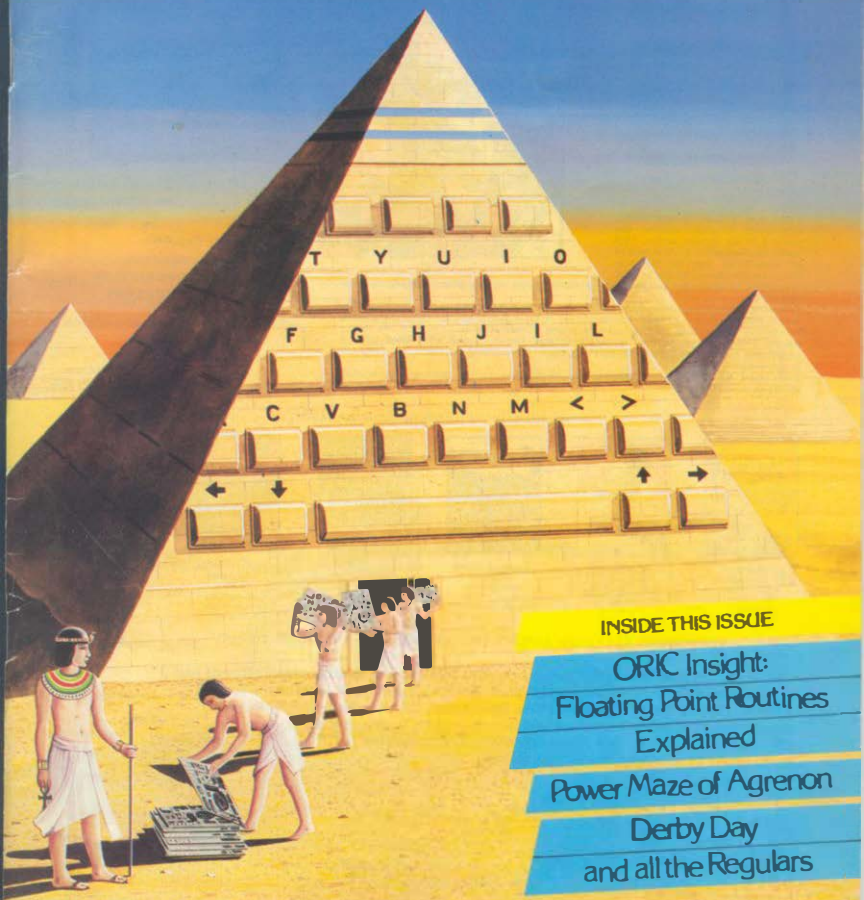
INCORPORATING THE TANSOFT GAZETTE

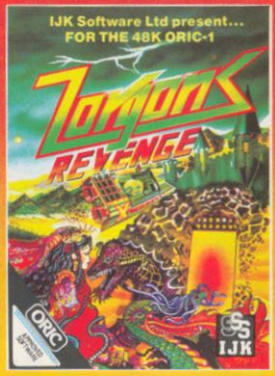
ORIC OWNER

ISSUE 4

OCT NOV



IJK Software and the ORIC-1 Micro bring you more exciting arcade action from the Xenon series...



From the author of the highly acclaimed Xenon-1 comes the second in the Xenon series. This game continues the high standard of excellence set by its predecessor. The amazing super high resolution graphics make this 100% machine code arcade game a must for your collection.

Following their defeat at the hands of the Xenon fleet the Zorgons have captured the Xenon Princess Roz, and have imprisoned her in their castle. You are commissioned to rescue her by scouring the four corners of the Zorgon Empire to capture the magic stones. These stones, guarded by the Quadmags, Terrapods and many other strange beasts, are needed to bridge the bottomless chasm surrounding the castle, enabling you to achieve your goal.

Each one of the many varied stages in this scintillating mission will test your arcade ability as never before.

FEATURES INCLUDE:-

- 100% MACHINE CODE
- SUPERB SOUND EFFECTS
- SUPER-SMOOTH HI-RES GRAPHICS
- MULTI-STAGE ARCADE ACTION
- SKILL LEVELS
- HALL OF FAME

For the 48K ORIC-1 £8.50 inc.

OTHER TITLES IN THE IJK SOFTWARE ORIC-1 RANGE...

XENON-1
100% M/C arcade game for the 48K ORIC-1
£8.50 inc.

INVADERS
Machine code arcade game for 16K & 48K ORIC-1
£7.50 inc.

FANTASY QUEST
Intriguing adventure for 48K ORIC-1
£6.50 inc.

REVERSE
Superb board game for 48K ORIC-1
£6.50 inc.

CANDYFLOSS & HANGMAN
Two top educational programs for 48K ORIC-1

3D MAZE & BREAKOUT
Arcade action for 48K ORIC-1
£7.50 inc.

ALL PRICES FULLY INCLUSIVE OF VAT and P&P – NO MORE TO PAY

Our software has been officially approved by **Oric Products International Ltd.**, and is available from branches of **W.H. Smiths, Laskys, Computers for all, Zappo, Boots, John Menzies, The Spectrum Chain, many other leading department stores, hundreds of independent dealers nationwide, and in 23 countries across the world.** You can also order direct from us – all advertised software is in stock now and will be despatched within 48 hours of receipt of order.

IJK Software Limited

24 HOUR ANSAFONE

9 King Street, Blackpool, Lancs.
Telephone (0253) 21555

ORIC OWNER

Issue 4 Oct/Nov

Features

	page
Oric Insight	10–12
2-D Arrays and Totals	14–16
Hints and Tips	19–21
How to get the most out of your Oric-1	28–29
Oric-tabbing	34–36

Regulars

Editor's Comment	5
Disaster Area	8
Captain Tanex	9
1/0 Page	47–49
Subscription Form	54

News

News Brief	6
Software Scan	7
New Products Review	13

Software

Character Generator Program	17
Phonics	23–25
Frog-Run	26–27
Power Maze of Agreeon	30–33 36–38
Ward–1	42–43
Oric Quickies	39–41 44–46 50–53

Oric Owner Magazine
Editor: *Paul B. Kaufman*
Graphics & Artwork Manager: *Carolyn Groeneveld*
Administration: *Carolyn Groeneveld*
Technical Consultant: *Dr Paul Johnson*
Printers: Heffers Printers Ltd, King's Hedges Road, Cambridge.
Photographic Equipment: Minolta

Oric Owner is published at bi-monthly intervals by Tansoft Ltd, Reg. No. 1632070 Units 1 and 2, Cambridge Techno Park, Newmarket Road, Cambridge. Advertising Rates are available on request.

Oric Insight: The Hidden Oric

How to access the floating point maths routines directly in machine-code

Hints and Tips

A new column of recipes for your Oric, to experiment with

New Products Review

We look at the new Oric 1 MCP-40 colour printer

Captain Tanex

More incredible adventures of our intrepid hero

Phonics

A test for your ears to spot the odd word out

Frog-Run

Can you get the frog to avoid all the traffic hurtling towards it and reach its destination?

ORIC Software

TANSOFT

ultima ZONE

A 100% machine code space trilogy—shoot the Warlons, avoid the bouncing bruises, battle your way through the satellite zone. Requires 48k Oric.

£8.50

inc. V.A.T. post free

The HOBBIT

In co-operation with Melbourne House.

At last, this best selling adventure is available for the Oric based on J. R. R. Tolkien's book. The Hobbit includes a complete text and graphics adventure. Requires 48k Oric.

£14.95 inc. VAT + post free

Our software is available from all Oric dealers and most good software suppliers. In case of difficulty please contact us on Teversham (02205) 2261 or write to us at:

TANSOFT

Unit 1 & 2, Techno Park, Newmarket Road, Cambridge

Editor's Comment

Paul B Kaufman

Summer is the worst time of year for the micro enthusiast. The sun beats down mercilessly on those forced to mow the lawn or lie on deckchairs. There's no welcome snowstorm or flood which means that be bas to stay indoors writing obscure machine code packages or solving adventure games in four dimensions. If you go abroad ten to one there's no mains outlet on the beach and someone's forgotten to pack your aerial lead. Yet in hundreds of offices all over the country software houses are planning their latest games or business programs for the coming autumn and winter rush. This bas been going on for about the last three years yet things are heginning to change.

Nowadays if you go to a computer exhibition you will find less people actually buying equipment or software and more just trying it out and seeing what's available before going to their local W. H. Smiths or Dixons to buy it. This is in contrast to the exhibitions of a few ycars ago such as the Breadboard shows where every exhibitor could expect to clear bis stand before the end of the show and perhaps even sell the stand as well!

It is obvious then that the whole retail market is maturing and rapidly becoming attuned to the consumer rather than the strict hobbyist. In a way a lot of the early magic is disappearing.

I can remember the excitement when my MK14 kit arrived on the doorstep from Science of Cambridge (Pre-Sinclair!) and after a few hurried hours of soldering, switching it on to find it didn't work. This, in a way, was quite gratifying because if it bad worked I would have just sat there and looked at it – there was no software around to do anything with. I remember even being able to get through to Nick Toop or David Johnson-Davies at Science of Cambridge and actually talk to these people who designed the machine. The same goes for when I upgraded to a Tangerine Microtan, I was always talking to Paul Johnson or Barry Muncaster about some technical point or other.

In a way things had to change. The small numbers of these machines would have kept no-one in business very long. Today when you buy a computer it is almost always certain to run first time (don't all write at once!), software of some sort is always available and although you can't get to talk to the designers, there are usually plenty of publications (such as this) to help with your technical problems.

The point of writing this is to show that although there are always complaints and problems, they are usually, in perspective, nowhere as devastating as they could have been a few years ago. So although some of the old magic has gone, today's product is far more complete and reliable than its more temperamental predecessors. Speaking as someone who was there when it started I think I prefer the present solid reliable products (well fairly reliable!) to the old circuit board and bits of wire designers nightmare.

News Brief

Addons from MCP

MCP have released a joystick interface that includes its own speech synthesizer. This is based on a fixed vocabulary chip which gives a limited vocabulary of around 200 words although the sound quality is higher than the phoneme types. The joystick socket is designed for Atari type connections.

MCP are also planning to release in the near future a digitiser (bit pad), an RS232 interface and a multichannel A/D converter.

More information from MCP, Tel (0792) 844465.

Financial Director for Oric

Allan Castle, F.C.A. has joined Oric Products International, manufacturers of the Oric 1 microcomputer, as their Financial Director. He took up the appointment which carries a seat on the Board on July 1.

Mr. Castle has just spent six years in Hong Kong as Financial Controller of the Ednasa Group which is involved amongst other things in shipping, oil field services and property in Singapore, Houston and Brisbane. He began his career as a Chartered Accountant with Coopers and Lybrand where he spent seven years before leaving to become Assistant Treasurer of P.&O.

Oric Sales Increase

Sales of Oric's microcomputer are expected to reach 350,000 during the first year, a six hundred percent increase over its initial projection. It exports in large quantities to France and other European countries and has recently set up joint ventures in Japan and Singapore to cover the Asian and Australasian market. The company is also actively seeking a similar joint venture with an American company.

Software Scan

Paul B Kaufman

Not many titles have arrived for review since the last issue so I suppose that everyone is off on their summer bols.

A company who I have not beard of before, Sector 7 Software, have sent me their Gamespack. It contains seven short games – Laser Station, Obstruction, Mazatronic, Demolition, Milliblox, Noughts and Crosses, and Maths-Test. Although written in Basic they are of a good standard with reasonable graphics and sound. Although they probably won't appeal to the arcade addict they are very suitable for the younger Oric user.

P.S.S. have sent me a preview copy of their 'Ultra.' This is a space invadcrs shoot down type game with a difference. The invaders keep changing shape and moving in various patterns across the screen. The graphics are very good, particularly the star background which seems to move independently of the aliens in front. A very enjoyable game.

House of Death is a follow up adventure game to Zodiac from Tansoft. The adventure takes place in a disused film set which was once used for horror films. As you would imagine there are all sorts of nasties out to get you and plenty of puns and bad jokes which are becoming hallmarks

of Geoff Phillips writing. As with Zodiac you will need several bours to play this adventure through, plus endless patience. Geoff tells me be is working on a proper follow up to Zodiac called – wait for it – Zodiac 2. Apparently this will be written mainly in machine code to speed things up.

I have managed to get hold of a preview copy of Author, the word processing package by John Dawson. It is cassette based and is written entirely in machine code. When you first enter it you are presented with a menu that gives you the various program options. These allow you to create new documents, edit existing ones, save or load to cassette, set up parameters for printers and count the number of words in your text.

Entering text is very simple, you simply type 'W' for write and answer 'Yes' to clear old text out of memory. You then just type in whatever you require.

