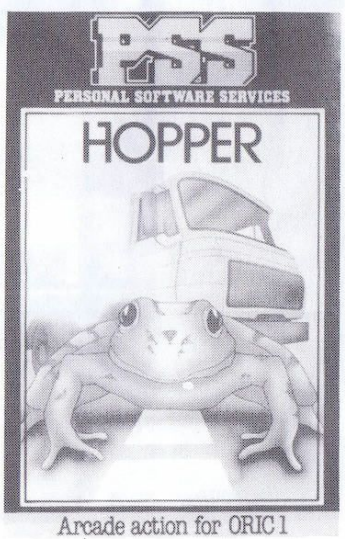


I HAVE TO say that against my better judgment I found this a compulsive and interesting game. As the title makes clear, this is a Post-Falklands war game — a genre of which I am deeply suspicious. What makes the game interesting is that unlike most real-time games, you can choose between mission options. If you get a kick out of bombing almost defenceless Argentinian conscripts on the ground, you may do that. If on the other hand, you prefer an opponent who hits back, you may go on an intercept mission and tussle with the Skyhawks at higher altitude. Not being given to dreams of military glory, I found the third option most appealing — jack the skill level up and dodge the flak and fighters to get to the other side of the island intact — it's not easy.

All in all the graphics and sound are unspectacular but adequate. The main flaw is that the rockets are almost impossible to use, which is very annoying as you're supplied with huge amounts of them. If this is your cup of tea, you'll probably find it value for money.

HOPPER

Personal Software Services
452 Station Road
Coventry
£6.95 **48K**



Arcade action for ORIC 1

THIS ONE IS MY 'Pick of the Month' but then Frogger — for this is it and no other — is usually my arcade favourite. For the benefit of the three people on the Isle of Skye who don't know the game — the object is to manoeuvre a bewildered looking frog across a motorway and a river without it being drowned, being squashed or bitten by a snake on the roadside. It makes a great change from zapping aliens and is guaranteed guilt-free. A fine version, although it lacks alligators which, in my opinion, makes the river crossing a bit too unperilous.

The game is attractively packaged and well documented and includes an arcade-style 'hall of fame'. I also like the PAUSE control which enables you to go and answer the telephone in mid-game.

My main complaint is the incessant daft musical background. While you can turn this off, you then lose all the auditory feedback if you do. A minor gripe, so on the whole good value.

SYSTEMS SOFTWARE SECTION

ORIC-MON

Personal Software Services
452 Stoney Stanton Road
Coventry
£8.95 **16/48K**

Programing aid for ORIC 1

A MACHINE CODE Monitor/Disassembler from PSS — this is all you could wish it to be. Memory can be displayed on the screen as hex or ASCII or as assembly code, block moved, altered or searched. Machine Code routines can be made relocatable, registers dumped and so on.

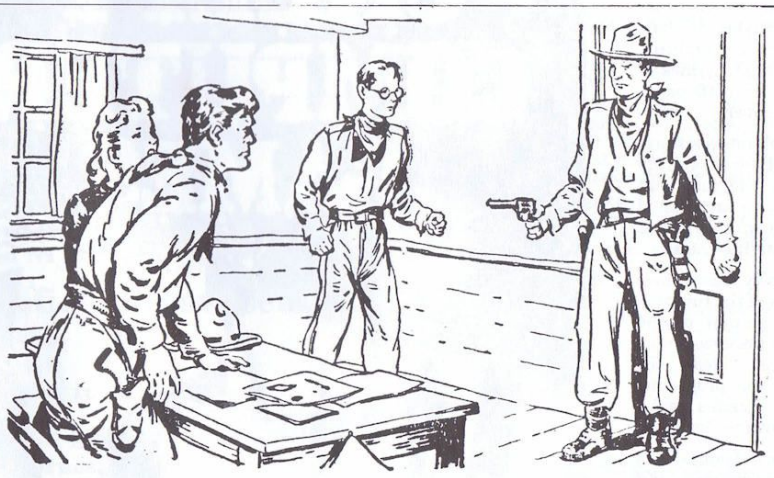
My only niggles are that, I feel, at nearly £9 it's slightly overpriced — this does partly reflect the systems software market. There are mistakes in documentation: the H and D functions seem to be reversed, for example.

Although unspectacular, this sort of program is a good, solid and reliable tool particularly for assembler programmes.