Editing text is also simple. You can move about your text page by using the cursor keys and you have the ability to insert new text, delete unwanted characters, words or paragraphs and search for specific words.

A clever feature is wordwrap. This ensures that a word is never broken in half if it overflows the cnd of the line. It is always moved down to the next line. There are some more sophisticated features such as auto centring, left and right margin setting and tabs. Author will number pages for you automatically if required and force new pages to be printed when needed. You can also have one paragraph always printed at the top of each form.

Although there isn't room to describe in detail everything that Author can do, in the few hours I have spent using it I was very impressed with its capabilities. It is quite happy to work with most printers including Oric's MCP 40.

Next time I hope to look at Orie Cale and the Hobbit adventure. We are also looking for independent software evaluators to review new Orie software. If you are interested why not drop me a line.

Disaster Area

A number of people have mentioned problems with Oric-Trek in Issue 1. Try adding the line 10 HIMEM #97FF.

In Issue 2 on page twenty we have had a few complaints about the data saving program. Try putting quotes at the beginning and end of each DATA statement.

In Issue 3, the program 'journey of a Space traveller', the introduction was printed at the end of the article instead of the beginning.

Also in Issue 3, the text for the program 'Skiing' should finish after:
1000-1080 Data statements for forming the screen. The rest was a printing error and should be ignored.

Let us know of any disasters you find.

THE EYE OF TANGER

Oric Insight

The Hidden Oric

Paul B Kaufman

This article is condensed from two articles that appeared in Issues 4 and 5 of the original Tansoft Gazette by Jim Rew and Andy Biggs. It will allow you to access your Orics built-in mathematical routines directly through machine code.

Before you can go leaping about in Orics ROMs you will need to know how BASIC handles floating point numbers. All floating point operations take place either directly in memory or in two special floating point accumulators in zero-page, known as FAC1 and FAC2.
The format of a floating point number held in memory is different from when it is held in the floating point accumulators:

IN MEMORY – Sign Packed 5 Byte Format
Exponent Mantissa – 4 bytes Least significant end

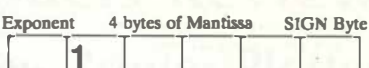
The exponent is in excess 128 (decimal) form, that is:

(a) A true exponent of zero is stored as (hex)	80
(b) A true exponent of one is stored as	81
(c) A true exponent of minus one is stored as	7F

A stored exponent of zero means the number is zero regardless of the value of the mantissa.

The 'binary point' is held to be at the most significant end of the mantissa and the number is 'normalised' such that the most significant bit of the mantissa is a '1'. This being always true this bit is then not explicitly stored, and is instead used to indicate the sign of the mantissa (0=+, 1=–), as indicated by the 'S' above.

IN THE FLOATING ACCUMULATORS – Unpacked 6 Byte Format



The exponent has the same form as above.

The sign is extracted to the SIGN byte (msb 0=+ mantissa)
(msb 1=– mantissa)

The lost 1 is reinstated at the most significant bit of the mantissa. The actual number had the value:

MANTISSA*2 TRUE EXPONENT

In a similar fashion, those ROM routines which store an accumulator value into memory will translate back to the packed format.

Error Handling

There are a number of problems which can cause the ROM routines to abnormally terminate:

(a) Floating point overflow (any operation)
(b) Division by zero
(c) Illegal operations with the transcendental functions (e.g. taking the square root of a negative number)

If you hit one of these problems while you are in a ROM routine, you will probably crash.

The ROM routines will, on error, attempt to enter the BASIC error handler, which would normally print a message, reset lots of zero page parameters and restart the interpreter – fine if you're already in BASIC, but not if you're just using the subroutines. The likely effects are that the error handler will mess up your zero-page usage and either the error print or the restarted interpreter will crash because zero page will not be set up the way the interpreter is expecting! Cheerful isn't it? Since the error call is not vectored you can't catch the problems yourself. So beware!

ROM Routines

Oric's BASIC ROMs contain a number of useful floating point maths routine that can be used from machine code programs. The entry points of some of these routines are listed below, together with enough information for the experimenter to make use of them. These routines make extensive use of zero-page for their operation, however $35 to $80 are available for user storage.

In the list, ACC1 is the primary floating point accumulator used for number crunching and is located in the six bytes $DO to $D5. ACC2 is the secondary floating point accumulator, used for mathematical arguments, and is located in the six bytes $D8 to $DD.

MEM is the address of the first byte of a five byte packed floating point number. The address of MEM is determined by the user. On entering these routines the address of MEM is usually held in the A and Y registers.

ORIC INSIGHT

The following example shows how the routines are used:

Example: Divide 154 by 5, store the result at $1FF0 for future use and print the result.

```
Example
1F00  A9 00     LDA  #$0
1F02  A0 9A     LDY  #$9A
1F04  20 ED D3  JSR  GIVAYF
1F07  20 DD DE  JSR  MOVAF
1F0A  A9 00     LDA  #$0
1F0C  A0 05     LDY  #$5
1F0E  20 ED D3  JSR  GIVAYF
1F11  20 E3 DD  JSR  FDIVT
1F14  A9 1F     LDY  #$1F
1F16  A2 F0     LDX  #$F0
1F18  20 A5 DE  JSR  MOVMF
1F1B  20 D1 E0  JSR  FOUT
1F1E  EA        NOP
1F1F  EA        NOP
1F20  EA        NOP
1F21  A2 01     LDX  #$01
1F23  BD 00 01  LDA  $100,X
1F26  F0 06     BEQ  $1F2E
1F28  9D 10 BE  STA  $BE10,X
1F2C  D0 F5     BNE  $1F23
1F2E  00        BRK

BD 00 01
```

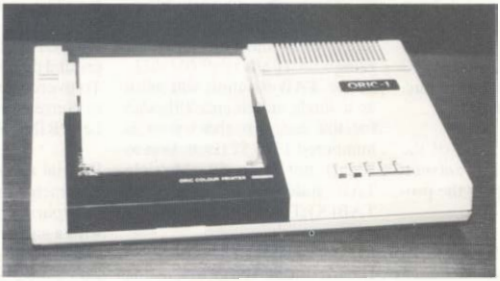
Listing of Entry Points

Name	*Address*	*Function*	*Description*
MOVFM	DE73	Mem→Acc1	Transfer Mem to Acc1. A=MemLO. Y=MemHI.
CONUPK	DD4D	Mem→Acc2	Transfer Mem to Acc2. A=MemLO. Y=MemHI.
MOVMF	DEA5	Acc1→Mem	Transfer Acc1 to Mem X=MemLO. Y=MemHI.
MOVAF	DEDD	Acc1→Acc2	Transfer Acc1 to Acc2.
MOVFA	DECD	Acc2→Acc1	Transfer Acc2 to Acc1.
FDIVT	DDE3	Acc2/Acc1→Acc1	Divide Acc2 by Acc1. Result in Acc1.
FDIV	DDEO	Mem/Acc1→Acc1	Divide Mem by Acc1. A=MemLO. Y=MemHI.
FDIV2	DDDA	Acc2/Mem→Acc1	Divide Acc2 by Mem. A=MemLO. Y=MemHI.
DIV10	DDBF	Acc1/10→Acc1	Divide Acc1 by 10.
SQR	EZZA	SQR Acc1→Acc1	Find the square root of Acc1.
POWER	E231	Acc2 ↑ Mem→Acc1	Acc2 raised to Mem. A=MemLO. Y=MemHI.
INT	DFA5	INT Acc1→Acc1	Find the integer of Acc1.
ABS	DF31	ABS Acc1→Acc1	Find the absolute value of Acc1.
FADD	DA97	Mem+Acc1→Acc1	Add Mem to Acc1. A=MemLO. Y=MemHI.
FADDH	DA79	Acc1+0.5→Acc1	Add 0.5 to Acc1.
FSUB	DA80	Mem-Acc1→Acc1	Subtract Acc1 from Mem. A=MemLO. Y=MemHI.
FMULT	DCB7	Mem*Acc1→Acc1	Multiply Mem by Acc1. A=MemLO. Y=MemHI.
MUL10	DDA3	Acc1*10→Acc1	Multiply Acc1 by 10.
SGN	DF12	SGN Acc1→Acc1	Find the sign of Acc1.
LOG	DC79	LOG Acc1→Acc1	Find the log of Acc1.
NEGOP	E26D	-Acc1→Acc1	Negate Acc1.
RND	E34B	RND Acc1→Acc1	Generate random number.
SIN	E38E	SIN Acc1→Acc1	Find the sin of Acc1.
COS	E387	COS Acc1→Acc1	Find the cos of Acc1.
TAN	E3D7	TAN Acc1→Acc1	Find the tan of Acc1.
ATN	E43B	ATN Acc1→Acc1	Find the atn of Acc1.
GIVAYF	D3ED	A,Y→Acc1	Convert integer to F.P. Y=IntLO. A=IntHI.
QUINTI	D871	Acc1→$D3,$D4	Convert Acc1 to int. $D4=IntLO. $D3=IntHI.
FOUT	EGD1	Acc1→$100+	Convert Acc1 to string. Acc1 destroyed.

FOUT destroys ACC1!

New Products Review

MCP40 Colour Printer/Plotter



This is the first official addon for the Oric-1, a complete printer and pen plotter all in one. Instead of the usual arrangement of heated pins or sparks this printer uses a clever mechanism consisting of four ball-point pens on a rotating cylinder. These pens are black, red, green and blue and so allow colour text or graphics to be printed. Each colour is selected either by pressing one of the buttons at the front of the printer or by entering LPRINT CHR$(29) on your Oric.

The printer has its own power supply cable and comes with a ribbon lead to connect it to the printer socket on the Oric. Make sure that you plug the lead in correctly otherwise your Oric will just freeze.

Apart from the normal printing of characters there is also the facility to control the size of characters in 64 steps. This coupled with the facility to print text in four different directions makes for some very interesting patterns and graphic designs.

The printer uses 4¼" plain roll paper and is selectable for 40 or 80 characters per line. It is also possible to select by switches whether a CR or CR/LF is printed at the end of each line. You also have a switch to select the katakana (Japanese) character set.

There is a graphics mode in the printer, which allows you to draw diagrams and pictures very easily. There is a command similar to the Orics Hi-Res Pattern command. This allows you to draw dotted lines of 16 different types. The resolution across of 480 by the length of the paper allows the detail of pictures to equal or surpass the quality of Hi-Res mode.

Plotting is made very easy by a set of commands that give you absolute and relative drawing, resetting the origin and mixing text with graphics. Most impressive is the facility to draw the axes of a graph automatically including the tick marks along the edge.

All in all, then, this printer is very well suited to the Oric and would be ideal for most hobbyists or even some business users. It would be silly to pretend that it could produce letter quality printing and its speed of writing is a little slow (12 characters per second) but I know of no other printers that give these facilities for the price. So for value for money I don't think you could go wrong here.

'2–D Arrays & Totals'

By L. Bancroft

The program presented demonstrates one solution to the problem of the peculiar TAB function found in the ORIC BASIC ROM, especially when producing 2 Dimensional Arrays.

The text discusses some of the problems and solutions encountered whilst developing the program on an ORIC-1.

ORIC PRODUCTS INTERNATIONAL have acknowledged the problem with the TAB function and have stated that this function only appears to work from position 13 on the screen.

These facts will become relevant if a short history of the program is detailed.

The program owes its origins to the N.E.C.'s 30 Hour BASIC Course (actually Assignment 8.1). The program was originally written for use on an Apple 2.

My special interest is writing Business Programs and upon owning my own ORIC-1 I decided to use this program as part of a larger one. Running this program in its original form on the ORIC provided some interesting, albeit incorrect, results. By experimentation, and a 'phone call to ORIC which confirmed that there was a problem with the TAB, a solution has been found. Also a better understanding of certain functions of the Oric has been gained.

The TAB function will work as a single statement, although for this function the screen is numbered 13 to 51 (from Left to Right) not 0 to 38. Multiple TAB statements e.g. PRINT TAB(X);TAB(Y);TAB(Z).... will not work.

Having found the problem the next step was to find a solution, obviously the next function to try was SPC.

After a few trial statements with SPC it was discovered that some of the problems associated with TAB statements were also present with SPC, plus one or two new ones! These, together with the results of further trials are summarised in TABLE 2.

Having decided that to make the program run correctly and produce a correctly formed table of figures the following functions were deemed necessary: TAB,SPC,LEN,STR$.

When using STR$ it has to be remembered that when used with a numeral an additional space is added in front of the number for the + or − character. E.g. Line 160 causes the current value of J to be printed 5 spaces from the last J, not 6.

It will be seen from Table 2 that SPC statements in loops where the FOR/NEXT Value is to be printed do not operate correctly. To overcome this it is necessary to convert the value to a string, i.e. PRINT STR$(J).

By trial and error the necessary statements were formulated and incorporated into the program with a successful result.

To summarise: The TAB function on the ORIC-1 can only be used as a single statement. The TAB positions on the screen are 13 to 51, and TAB position below 13 will be printed at 13 (e.g. TAB(5) will appear at the extreme Left of the screen exactly as would TAB(13)). Table 1 gives some examples of statements that will work and some that won't.

For multiple printing on one line either SPC alone or following a TAB statement is required. It has to be remembered that when using SPC in loops, where the FOR/NEXT value is required to be printed that this numeral has to be converted to a string. From personal experience I have concluded that the SPC function, within a loop for multiple printing on one line, only operates correctly with strings or with numbers already designated storage labels, e.g. Lines 200, 260.

'2–D Arrays & Totals'

In conclusion the lines that have been altered in the program are listed below together with their function:

150–160 Print the Headings
180–210 Print Day No., Data, Daily Totals
230–260 Print Product Totals & Gross Total

Table 1

TAB Functions Correctly	*TAB Functions Incorrectly*
10FOR X ± 13TO51 20 PRINT TAB(X)X 30NEXTX ?TAB(X-LEN(T$))T$?TAB(X-LEN(STR$(X)))X ?TAB(X-LEN(T)T	Any Multiple TAB Statement. E.g. PRINTTAB(X);TAB(Y);etc. FOR X+13TO51 STEP 5 ?TAB(X)X;:NEXT X FOR X = 13TO51 STEP 5 ?TAB(X)"A";:NEXTX

Table 2

SPC Functions Correctly	*SPC Functions Incorrectly*
10FOR X = 1 TO 5 20?SPC(X)X 20?SPC(5)X 20? SPC(10-LEN(T$))T$; 20? SPC(10-LEN(STR$(X)))"A"; 20? SPC(10-LEN(STR$(T)))T; 20 ?SPC(10-LEN(STR$(X)))STR$(X); 30NEXTX	10FOR X = 1 TO 5 20 ?SPC(10-LEN(STR$(X)))X; 30NEXTX 10FOR X = 1 TO 5 20? SPC(5)X; 30NEXT X The above examples will print the first value correctly but subsequent values will be printed at SPC(5+1) i.e. SPC(6)

(The above examples will function correctly/incorrectly whether in loops or not.)

'2–D Arrays & Totals'

```
 10   REM "2-D ARRAYS & TOTALS"
 20   DIM N(7,6) : DIM A(6) : DIM B(6)
 30   REM ***READ & TOTAL ROUTINE***
 40   FOR I = 1 TO 5
 50   FOR J = 1 TO 4
 60   READ N(I,J)
 70   A(J) = A(J) + N(I,J)
 80   B(I) = B(I) + N(I,J)
 90   NEXT J
100   T = T + B(I)
110   PRINT : NEXT I
120   REM ***PRINT ROUTINE***
130   PRINT TAB(14)"DAY";SPC(8)"PRODUCT"
140   PRINT TAB(14)"---";SPC(8)"--------"
150   PRINT TAB(19)"1";
160   FOR J = 2 TO 4 : PRINT SPC(7-LEN(STR$(J)))STR$(J); : NEXT J :
      PRINT SPC(5)"TOTALS"
170   FOR X = 13 TO 50 : PRINT "-"; : NEXT X
180   FOR I = 1 TO 5 : PRINT TAB(14)I;
190   FOR J = 1 TO 4 : PRINT SPC(6-LEN(STR$(N(I,J))))N(I,J); : NEXT J
200   PRINT SPC(9-LEN(STR$(B(I))))B(I)
210   PRINT : NEXT I
220   FOR X = 13 TO 50 : PRINT "-"; : NEXT X : PRINT
230   PRINT TAB(22-LEN(STR$(A(1))))A(1); : J=2
240   PRINT SPC(6-LEN(STR$(A(J))))A(J); : J=J+1
250   IF J< 5 THEN 240
260   PRINT SPC(9-LEN(STR$(T)))T
270   FOR X = 13 TO 50 : PRINT "-"; : NEXT X : END
280   DATA 500,300,20,25,600,700,40,0,200,550
290   DATA 60,20,250,450,100,5,400,200,100,11
```

Character generator program

By Carsten Skjerk

This character generator program is dumped directly to the printer using LLIST.

The following changes are needed for the character generator to run on the 16K:

Line 10: #97FF should be #17FF
Line 120: #B400 should be #3400
Lines 190, 290, 420: 48123 should be 15335
Line 315: the CSAVE – addresses should be A#3400, E#3B80

General notes;
Line 50, 395: 6 spaces

```
 1 REM ` 1983 CARSTEN SKJERK
 5 REM ** CHARACTER GENERATOR **
 10 INK7:PAPER1:HIMEM#97FF
 20 (X,S:A$=CHR$(255):TEXT:PRINTCHR$(20);
 30 PRINT:PRINTA$A$A$A$A$A$A$
 40 FORX=1TO8
 50 PRINTA$"       "A$
 60 NEXTX
 70 PRINTA$A$A$A$A$A$A$A$
 80 PRINT:PRINT"Input the character to be redefined"
 90 B$=KEY$:IFB$=""THEN90
 91 IFA5C(B$)<320RA5C(B$)>120THEN90
 100 PRINT"Characterset 0 or 1 ?
 110 GETC$:IFC$<>"0"ORC$>"1"THEN110
 115 C=VAL(C$)
 120 A=#B400+VAL(C$)*#400+BaA5C(B$)
 130 PRINT:PRINT"Press 's' to set a pixel and 'd' to   delete a pixel."
 132 PRINT"Press Space Bar to clear character."
 135 PRINT"Press 't' to save current characterset tape."
 140 PRINT:PRINT"Use the arrow-keys to move the cursor.";
 150 PRINT:PRINT"Press [RETURN] to store the character.";
 160 FORX=1TO22:PRINTCHR$(11);:NEXT
 180 PRINTCHR$(9)CHR$(9)CHR$(9);
 190 Z=48123:G=126
 200 FORX=0TO7
 210 Y=PEEK(X+A)
 220 IFYAND32THENPOKEZ+X,(G)ELSEPOKEZ+X,32
 230 IFYAND16THENPOKEZ+X+1,(G)ELSEPOKEZ+X+1,32
 240 IFYANDBTHENPOKEZ+X+2,(G)ELSEPOKEZ+X+2,32
 250 IFYAND4THENPOKEZ+X+3,(G)ELSEPOKEZ+X+3,32
 260 IFYAND2THENPOKEZ+X+4,(G)ELSEPOKEZ+X+4,32
 270 IFYAND1THENPOKEZ+X+5,(G)ELSEPOKEZ+X+5,32
 280 Z=Z+39:NEXT
 290 Z=48123
 300 GETC$:B=A5C(C$)
 310 IFB(@ORB)11ANDC$()"s"ANDC$()"d"ANDC$()"t"ANDB()13ANDC$()" "THEN300
 315 IFC$="t"THENCSAVE"CHARACTERSET",A#B400,E#B880
 320 IFB=8ANDPOS(0)3THENZ=Z-1:PRINTC$;
 330 IFB=9ANDPOS(0)<BTHENZ=Z+1:PRINTC$;
 340 IFB=1OANDPEEK(616)<1DTHENZ=Z+40:PRINTC$;
 350 IFB=11ANDPEEK(616))3THENZ=Z-40:PRINTC$;
 370 IFC$="s"THENPOKEZ,126
 380 IFC$="d"THENPOKEZ,32
 390 IFB=13THEN410
 395 IFC$=" "THENFORX=2TO9:PLOT2,X,"       ":NEXT
 400 GOTO300
 410 PRINTCHR$(17);
 420 Z=48123
 425 IFC=1THENPLOT14,5,9ELSEPLOT14,5,8
 426 PLOT15,5,5$
 430 FORX=0TO7
 440 C=0
 450 FORY=0TO5
 460 D=PEEK(Z+Y+(40aX))
 470 IFD=126THENC=C+2^(5-Y)
 480 NEXTY
 485 C$=RIGHT$(STR$(C),LEN(STR$(C))-1)
 490 POKEA+X,C:PLOT9,X+2," "+C$
 500 NEXTX
 505 PLOT1,25,"Press [RETURN] to continue."
 510 GET A$
 520 PRINTCHR$(17)CHR$(20):RUN
```

Hints & Tips

By D. Lam

Program 1 deals with Discounting (i.e. compounding in reverse to find the present value of a given amount of money receivable in the future).

Program No. 1

```
5   REM DISCOUNTING
7   CLS: PRINT
10  INPUT"ENTER
    AMOUNT
    RECEIVABLE IN Y
    NUMBER OF YEARS
    TIME =£";M
15  PRINT
20  INPUT"NUMBER OF
    YEARS"; Y
22  PRINT
25  INPUT"RATE OF
    INTEREST PER
    ANNUM=%";R
30  PRINT
40  PV=M*1/(1+R/100)↑Y
50  PRINT"GIVEN
    ";R;"PERCENT OF
    INTEREST PER
    ANNUM, THE
    PRESENT VALUE OF
    £";
55  PRINT M;
    "RECEIVABLE IN
    ";Y;"YEARS TIME IS
    £";INT(PV*100)/100
60  END
```

Example:

GIVEN 12 PERCENT OF INTEREST PER ANNUM, THE PRESENT VALUE OF £1250 RECEIVABLE IN 5 YEARS TIME IS £709–28

Program 2 is an effective way to calculate (a) the sales units/value in order to break even, and (b) the sales units/value to achieve the budgeted profit for the year.

Program No. 2

```
5   REM BREAKEVEN
    CALC
10  CLS: PRINT
20  INPUT"SELLING PRICE
    PER UNIT =£";SP
25  PRINT
35  INPUT"VARIABLE
    COST PER UNIT
    =£";VC
40  PRINT
45  INPUT"TOTAL FIXED
    COST PER ANNUM
    =£";FC
50  PRINT
60  INPUT"WHAT IS YOUR
    EXPECTED PROFIT
    FOR THIS YEAR ?
    £";EP
65  PRINT
80  BEU=FC/(SP-VC)
90  EPU=(FC+EP)/(SP-VC)
95  PRINT:PRINT:PRINT
100 PRINT"(1)BREAKEVEN
    POINT =
    ";INT(BEU);"UNITS"
105 PRINT
110 PRINT"BREAKEVEN
    SALES =£";INT (BEU*
    SP)
115 PRINT
120 PRINT"(2) TO
    ACHIEVE THE
    EXPECTED PROFIT,
    YOU MUST BE ABLE
    TO SELL ";
125 PRINT INT(EPU);
    "UNITS"
130 PRINT
135 PRINT"SALES
    REVENUE TO
    GENERATE THE
    EXPECTED PROFIT
    =£";INT(EPU*SP)
140 END
```

Example:

SELLING PRICE PER UNIT =£? 5
VARIABLE COST PER UNIT =£? 3
TOTAL FIXED COST PER ANNUM =£? 12000
WHAT IS YOUR EXPECTED PROFIT FOR THIS YEAR ? £? 9500

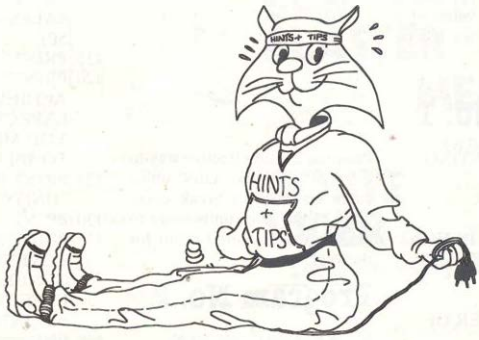
(1) BREAKEVEN POINT = 6000 UNITS
BREAKEVEN SALES =£30,000

(2) TO ACHIEVE THE EXPECTED PROFIT, YOU MUST BE ABLE TO SELL 10750 UNITS

SALES REVENUE TO GENERATE THE EXPECTED PROFIT =£53,750

Hints & Tips

By D. Lam



1. Type HIRES and draw a few lines all over the screen (and circles, etc if you want). Then type TEXT. Use the CURSOR-DOWN key to get down to the last line on the screen. Then press ESC and finally press the DEL key . . . . . .

2. Locations 520 and 521 return, if PEEKED, different numbers depending on which key is pressed (much better than KEY$ or GET$) location 520 holds most of the keyboard whereas 521 changes if the SHIFT or CTRL keys are pressed (the two SHIFT keys return different values).

3. In HIRES, if you have 'FILL'ED parts of the screen with colours, etc you will have noticed that you cannot draw a line or a CIRCLE over that part. To clear a 'window': set the cursor to the appropriate point and FILL the desired area with a value of 64.