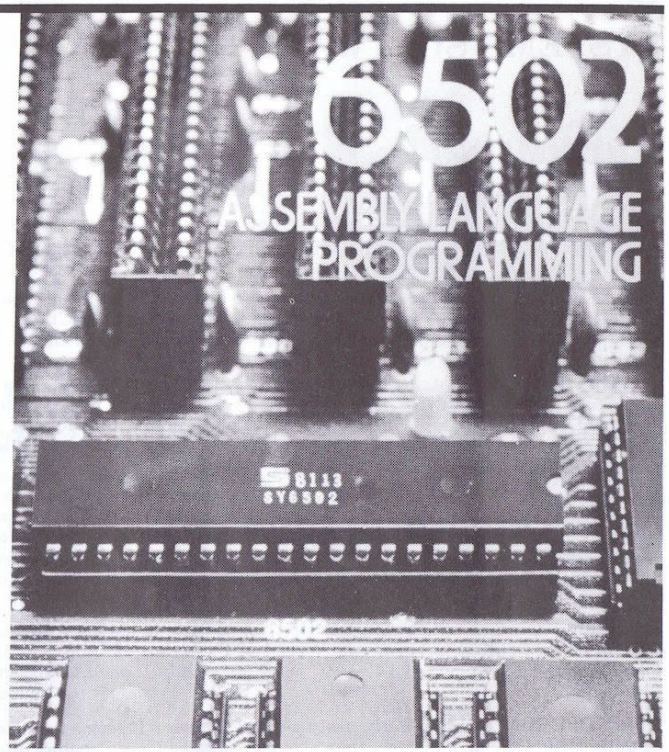
ORIC-1 ASSEMBLER/ DISASSEMBLER

Microplot
19 The Earls Croft
Cheylesmore
Coventry **16/48K**

IN MY OPINION, this is a pretty nasty piece of software. It seems to me to be unfinished: I couldn't get more than about five assembler mnemonics into a program without the whole thing bombing out due to one bug or another. Worse still — even if it did work — I feel it would be almost impossibly cumbersome to use. I don't feel that this package represents value for money.

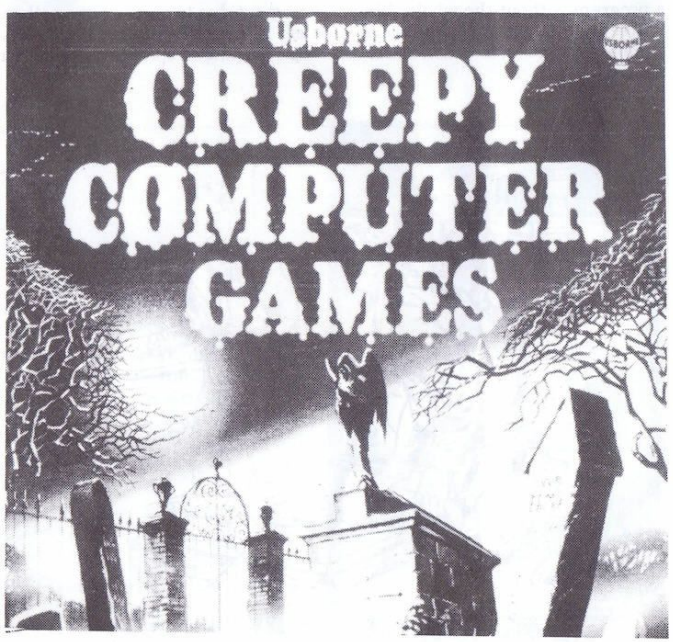


6502



6502 Assembly Language Programming
Judi N. Fernandez, Donna N. Tabler & Ruth Ashley
John Whiley & Sons Ltd
£9.95

IF YOU WANT to get started on learning assembly language, it's hard to do better than this. It assumes no initial knowledge of 6502 assembler and takes you step by step through hex arithmetic addressing and the instruction set. Best of all, it assumes that you will want to use an assembler and explains in some detail what this involves. Too many authors try to encourage people to POKE machine code in from BASIC. Rodney Zak's books on 6502 are still indispensable reference works but this book will be much more useful to the absolute beginner.
Fin Fahey



CREEPY COMPUTER Games is an interesting little book from Usborne Publishing, a name known for producing a range of highly polished books aimed at the home computing beginner. There are eight games in the book, which has only 16 pages and a rather spooky cover.

What makes the book interesting is not the programs — they are hardly sophisticated — but the fact that each program, although only listed once, can be run on any one of eight different micros, including the Oric.

Another praiseworthy feature is the clear descriptions of the function performed by each section of the program. These descriptions and a section which gives ideas for experimenting with the programs, together with some puzzles make the book a useful learning device.

Creepy Computer Games is available in most bookshops and costs 99p.

How to subscribe

£10 for a year's subscription 12 issues

£20 for overseas subscriptions

Name: .....

Address: .....

.....

ORIC USER
20 Wynford House
Wynford Road
London N1 9QY

Classified advertisements

Free to user groups
10p a word
Advertising rates on request

Distributed by
Moore Harness Ltd
1 Eagle Wharf Road
London N1

FINE SOFTWARE BY CO-OPERATIVE COMPUTERS & COMMUNICATIONS LTD

All for 48K ORIC-1

SIGNING ON

Can you survive the pressures of unemployment? It isn't easy!

UTILITY PACK NUMBER 1

Get your ORIC working with our Advanced Character Editor, Program Compressor and Graphics Designer programs. The latter will dump your 4 Colour Hi-Res screens to the ORIC printer.

GAMES PACK NUMBER 1

A Bumper Bundle: Tank Trap, Neptune Rescue, Nuclear Train, Poetry, Hangman.

Prices: ORIC USER subscribers £4.95
Non-subscribers £5.95
In the unlikely event that your cassette fails to load we will exchange it promptly

CCC WANTS YOU!!! *Got any software lying around? If we like it and we think other people might too, then there might just be a hefty 25% royalty coming your way! Try us and see. . .*