4. I have discovered a way of retrieving NEW'ed programs and am working on a short machine code routine to bring back a BASIC program which has been NEWed (providing no errors have since occurred or no program lines have been entered) I am also working on a routine (in BASIC) which will highlight all REM statements in a program with any desired attribute. More details later.

5. CALL 62509 simulates pulling out and pushing the power plug in again. This saves wear on the power socket.

6. The alternate character set graphics are made up as follows:

1	2
4	8
16	32

So, to get a particular shape, add up the corresponding blocks and add 32.

eg. to get:

the code is (1+8+16)+32=57

Hints & Tips

By C R Burnham

I have discovered that the following addresses perform the specified functions on the ORIC 1 48K Micro. I thought you might like to include them in your magazine for other Users.

Decimal Address	Hex Address	Function
621	#26D	Address of text screen (Default 48000) but can be DOKED with 48280 for example which will stop the top 7 lines from scrolling (ie. 48000 + (7 × 40) = 48280)
623	#26F	Text lines per screen Default 27) but can be POKED with 20 to 'reduce' the text screen size.

So to scroll line 13 to 27 but leave lines 1 to 12 constant:

```
10  CLS: FOR A=1 TO 12: PRINT "LINE"A: NEXT
20  DOKE 621, 48480: REM BASE ADDR + 12 LINES
30  POKE 623,14: REM 14 LINES TO SCROLL
40  FOR A=1 TO 100
50  PRINT "THIS LINE WILL SCROLL"
60  NEXT
70  DOKE 621,48000: REM RESET BASE ADDR
80  POKE 623,27: REM RESET LINES/SCREEN
```

ISO\PAGE

Dear Sir,
re *Forth on the ORIC*
I am very excited about FORTH on the ORIC. It really is an excellent implementation of fig-FORTH.

I also enjoy 'Starting FORTH' by Leo Brodie which is recommended in the introduction to the FORTH Programming Manual.

As is pointed out there, there are differences between poly-FORTH and fig-FORTH. One major difference is in the use of the word CREATE. If you log using it in an attempt to copy Brodie's examples disaster strikes! It is possible to define a defining word which has the same effect as poly-FORTH CREATE.- : STORE <BUILDS DOES>; eg the phrase—
STORE LIST n_1, n_2, n_3, will create a dictionary entry called LIST followed by the numbers n_1, n_2 and n_3.

Subsequent use of the word LIST returns the address of the first value. Thus--LIST @ will put n_1 on the top of the STACK. LIST 2+ @ will return n_2
LIST 4+ @ will return n_3

The various examples on the ORIC FORTH tape which make use of calls to functions in ROM are fascinating. Can we persuade ORIC to publish a list of entry points and parameters? Then we really could demonstrate the combined power of FORTH and ORIC!

Best wishes to ORIC OWNER magazine. I'm looking forward to receiving my next copy.

C. Gavin Fairpo

Dear Sir,
After acquiring my ORIC-1 through some difficulty, I must say it was well worth the wait.

As it is not my first personal micro, I know enough to recognize its excellent properties. Everything is a plus here, including color (via RCB monitor), sound (hooked up to the stereo) and the microsoft basic is certainly up to standard. The super-fast graphics commands are specially useful.

But, everything considered, ORIC-1's full computing potential is not realized without essential peripherals.

Can you please suggest a few economical printers (SEIKOSHA AP80A?) and disk drives easily available that may be plugged in with little or no need for interfacing?

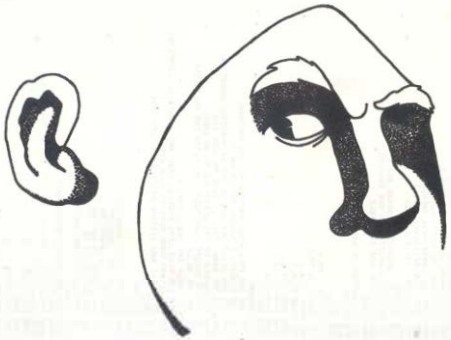
How about a circuit diagram to connect a standard switched joystick to 9 pin D-plug?

Thanks for your attention.

Rodolfo L. Laygo
Manama, Bahrain

Phonics

By S. Lucas



This program was written in conjunction with a teacher of 5-8-year olds. Its aim is to help young children to develop an understanding of the ideas of PHONICS, in a way which is both fun and educational. The program will work in a 48K machine, although I have not tried it in a 16K machine.

The listing contains full instructions in the form of REMarks as to how the program works and how it can be changed to suit the user. I have tried to keep its operation as structured as possible and have included no GOTO statements. The main section of the program occurs between lines 120 and 210, the following lines all being subroutines.

In use, the child will be presented with four boxes containing 3 similar words and one "ODD MAN OUT". An arrow will jump from one word to the next at a rate which is determined by line 1950. This can easily be speeded up if required. The child must then press any key when the arrow points at the odd one out. This was felt to be better than asking children of this age to use the keyboard more fully. They will be given a different response for a right or wrong answer. The computer chooses 10 questions at random from 60 alternatives. After the ten questions are up, the child will be told their score and asked to enter their name. It should therefore be quite easy to extend the program to keep a 'HIGH SCORES' league which could be useful in a class situation. Scores of over 5 out of 10 get a comment 'good' whilst over 7 get 'very good'. They will then be asked if they want to play again.

To alter the questions, all that is necessary is to alter the data from line 330 onwards. Each line of data contains the 4 words followed by the number (out of the 4 alternatives) of the right answer. The boxes will hold words containing up to 5 characters only.

Phonics Listing

Software

```
1 REM ### PHONICS PROGRAM ###
2 REM ### SUITABLE FOR 5/7 YEAR OLDS ###
3 REM ### WRITTEN BY S. W. LUCAS ###
4 REM ### (C) 1983 ###
5 AA$="Phonics Program"
6 AB$="written by"
7 AC$="S.W. Lucas 1983"
10 HIRES
11 PAPER7:INK4
15 REM TITLES ON THE HIRES SCREEN
20 CURSET20,20,3
30 FOR A = 1 TO LEN(AA$)
31 CHAR ASC(MID$(AA$,A,1))+128,0,1
32 CCR=CV)0.0.0
33 NEXT A
35 CURSET 20,39,3
36 FOR A = 1 TO LEN(AB$)
37 CHAR ASC(MID$(AB$,A,1))+128,0,1
38 CCR=CV)0.0.0
41 NEXT A
45 CURSET 20,58,3
46 FOR A = 1 TO LEN(AC$)
47 CHAR ASC(MID$(AC$,A,1))+128,0,1
48 CURMOV 10,0,0
49 NEXT A
100 DIM A$(68,4),B(68),C(10)
110 GOSUB 270
120 GOSUB1280
130 GOSUB 1710
140 GOSUB1330
150 SC=0
160 IF (B$="N" OR B$="n") THEN END
200 RUN
210 END
220 REM THIS SECTION OF THE PROGRAM
230 REM READS THE DATA INTO THE 2-DIMENSIONAL ARRAY A$(X,Y)
240 REM EACH QUESTION HAS 4 ALTERNATIVES
250 REM FOLLOWED BY THE NUMBER OF THE CORRECT ALTERNATIVE
260 REM THIS ANSWER IS HELD IN THE ARRAY B(X)
270 FOR X =1 TO 68
280 FOR Y = 1 TO 4
290 READ A$(X,Y)
300 NEXT Y
310 READ B(X)
320 NEXT X
330 DATA cat,eat,bat,let,3
340 DATA eat,meat,tub,seat,3
350 DATA cee,dee,hoo,see,3
360 DATA hill,ball,call,fall,1
370 DATA miss,put,nut,hut,1
380 DATA hiss,miss,mat,kiss,3
390 DATA cee,dee,hoo,see,3
400 DATA cam,pen,pan,pam,1
410 DATA tap,lap,map,bat,4
420 DATA bid,rid,dun,sid,3
430 DATA had,rid,mad,sad,2
440 DATA rod,rat,cod,pod,2
450 DATA bake,cake,rake,plate,3
460 DATA late,ball,hate,gate,2
470 DATA  fill,Hill,call,pill,3
480 DATA shell,well,pull,bell,3
490 DATA pull,bull,hull,call,3
500 DATA hopp,shut,shop,pop,22
510 DATA shed,bed,red,hop,4
520 DATA pop,lip,pip,tip,1
530 DATA cee,dee,hoo,see,3
540 DATA lad,hit,sit,pit,1
550 DATA fan,sun,fun,bun,1
560 DATA bid,rid,dun,sid,3
570 DATA face,pace,plate,race,1
580 DATA meat,heat,seat,sit,4
590 DATA fair,hair,chair,chap,3
600 DATA lid,bin,pin,tin,1
610 DATA cut,fish,put,hut,2
620 DATA take,lake,cake,hunt,3
630 DATA car,far,call,bar,3
640 DATA wake,pin,rake,make,2
650 DATA cam,pen,pan,pam,1
660 DATA ant,pant,hen,rant,3
670 DATA mast,last,fish,past,3
680 DATA lock,tank,rock,sock,3
690 DATA lamp,vest,ramp,camp,3
700 DATA vest,rest,pest,lock,3
710 DATA nine,mine,pine,kite,3
720 DATA pram,ham,land,ham,1
730 DATA land,sand,hand,pin,4
740 DATA hat,pin,pop,top,1
750 DATA mark,park,lack,land,4
760 DATA pond,jet,pet,met,1
770 DATA ride,side,tide,land,4
780 DATA play,clay,say,pet,4
790 DATA can,pan,plug,ran,3
800 DATA pad,mad,dad,cat,3
810 DATA map,sap,cap,cat,4
820 DATA late,Kate,hate,cat,3
830 DATA Tate,cap,gate,rate,2
840 DATA car,far,bar,ball,3
850 DATA cat,bike,like,mike,1
860 DATA late,plank,ate,plate,2
870 DATA bank,pull,plank,rank,2
880 DATA shade,glade,blade,glad,4
890 DATA chop,cop,cot,shop,3
900 DATA wait,wake,bake,take,1
910 DATA lake,full,make,rake,2
920 DATA shut,run,rut,nut,2
930 RETURN
940 REM THIS SUBROUTINE DRAWS THE 4
950 REM BOXES WHICH HOLD THE CHOICES
960 REM WHICH ARE AVAILABLE
970 HIRES
971 REM NEEDED TO RESET COLOURS
972 PAPER7
973 INK1
974 YY=20
975 FOR Q= 1 TO 4
977 CURSET YY,50,1
980 DRAW 39,0,1
990 DRAW 0,30,1
1000 DRAW -39,0,1
1010 DRAW 0,-30,1
1020 YY=YY+52
1030 NEXT Q
1040 RETURN
1050 REM THE FOLLOWING SECTION OF THE PROGRAM
1110 REM CHOOSES A RANDOM SELECTION OF
1260 REM TEN QUESTIONS OUT OF THE 68 POSSIBLE QUESTIONS
1350 REM THE ARRAY C(X) IS USED TO HOLD THE
1360 REM NUMBER OF THE QUESTIONS TO BE ASKED
1370 REM TO CHOOSE MORE QUESTIONS ALTER THE NEXT LINE
1380 FOR X=1 TO 10
1390 LET C(X)= INT(RND(1)*68)+1
1391 REM THE NEXT LINE ENSURES THAT TWO IDENTICAL QUESTIONS
1392 REM DO NOT COME NEXT TO EACH OTHER
1393 IF C(X)>1 AND (C(X)=C(X-1)) THEN GOTO 1390
1400 NEXT X
1410 RETURN
1430 SC=0
1440 REM THIS SETS SCORE =0
1450 REM THE NEXT SECTION
1455 REM SETS 10 QUESTIONS
1457 REM CHOSEN AT RANDOM
1460 FOR TT = 1 TO 10
1480 REM THIS CLEARS THE SCREEN
1490 HIRES:
1495 REM DRAW THE 4 BOXES AGAIN
1499 GOSUB 970
1419 GOSUB1470
1429 GOSUB1760
1479 GOSUB2140
1489 NEXT TT
1499 GOSUB2460
1469 RETURN
1470 G=B(A)
1480 A=C(TT)
1490 REM A =C(TT) HOLDS THE NUMBER
1500 REM OF THE CORRENT QUESTION
1510 REM G=B(A) HOLDS THE ANSWER
1520 REM TO THE CURRENT QUESTION
1530 REM THE NEXT SECTION PRINTS THE
1540 REM WORDS IN THE BOXES
1541 PP=22:REM SET X POSITION OF CURSOR FOR THE FIRST WORD
1545 FOR P= 1 TO 4:REM COUNTER FOR 4 WORDS
1550 CURSET PP,60,3
1555 W$=A$(A,P)
1560 FOR Z= 1 TO LEN(W$)
1610 CHAR ASC(MID$(W$,Z,1))+128,0,1
1620 CURMOV 7,0,0
1630 NEXT Z
```

```
1635 PP=PP+52:REM MOVE CURSOR FOR NEXT WORD
1640 NEXT P
1650 RETURN
1670 REM THIS SUBROUTINE IS TO DEFINE
1680 REM THE CHARACTERS TO BE USED
1690 REM AS A POINTER
1700 REM TO POINT TO THE WORDS
1710 GF$=CHR$(129)
1720 GG$= CHR$(94)
1730 GH$= CHR$(124)
1740 RETURN
1760 D=32
1770 E=100
1780 REM PROCEEDURE TO MOVE ARROW ACROSS
1790 REM THE SCREEN . TO POINT AT THE 4 BOXES IN TURN
1800 REM UNTIL ANY KEY IS PRESSED
1820 REPEAT
1830 REM D = X POSITION AND E = Y POSITION
1840 REM OF THE ARROW
1850 PING
1860 REM SOUNDS PING EACH TIME THE ARROW MOVES
1900 CURSET D,E,3
1910 CHAR94,0,1
1920 CURMOV 0,8,3
1930 CHAR124,0,1
1950 WAIT 150
1951 REM TO MAKE THE ARROW PAUSE BEFORE MOVING TO THE NEXT BOX
1960 REM NOW ERASE ARROW
1970 CURSET D,E,3
1980 CHAR 94,0,2
1985 CURMOV0,8,3
1988 CHAR124,0,2
2000 D=D+52
2010 REM MOVE ARROW
2015 REM F= POSITION ARROW STOPS IN
2020 REM IF ARROW GOES OFF SCREEN, RESET IT TO POSITION 1
2040 IF D>198 THEN D=32
2050 REM RESET CHOICE WHEN ARROW GOES OFF SCREEN
2070 S$=KEY$
2075 REM TESTS FOR ANY KEY BEING PRESSED
2080 UNTIL S$<>""
2085 REM SET VALUE OF F
2090 IF D=32 THEN F=4
2095 IF D=84 THEN F=1
2100 IF D=136 THEN F=2
2110 IF D=188 THEN F=3
2120 IF F>4 THEN F=4
2125 IF F <1 THEN F=1
2130 RETURN
2140 HIRES
2150 PAPER7
2160 INK 4
2180 REM CHECK FOR CORRECT RESPONSE
2190 IF F=B(A) THEN GOSUB2210 ELSE GOSUB2310
2200 RETURN
2209 REM SUBROUTINE FOR WRONG ANSWER
2210 SC=SC+1
2220 REM INCREMENTS THE SCORE
2230 REM NOW PRINT ON THE SCREEN RIGHT
2235 RR$="right"
2236 CURSET 89,75,3
2240 FOR Z= 1 TO LEN(RR$)
2242 CHARASC(MID$(RR$,Z,1)),0,1
2245 CURMOV8,0,0
2248 NEXT Z
2250 REM NOW GIVE A SOUND FOR RIGHT ANSWER
2252 FOR X=1 TO 150
2255 SOUND1,X,15
2260 NEXT X
2262 FOR X=150 TO 1 STEP -1
2264 SOUND 1,X,15
2265 NEXT X
2267 SOUND 1,0,0
2300 RETURN
2310 WW$="wrong"
2315 REM TO WRITE WRONG ON THE SCREEN
2320 CURSET 89,75,3
2325 FOR Z= 1 TO LEN(WW$)
2330 CHARASC(MID$(WW$,Z,1)),0,1
2335 CURMOV 8,0,0
2336 NEXT Z
2340 WW$="It was "+A$(A,B(A))
2341 REM TO PRINT THE CORRECT ANSWER
2344 CURSET 50,150,3
2346 FOR Z= 1 TO LEN(WW$)
2348 CHARASC(MID$(WW$,Z,1)),0,1
2349 CURMOV 8,0,0
2350 NEXT Z
2353 REM ZAP SOUND FOR WRONG ANSWER
2355 ZAP
2360 WAIT 200
2370 REM PAUSE TO LET CHILD READ WHAT THE ANSWER SHOULD HAVE BEEN
2390 RETURN
2400 TEXT
2410 REM ROUTINE FOR PRINTING RESULTS
2420 LET A$=""
2430 IF SC>5 THEN A$="good"
2440 IF SC>7 THEN A$=" very good"
2450 PAPER 7
2460 INK 0
2470 PRINT:PRINT:PRINT:PRINT:PRINT" your score was ";SC;" out of 10
"
2480 PRINT:PRINT:PRINT"What is your name ";
2490 INPUT NA$
2500 PRINT:PRINT:PRINTCHR$(129);"Thank you for playing ";CHR$(132);
NA$
2510 PRINT:PRINT:PRINTCHR$(130);"Do you want to play again?"
2520 GETB$
2530 RETURN
```

Phonics Listing

Software

Frog Run

By M. Caldwell

Frog Run, this program is better known to most as 'FROGGER'. The game fits into all models, the only thing that needs to be changed are the addresses for the user definable graphics. After typing in the program, and before you run it make sure that

a) THE KEYBOARD CLICK ISON.
b) THE CURSOR IS FLASHING.

On running the program you will find a different way of showing a title. The way the title (and the traffic) are done is by using the LEFT$ and the RIGHT$ functions (chapter 8 in the ORIC manual).

WHEN desired, press any key and the game begins. The game is in two parts, when the screen clears you are on level one, this bit is the simple bit (or to some it is). When you reach 500 points the game moves on to level two (or the rush hour). The game is very similar to the original but bas a few differences, e.g. it has no lake. It has ten lanes of traffic, but can be improved to twelve by adding this. . .

```
128 PLOT1,15,B$:PLOT1,-13,B$
```

THESE two lines make the game a lot harder on both levels, as they plot a lane of traffic on either side of the snake (or snakes on level two).

THERE is no time limit (yippee I hear you shout) but you only get five lives. My highest score is displayed on the screen but I know that will be beaten by the ED.

Here is what each line (well nearly each line) does.

1–90 Sets variables and GOSUBs all the routines.
120–280 The main loop, which moves, plots the traffic, moves and plots the man, checks to see if anything has been bit.
3000–3020 Defines A$, B$, C$
4000–4030 Lose a life
5000–5060 Sets up the screen
6000–6100 No lives left
7000–7060 The rush hour
8000–8080 User definable graphics
9000–9190 Title and instructions

Variables

sc=SCORE L=LIVES
HS=HIGH SCORE A,B SCREEN POSITIONS

HAPPY HOPPING

Frog Run Listing

```
1 CLS
2 PRINTCHR$(6)CHR$(17)
4 HS=700
5 GOSUB6000'U.D.C
8 GOSUB9000
10 SC=0
15 L=5
18 PAPER7:INK0
20 GOSUB5000'SCREEN
90 GOSUB3000
95 FORA=2TO26:S=INT(RND(1)*2)+1:PLOT0,A,S:NEXT
96 PLOT0,14,16
100 A=26
110 B=19
112 PLOT9,0,STR$(SC)
120 PLOT3,A,"  "
125 PLOT1,24,A$:PLOT1,22,B$: PLOT1,12,A$:PLOT1,10,B$
130 X$=KEY$
140 IFX$=""THEN160
150 IFX$=";"THENA=A-1
165 Q=SCRN(B,A)
170 IFX$="."ANDA<26THENA=A+1
175 PLOT1,14,C$
180 IFX$="Z"ANDB<>1THENB=B-1
190 IFX$="X"ANDB<>37THENB=B+1
200 PLOT1,20,A$:PLOT1,18,B$: PLOT1,8,A$:PLOT1,6,B$
210 PLOT1,16,A$: PLOT1,4,A$
220 Q=SCRN(B,A)
225 IFQ=81THENSC=SC+100:PING:PLOT9,0,STR$(SC):GOTO100
230 IFQ<>32THEN4000'LOOSE A LIFE
235 IFSC=300THENGOSUB7000
240 PLOTB,A,"ß"
250 L$=LEFT$(A$,1):R$=RIGHT$(A$,37)
255 L2$=L$+T$(C$,37):R2$=RIGHT$(C$,1)
260 L1$=LEFT$(B$,37):R1$=RIGHT$(B$,1)
270 A$=R$+L$:B$=R1$+L1$
275 C$=R1$+L2$
280 GOTO120
3000 A$="Z                    'Z     'Z   "
3010 B$="Ä              Ä       Ä     Ä   "
3015 C$="/
3020 RETURN
4000 XPLODE:WAIT50
4001 L=L-1:IFL=0THEN6000
4010 PLOT21,0,STR$(L)
4030 GOTO100
5000 CLS
5010 PLOT3,0,"SCORE ":PLOT1,0,1:PLOT15,0,"LIVES 5":PLOT10,0,7:PLOTO
.0,20
5020 FORA=0TO37:PLOTA,2,"-":NEXT
5025 PLOT26,0,"HI-SCORE"
5028 PLOT34,0,STR$(HS)
5030 PLOTO,2,0
5040 PLOTO,14,16
5050 FORA=1TO26:S=INT(RND(1)*5)+1:PLOTO,A,S
5060 RETURN
6000 PLOT22,0,"0":
6010 WAIT300:CLS
6020 PRINT:PRINT:PRINT:PRINT"HARD LUCK YOU RAN OUT OF LIVES "
6030 PRINT:PRINT"YOU GOT A SCOVE OF ";SC" POINTS"
6040 IFSC<HSTHENPRINT:PRINT:PRINT"YOU BEAT THE HIGH SCORE OF "HS"
POINTS"
6050 IFSC<HSTHENHS=SC
6060 PRINT:PRINT:PRINT"ANOTHER GO Y/N"
6070 GETA$
6080 IFA$="N"THENEND
6090 IFA$="Y"THENGOTO8
6100 GOTO6070
7000 PLOT3,14,"THE RUSH HOUR IS ABOUT TO BEGIN!!"
7010 A$="      'Z  ')   'Z   'A  'Z   'Z     "
7020 C$="/     /   /    /     /   /    /     "
7040 B$="   Ä  Ä   Ä    Ä    '(    Ä  Ä     "
7045 SC=SC+50:PLOT9,0,STR$(SC)
7050 WAIT300:PLOT3,14,"
7060 RETURN
8000 FORA=46360TO46423
8010 READD
8020 POKEA,D
8040 NEXT
8045 RETURN
8050 DATA1,3,5,15,7,2,4,8,32,48,40,60,56,16,8,4,63,63,63,63,63,63,2
4,24
8060 DATA48,48,48,63,63,63,6,6
8070 DATA3,3,3,63,63,63,24,24,48,48,48,63,63,63,6,6
8080 DATA0,0,31,31,31,63,1,1,0,0,60,60,60,63,32,32
9000 PAPER0:INK7:CLS
9010 Z$="FROG RUN      FROG RUN      FROG RUN "
9020 X$=CHR$(96)+" NICREAL CALDWELL AUGUST 1983
9030 REPEAT
9040 PLOT1,2,Z$:PLOT1,26,X$
9050 C$=LEFT$(Z$,37):T$=RIGHT$(Z$,1)
9060 Z$=T$+C$
9070 T$=LEFT$(X$,37):Y$=RIGHT$(X$,1)
9080 X$=Y$+T$
9090 PLOTO,4,"HELP YOUR FROG ACROSS THE VERY BUSY"
9100 PLOTO,6,"ROAD YOU GET 100 POINTS EACH TIME YOU"
9110 PLOTO,8,"DO IT.BUT IT GETS HARDER AS WELL"
9120 PLOTO,12,"USE THE KEYS:"
9130 PLOTO,14,"Z - LEFT            X - RIGHT      "
9140 PLOTO,16,";  - UP             . - DOWN       "
9150 PLOT4,22,"PRESS ANY KEY TO START"
9160 PLOT2,22,12
9170 A$=KEY$
9180 UNTILA$<>""
9190 RETURN
```

How to get the most out of your Oric-1

Mass Delete Line Facility for Oric BASIC

By G. M. Phillips

Along with RENUMBER, the DELETE function is one of the major omissions in ORIC BASIC. It is often useful to be able to take out chunks of a large program, particularly when starting a new one.

For example, if you want to copy the tape handling subroutine from the ORIC BASE program, you could delete all lines except 20000 onwards.

The following machine-code subroutine gives you this facility, without too much fuss.

The program should be entered as below, and then RUN. (It would be advisable to save the BASIC program as well.)

After a few seconds, save the RAM area #400 to #4FF onto tape – this is the delete program.

Then, whenever you have a program which is to be partially deleted, CLOAD the machine-code program, and type in (as an immediate command):

DOKE0, line-from: DOKE2, line-no to: CALL#400

Do not worry if the message 'Syntax error' is printed. If the BASIC hangs up check that your BASIC program was typed correctly.

Example:
DOKE0, 200:DOKE2, 1000:CALL#400
would delete lines 200–1000.

Note that the program may work incorrectly if the line numbers supplied do not exist.

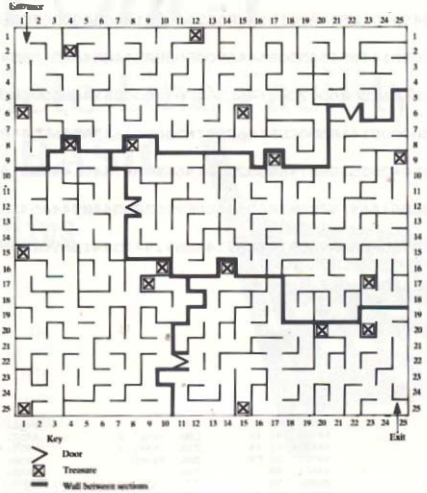
For Microtan owners, the routine has been dissassembled below, and should need little conversion. (Change LDA #5 to LDA #4 – use TANBUG to run)

```
20 REM RUN ONCE AND SAVE 0400-04FF
30 X=0400
40 READD$:IFD$="Z"THEN100
50 FORZ=1TOLEN(D$)/2:Z1=VAL("0"+MID$(D$,(Z-1)*2+1,2)):POKEX,Z1:X=X+
1:NEXT
60 GOTO 40
100 END
200 DATAAA905B545A901B544A001B144F032A002B104C500D007C8B144C501F013A
000B1
210 DATA4468550C8B1448545A550E544A1900F0D8A5448546A545B547A001B146F00
20001
220 DATA60A002B146C502D007C8B146C503F013A000B1468550C8B1468547A5508
546A9
230 DATA00F005A000B1468550C8B1068547A550654638A546E544854AA547E5458
54BA0
240 DATA01B146F042B553B8B1468552B8B146E54A91469144C8B146E54B9146914
4C8B1
250 DATAA9469144C8B1469144C8B1469144D0F9783B6544B544A545A690085A5A52B
546A5
260 DATA538547A900F0BBA9009144C8B144C8B144A5441B6902B59CB59EB54CB54
56900
270 DATAB59DB59FB5A160,Z
```

```
4400: A9 05     LDA #$05
4402: 85 45     STA $45
4404: A9 01     LDA #$01
4406: 85 46     STA $46
4408: A0 01     LDY #$01
440A: B1 44     LDA ($44),Y
440C: F0 32     BEQ $4440
440E: A0 02     LDY #$02
4410: B1 44     LDA ($44),Y
4412: C5 00     CMP $00
4414: D0 07     BNE $441D
4416: C8        INY
4417: B1 44     LDA ($44),Y
4419: C5 01     CMP $01
441B: F0 13     BEQ $4430
441D: A0 00     LDY #$00
441F: B1 44     LDA ($44),Y
4421: 85 50     STA $50
4423: C8        INY
4424: B1 44     LDA ($44),Y
4426: 85 45     STA $45
4428: A5 50     LDA $50
442A: 85 46     STA $46
442C: A9 00     LDA #$00
442E: F0 D0     BEQ $4400
4430: A5 44     LDA $44
4432: 85 46     STA $46
4434: A5 45     LDA $45
4436: 85 47     STA $47
4438: A0 01     LDY #$01
443A: B1 46     LDA ($46),Y
443C: F0 02     BEQ $4440
443E: D0 01     BNE $4441
4440: 60        RTS
4441: A0 02     LDY #$02
4443: B1 46     LDA ($46),Y
4445: C5 02     CMP $02
4447: D0 07     BNE $4450
4449: C8        INY
444A: B1 46     LDA ($46),Y
444C: C5 03     CMP $03
444E: F0 13     BEQ $4463
4450: A0 00     LDY #$00
4452: B1 46     LDA ($46),Y
4454: 85 50     STA $50
4456: C8        INY
4457: B1 46     LDA ($46),Y
4459: 85 47     STA $47
445B: A5 50     LDA $50
445D: 85 46     STA $46
445F: A9 00     LDA #$00
4461: F0 D5     BEQ $4438
4463: A0 00     LDY #$00
4465: B1 46     LDA ($46),Y
4467: 85 50     STA $50
4469: C8        INY
446A: B1 46     LDA ($46),Y
446C: 85 47     STA $47
446E: A5 50     LDA $50
4470: 85 46     STA $46
4472: 38        SEC
4473: A5 46     LDA $46
4475: E5 44     SBC $44
4477: 85 48     STA $48
4479: A5 47     LDA $47
447B: E5 45     SBC $45
447D: 85 49     STA $49
447F: A0 01     LDY #$01
4481: B1 46     LDA ($46),Y
4483: F0 42     BEQ $44C7
4485: 85 53     STA $53
4487: 88        DEY
4488: B1 46     LDA ($46),Y
448A: 85 52     STA $52
448C: 38        SEC
448D: B1 46     LDA ($46),Y
448F: E5 44     SBC $44
4491: 91 46     STA ($46),Y
4493: 91 44     STA ($44),Y
4495: C8        INY
4496: B1 46     LDA ($46),Y
4498: E5 48     SBC $48
449A: 91 46     STA ($46),Y
449C: 91 44     STA ($44),Y
449E: C8        INY
449F: B1 44     LDA ($46),Y
44A1: 91 44     STA ($44),Y
44A3: C8        INY
44A4: B1 44     LDA ($46),Y
44A6: 91 44     STA ($44),Y
44A8: C8        INY
44A9: B1 46     LDA ($46),Y
44AB: 91 44     STA ($44),Y
44AD: D0 F9     BNE $44A8
44AF: 98        TYA
44B0: 38        SEC
44B1: E5 44     ADC $44
44B3: 85 44     STA $44
44B5: A5 45     LDA $45
44B7: E9 00     ADC $0000
44B9: 85 45     STA $45
44BB: A5 52     LDA $52
44BD: 85 46     STA $46
44BF: A5 53     LDA $53
44C1: 85 47     STA $47
44C3: A9 00     LDA $0000
44C5: F0 B8     BEQ $447F
44C7: A9 00     LDA $0000
44C9: 91 44     STA ($44),Y
44CB: C8        INY
44CC: 91 44     STA ($44),Y
44CE: C8        INY
44CF: 91 44     STA ($44),Y
44D1: A5 44     LDA $44
44D3: 18        CLC
44D4: 69 02     ADC $0002
44D6: 85 9C     STA $9C
44D8: 85 9E     STA $9E
44DA: 85 A0     STA $A0
44DC: A5 45     LDA $45
44DE: 69 00     ADC $0000
44E0: 85 9D     STA $9D
44E2: 85 9F     STA $9F
44E4: 85 A1     STA $A1
44E6: 60        RTS
44E7: EA        NOP
44E8: EA        NOP
44E9: EA        NOP
44EA: EA        NOP
44EB: EA        NOP
44EC: EA        NOP
44ED: EA        NOP
44EE: EA        NOP
44EF: EA        NOP
44F0: EA        NOP
```

Power Maze of Agrenon

By D. Hunt

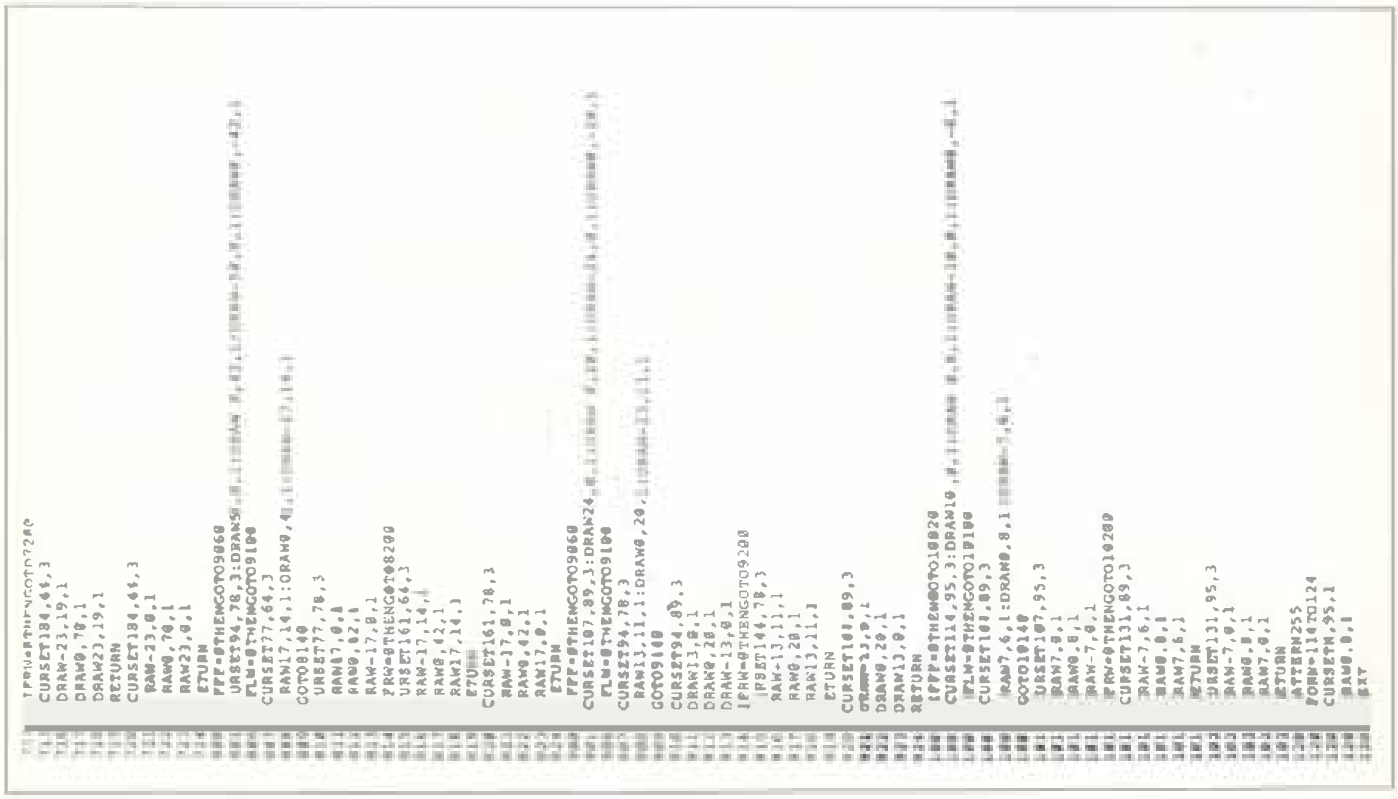


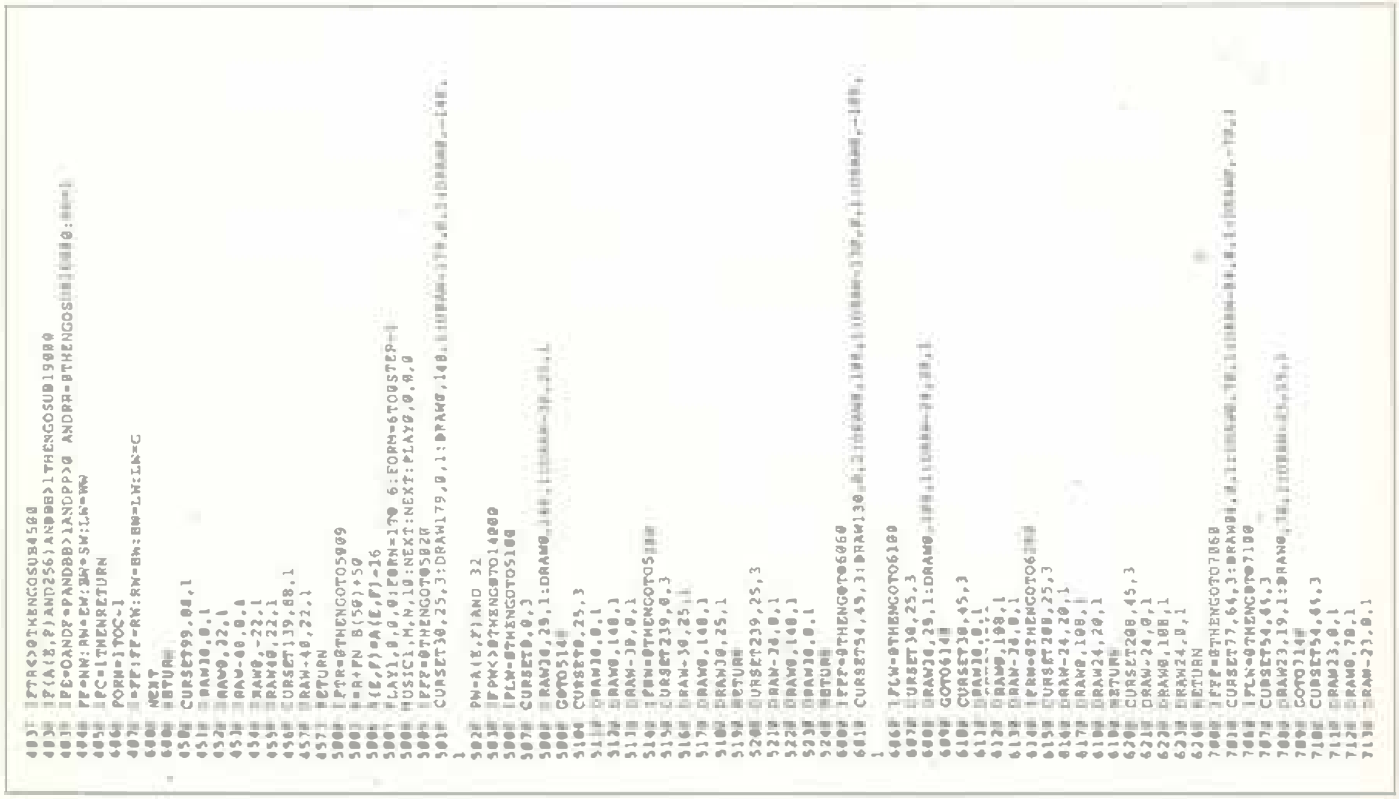
The game is played in a 25 by 25 preset maze, the game is in real time and the aim is to kill the 'vicious power matrix', the "GEOMITRIX" and to collect the "Diametric Tapestries" which are spread throughout the maze. The maze is divided into 3 sectors separated by doors, the exit is at location 25, 25.

The "GEOMITRIX" is destroyed by firing 'High power energy zones' at it which are spread throughout the maze. There is also a status check telling you where the "GEOMITRIX" is, how many 'High power energy zones' there are left and how much treasure you have. There are also ratings at the end.

Agrenon Listing

Agrenon Listing





Agrenon Listing

```
10 A(A,B)=A(A,B) OR 32
20 A=A+25 AND B<25
30 B=B+25 AND B<25
40 IF A<A,B> AND 16 THEN GOTO 14140
50 IF A<25 AND B<25 THEN GOTO 14140
60 C=C+ B(0)
70 C=C+ B(1000)-500
80 POP:GOSUB 3000:RETURN
90 PRINT
100 INK 3
110 POKE #26A,10
120 PRINT CHR$(4);
130 PRINT CHR$(27);"D";CHR$(27+5)          POWER MAZE OF AGRENON"
140 PRINT CHR$(4)
150 PRINT
160 PRINT CHR$(96);" LOPTON/HUNT 1983"
170 PRINT
180 PRINT"In your search for rare earth elements"
190 PRINT"you have landed on the planet Agrenon."
200 PRINT
210 PRINT"Where you have discovered the"
220 PRINT"remains of an ancient pan dimensional"
230 PRINT"power civilisation"
240 PRINT
250 PRINT"In searching one of the settlements"
260 PRINT"you have fallen through a power matrix"
270 PRINT"into the power maze of Agrenon"
280 PRINT:PRINT"This was a maze built by the ancient"
290 PRINT"race filled with diametric tapestries"
300 PRINT"and used to test their own kind"
310 PRINT
320 PRINT:PRINT:PRINT
330 PRINT CHR$(27);"L PRESS ANY KEY TO CONTINUE"
340 WAIT#DF,#80:IF BS="" THEN GOTO 15190
350 BS=""
360 CLS
370 INK 6
380 PRINT CHR$(4);
390 PRINT
400 CHR$(27);"B";CHR$(27);"J          POWER MAZE OF AGRENON"
410 PRINT
420 CHR$(4);
430 "within the maze there is a vicious"
440 "spatial vortex of great intelligence"
450 "known as the"
460 CHR$(27);"■";CHR$(27);"LGEOMITRIX";CHR$(27);"Q";CHR$(27
470 "
480 CHR$(2);"F";
490 "who when you"
500 "are caught will distort your spatial"
510 "sector and you will be quickly"
520 "over come by the deadly specs of the"
530 "planet"
540 PRINT
550 PRINT"there are spatial anomilies within the"
560 PRINT"maze which when entered will displace"
570 "you to a new spatial sector"
580 PRINT
590 "There are high power energy zones"
600 "throughout the power maze.These can"
610 "be used to destroy the GEONITRIX"
620 PRINT
630 "The only exit is at the far end of"
640 "the maze. This is reached through 3"
650 "doors"
660 PRINT CHR$(27);"L PRESS ANY KEY TO CONTINUE"
670 WAIT#DF,#80:IF BS="" THEN GOTO 15370
680 CLS
690 INK 5
700 BS=""
710 PRINT
720 PRINTCHR$(4);
730 CHR$(27);"F";CHR$(27);"J          CONTROL FUNCTIONS"
740 :PRINT
750 CHR$(6);
760 "F:-Fires high power energy zone which"
770 "are directed by the cursor keys"
780 PRINT
```

ORIC – Tabbing

[Decimal point aligned]
or Right-justified if
no decimal places.

```
Screen                    1                   2                   3
Column No   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  etc...upto40
'Col' No      1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8  9 0 1 2 3 4 5 6 7 8  etc...upto38

                  1 2 3 . 4 5       2 8 . 3 6       4 4 6 . 2 7
                      6 . 3 3     2 1 7 . 6 0     4 2 5 3 . 2 0
                    ITEM 'A'          ITEM 'B'          ITEM 'C' etc...
```

TWO BLANK LEFT MARGIN
COLUMNS ON THE SCREEN

```
 05  REM
 10  REM
 20  REM
 30  REM
 40  INPUT A,B,C
 50  PRINT
100  COL = 5

110  ITEM = A
120  GOSUB 500
200  COL= 15
210  ITEM = B
220  GOSUB 500
300  COL = 25
310  ITEM = C
320  GOSUB 500
330  PRINT

340  GOTO 40(or?)
500  COL=COL+2

510  PRINT
520  RETURN
```

```
••••••••••••••••••••••••••••••••••••••••••••••••
ROUTINE FOR TABBING – IT ALSO ALIGNS
AMOUNTS EITHER SIDE DECIMAL POINT
••••••••••••••••••••••••••••••••••••••••••••••••
:REM FOR TESTING PURPOSES – ENTER VALUES.

:REM COLUMN NO OF UNIT DIGIT OF FIRST ITEM
      IGNORING TWO BLANK COLS ON LEFT OF
      SCREEN

:REM COLUMN NO OF UNIT DIGIT OF SECOND ITEM

:REM COLUMN NO OF UNIT DIGIT OF THIRD ITEM

:REM SPACE-UP AFTER LAST ITEM REQUIRED ON
THELINE.
:REM CONTINUE YOUR PROGRAMME.
:REM ADD TWO COLUMN NO TO ALLOW FOR FIRST
2 MARGIN COLS.
SPC(COL–POS(×)–(LEN(STR$(INT(ITEM)))–1));ITEM;
```

[Left-justified] Alternative A.

```
                    1         2         3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
```

2 blank columns on left edge of screen — ONE BILL | TWO CHRIS | THREE HARRY | FOUR TOMetc

```
05 REM*****************************************
10 REM SUGGESTED ROUTINE FOR TABBING
20 REM LEFT-JUSTIFIED ITEMS
30 REM*****************************************
40 DATA 0, 10, 18, 26          :REM TAB COLUMN NUMBERS
50 FOR I = 1 TO 4
60 READ C(I)                   :REM READ-IN TAB COL NUMBERS
70 C(I) = C(I) + 12
80 NEXT I
90 PRINT
100 FOR I = 1 to 4
110 PRINT TAB(C(I));           :REM TAB TO APPROPRIATE COLUMN
120 PRINT CHR$[#11);           :REM BACK-SPACE CURSOR ONE LINE
130 INPUT X$                   :REM ENTER AN ITEM
140 NEXT I
150 GOTO 90
```

Note: BECAUSE THE ABOVE "TESTING" PROG USES INPUT X$ A QUESTION-MARK? WILL "PRINT" IN FRONT OF THE ITEM TYPED IN

[Left-justified] Alternative B.

THIS IS A STRAIGHT "IN-LINE" VERSION OF THE PROG ON PAGE 2. (ALTERNATIVE A)

```
10 CL$
20 PRINT: PRINT
30 PRINT TAB(12);
40 INPUT X$                    :REM FIRST ITEM
50
60 PRINT TAB(22);
70 PRINT CHR$(#11);
80 INPUT X$                    :REM SECOND ITEM
90
100 PRINT TAB(30);
110 PRINT CHR$[#II);
120 INPUT X$                   :REM THIRD ITEM
130
140 PRINT TAB(38);
150 PRINT CHR$(#11);
160 INPUT X$                   :REM FOURTH ITEM
170
180 GOTO 30
```

ORIC – Tabbing

(Left-justified) Alternative "C"

This is similar to the "DECIMAL POINT ALIGNMENT" Routine. But without taking into account the length of the field to be printed. It takes the difference between the position you wish to TAB to, and the present position of the CURSOR, and then SPC(spaces) that number of spaces. It will work for string variables, and IS AS too.

```
 40 INPUT A,B,C
 50 PRINT
100 COL = 0
110 ITEM = A
120 GOSUB 500
200 COL = 15
210 ITEM = B
220 GOSUB 500
300 COL = 26
310 ITEM = C
320 GOSUB 500
330 PRINT
340 GOTO 40                    (or GOTO wherever you want to Continue)

500 COL = COL + 2
510 PRINT                      SPC(COL - POS(0)); ITEM;
520 RETURN
```

(Page reorganised for convenience)

```
15452 PRINT"LEFT - turns left within the spatial"
15460 PRI  sector"
15470 PRINT"RIGHT - turns right within the spatial"
15480 PRINT"sector"
15490 PRINT
15500 PRINT"UP - moves forwards"
15510 PRINT
15520 PRINT"BACK AND - displays final information"
15530 PRINT"During this period time decreases at"
15540 PRINT"at a greater rate"
15550 PRINT
15560 PRINT"The plant base is displayed in three"
15570 PRINT"dimensions"
15580 PRINT:PRINT:PRINT:PRINT
15590 PRINT"THESE ARE NOT TO Confuse"
15600 A$=KEY$:IF A$="" THEN GOTO 15730
15740 CLS
15760 B$="":RETURN
16000 GOSUB16700:ON BB1GOTO16100,16200,16300,16400,16500,16600
16110 RETURN
16120 CURSET 144,120,3
16130 DRAW -67,14,1
16140 CURSET 77,66,3
16150 DRAW 67,14,1
16160 CURSET 141,64,1
16170 DRAW -67,14,1
16180 CURSET 138,127,1
16190 DRAW 0,-54,1
16195 RETURN
16200 CURSET 54,153,3
16210 DRAW 154,20,1
16220 CURSET 184,153,1
16230 DRAW -154,20,1
16240 CURSET 30,25,3
16250 DRAW 154,20,1
16260 CURSET 208,25,3
16270 DRAW -154,20,1
16280 CURSET 120,163,1
16290 DRAW 0,-126,1
16295 RETURN
16300 CURSET 77,134,3
16310 DRAW 107,19,1
16320 CURSET 161,134,1
16330 DRAW -107,19,1
16340 CURSET 54,45,3
16350 DRAW 107,19,1
16360 CURSET 184,45,3
16370 DRAW -107,19,1
16380 CURSET 120,144,1
16390 DRAW 0,-89,1
16395 RETURN
16400 CURSET 94,129,3
16410 DRAW 67,14,1
16420 CURSET 144,120,3
16430 DRAW -67,14,1
16440 CURSET 77,68,3
16450 DRAW 67,14,1
16460 CURSET 101,66,3
16470 DRAW -67,14,1
16480 CURSET 120,126,1
16490 DRAW 0,-93,1
16495 RETURN
16500 CURSET 107,109,3
16510 DRAW 37,11,1
16520 CURSET 131,109,3
16530 DRAW -37,11,1
16540 CURSET 94,78,3
16550 DRAW 37,11,1
16560 CURSET 144,78,3
16570 DRAW -37,11,1
16580 CURSET 120,115,3
16590 DRAW 0,-31,1
16595 RETURN
16600 CURSET 114,103,3
16610 DRAW 17,6,1
16620 CURSET 124,103,3
16630 DRAW -17,6,1
```

Agrenon Listing

Agrenon Listing

```
CURSET 107,89,3
DRAW 17,6,1
CURSET 131,89,3
DRAW -17,6,1
CURSET 119,104,3
DRAW 0,-11,1
RETURN
MUSIC 1,0,1,0
MUSIC 2,0,4,0
MUSIC 3,1,6,0
PLAY 7,0,4,200
H=1
RETURN
IF BB=1 THEN RETURN
SOUND 1,419,0
SOUND 2,419,0
SOUND 3,464,0
PLAY 7,0,4,1000
OO=1
IF BB=1 THEN Q=Q+1:A(E,F)=A(E,F)-64:GOSUB 17730
ON BB GOTO 17100,17200,17300,17400,17500,17600,17700
FOR N=183 TO 193
PATTERN 170
CURSET 80-(N-183),N,3
DRAW 80*2*(N-183),0,00
NEXT
PATTERN 255
RETURN
FOR N=160 TO 166
PATTERN 170
CURSET 85-(N-160),N,3
DRAW 70*2*(N-160),0,00
NEXT
PATTERN 255
RETURN
FOR N=140 TO 144
PATTERN 170
CURSET 90-(N-140),N,3
DRAW 60*2*(N-140),0,00
NEXT
PATTERN 255
RETURN
FOR N=126 TO 128
PATTERN 170
CURSET100-(N-126),N,3
DRAW 40*2*(N-126),0,00
NEXT
PATTERN 255
RETURN
FOR N=114 TO 116
PATTERN 170
CURSET 110-(N-114),N,3
DRAW 20*2*(N-114),0,00
NEXT
PATTERN 255
RETURN
FOR N=106 TO 107
PATTERN 170
CURSET 115-(N-106),N,3
DRAW 10*2*(N-106),0,00
NEXT
PATTERN 255
RETURN
FOR N=558 TO 449 STEP-1
PLAY 7,0,0,0
SOUND 1,N,10
SOUND 2,N+15,10
SOUND 3,N+21,10
NEXT
RETURN
IF BB=1 THEN RETURN
GOSUB 20000:ON BB-1 GOTO
CURSET 120,25,3
DRAW 0,148,1
CURSET 30,100,3
DRAW 170,0,1
PATTERN 85
RETURN
```

```
CURSET 120,45,3
DRAW 0,188,1
CURSET 54,100,3
DRAW 130,0,1
PATTERN 85
RETURN
CURSET 120,64,3
DRAW 0,78,1
CURSET 77,100,3
DRAW 54,0,1
PATTERN 85
RETURN
CURSET 120,78,3
DRAW 0,42,1
CURSET 94,100,3
DRAW 54,0,1
PATTERN 85
RETURN
CURSET 120,89,3
DRAW 0,20,1
CURSET 107,100,3
DRAW 24,0,1
PATTERN 85
RETURN
CURSET 120,95,3
DRAW 0,8,1
CURSET 95,100,3
DRAW 10,0,1
PATTERN 85
RETURN
SOUND 1,550,5
SOUND 2,1000,0
PLAY 3,5,4,1000
RETURN
IF FF=0 THEN RETURN
CC=SGN(A-64):DD=SGN(B-P)
IF(A(0,P)AND 128)<>0THEN A(0,P)=A(0,P)-128
IF(A(0,P)AND 128)<>0THEN EE=0:GOSUB 22000
IF CC=1 AND(A(0,P)AND2)=0 THEN E=0+1
IF DD=1 AND(A(0,P)AND4)=0 THEN P=P+1
IF CC=-1 AND(A(0,P)AND8)=0 THEN E=0-1
IF DD=-1 AND(A(0,P)AND1)=0 THEN P=P-1
A(0,P)=A(0,P) OR 128
RETURN
GG=GG+1:IF GG>2 THEN P=P+DD:E=0+CC:POKE[illegible]:RETURN
IF(A(0,P)AND1)=0 THEN GOTO 22010
IF(A(0,P)AND8)=0 THEN GOTO 22060
IF(A(0,P)AND2)=0 THEN GOTO 22010
IF(A(0,P)AND2)=0 THEN GOTO 22020
IF(A(0,P)AND1)=0THEN P=P-1:GOTO 22010
POP:RETURN
IF(A(0,P)AND2)=0THEN O=O+1:GOTO 22020
POP:RETURN
IF(A(0,P)AND4)=0THEN P=P+1:GOTO 22040
POP:RETURN
IF(A(0,P)AND8)=0THEN O=O-1:GOTO 22060
POP:RETURN
PATTERN 170
INK 1:PAPER 7
FOR N=1 TO 25
CURSET FN B(200)+20,FN B(1)*20,2
FOR M=1 TO 10
CIRCLE M,1
SOUND 1,N,0
SOUND 2,N+10,0
SOUND 3,N+20,0
PLAY 7,0,3,10
NEXT
NEXT
EXECODE
WAIT 100
SOUND 1,1000,0
PLAY 1,7,4,400
WAIT 700
PLAY 7,0,0,0
FOR N=1100 TO [illegible] STEP-10
SOUND 1,N,15
NEXT
```

Agrenon Listing


```
 IF QQ=1 THEN RETURN
 BS="YOU HAVE BEEN OVERCOME BY THE SPORES"
 GOTO 31200
 CLS
 GOTO 31200
 IF Q=0 THEN RETURN
 FOR N=6 TO 1 STEP-1:ZAP:WAIT NUMBER
SHOOT
CLS
 JJ=A:KK=B:LL=C:MM=0:NN=1:QQ=1
 BB=8:B=JJ:F=KK:C=LL:RR=1
 GOSUB 25910
E=JJ:F=KK
 A$=KEY$
 IF A$<>"" THEN GOTO25100
HH=1
 PLAY 7,0,0,0
 SS=15-(SQR((ABS(A-JJ))^2+(
 SOUND 1,449,SS
 SOUND 2,464,SS
 SOUND 3,475,SS
 T=T-1:IF T<0 THEN POP:GOTO 130
 PRINT CHR$(11);CHR$(11);
 PRINT" ";CHR$(27);"DTIME=";INT(T/100)+1;"TREASURE=_";R
 PRINT" ";CHR$(27);"ADJRECT THE HIGH POWER ENERGY ZONE"
 GOTO 25030
D=ASC(A$)
 IF D=11 THEN CLS:GOSUB 25200
 IF D=8 THEN CLS:GOSUB 25300
 IF D=9 THEN CLS:GOSUB 25300
 IF D=32 THEN E=0:RR=0:RETURN
 GOTO 25010
 FOR N=1 TO BB
 IFJJ=B(N,1)ANDKK=
 NEXT
E=JJ:F=KK
 GOSUB 4000
 IF LL=3 AND SW<>0 THEN EXPLODE:POP:RR=0:RETURN
 IF LL=4 AND SW<>0 THEN EXPLODE:POP:RR=0:RETURN
 IF LL=3 AND SW<>0 THEN EXPLODE:POP:RR=0:RETURN
 IF LL=4 AND SW<>0 THEN EXPLODE:POP:RR=0:RETURN
 IF LL=1 THEN KK=KK-1
 IF LL=2 THEN JJ=JJ-1
 IF LL=3 THEN KK=KK+1
 IF LL=4 THEN JJ=JJ-1
 IF JJ=0 AND KK=P THEN GOSUB29000
 FOR N=1 TO BB
 IF B(N,1)=JJ AND B(N,2)=KK THEN
 NEXT
 RETURN
 GOSUB 3600
 GOSUB 3500:GOSUB 4000:B(BB,1)=E:B(BB,2)=F
 GOTO 25900
 LL=LL-1:IF LL<1 THEN LL=4
 NN=0
 RETURN
 LL=LL+1:IF LL>4 THEN LL=1
 NN=0
 RETURN
 PRINT CHR$(7);:FOR O=0 TO 4
 FOR N=6 TO 0 STEP-1
 FOR M=1 TO 12
 MUSIC 1,N,M,15
 NEXT
 NEXT
 NEXT
 FOR N=1 TO 2700 STEP 10
 SOUND 1,N,15
 NEXT
 WAIT 100
 PLAY 0,0,0,0
 EXPLODE
 IF QQ=1 THEN RETURN
 R=R+FN B(50)+950
 A(O,P)=A(O,P)-128
 POP:POP
 PP=PP-1:IF PP>0 THEN
 NR=0
 RETURN
```

```
 CLS
 POKE #26A,10
 PRINT
 PRINT CHR$(27);CHR$(4
 AGRENON"
 PRINT CHR$(4)
 PRINT
 PRINT"Do you want a short demonstration of"
 PRINT"the power mazes features?"
 PRINT
 PRINT CHR$(27);"LTYPE 'YES' OR 'NO'";:INPUT B$
 IF LEFT$(B$,1)<>"Y" THEN RETURN
 QQ=1
 CLS
 HIRES
 PRINT"A spatial anomaly is like this"
 PRINT"It will move you to a new spatial"
 PRINT"sector";
 WAIT 100
 GOTO 14000
 WAIT 500
 HIRES
 CLS
 PRINT CHR$(4)
 PRINT"A high power energy zone looks like"
 PRINT"this. These are used to kill the"
 PRINT"GEOMITRIX";
 BB=2:GOSUB17000
 WAIT 800
 HIRES
 CLS
 PRINT"The GEOMITRIX looks like this"
 BB=2:GOSUB 16000
 WAIT 100
 PLAY 0,0,0,0
 CLS
 HIRES
 PRINT"A door connecting sections looks like"
 PRINT"this"
 LW=1:GOSUB 5000
 LW=0:GOSUB 6000
 BB=3:GOSUB 19000
 LW=1:RW=1:GOSUB 7000
 GOSUB 8000
 LW=0:RW=1:FF=1:GOSUB 9000
 WAIT 800
 PLAY 0,0,0,0
 HIRES:PATTERN 255:CLS
 PRINT"You can see up to 6 levels"
 PRINT"anything beyond this will be
 LW=0:RW=0:FF=0:GOSUB 5000
 LW=1:RW=1:GOSUB 6000
 LW=0:GOSUB 7000
 RW=0:LW=0:GOSUB 8000
 LW=1:RW=1:GOSUB 9000
 RW=0:GOSUB 10000
 GOSUB 12000
 WAIT 700
 CLS
 GOSUB 16300:GOSUB 16700
 PRINT"This is what the GEOMITRIX looks like"
 PRINT"in the maze"
 QQ=0
 WAIT 800
 PLAY 0,0,0,0
 TEXT
 PRINT
 PRINT CHR$(4);
 PRINT CHR$(27);"C";CHR$(27);"J      POWER MAZE OF AGRENON"
 PRINT:PRINT
 PRINTCHR$(4);
 PRINT"Do you want to see the instructions"
 PRINT"again";:INPUT B$
 IF LEFT$(B$,1)<"Y" THEN POP:CLS:GOTO 15
 CLS
 RETURN
 HIRES
 FOR N=500 TO
 PLAY 7,0,0,0
```

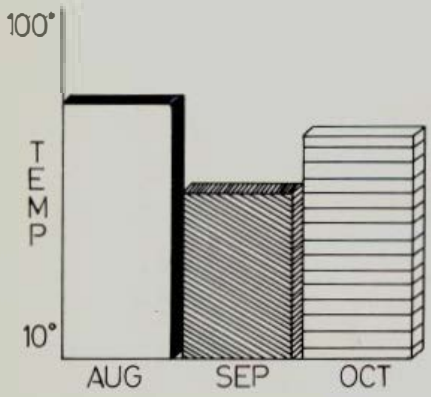
Agrenon Listing

Software

```
30810 SOUND 1,0,10
30820 SOUND 2,N+20,10
30830 SOUND 3,N+40,10
30840 NEXT
30850 WAIT 200
30860 CURSET 128,128,3
30880 FOR N=95 TO 1 STEP-4
30890 CIRCLE N,1
30900 SOUND 1,N,10
30910 NEXT
30920 PLAY 0,0,0,0
30925 EXPLODE
30930 PP=0:LW=0:MM=0:GOSUB 6000:GOSUB 5000:GOSUB 6000:GOSUB 7000:GO
SUB 6000
30940 GOSUB 9000:GOSUB 10000
30950 EXPLODE:WAIT 500:CURSE 31200
31000 FOR N=1 TO 199 STEP 5
31001 PLAY 1,0,0,0
31010 CURSET 1,N,3
31015 SOUND 1,N,10
31020 DRAW 230,0,1
31030 NEXT
31040 FOR N=1 TO 230 STEP 5
31050 CURSET N,1,3
31055 SOUND 1,230-N,10
31060 DRAW 0,199,2
31070 NEXT
31072 PLAY 7,0,0,0
31075 CURSET 128,128,3
31080 FOR N=1 TO 99 STEP 3
31081 SOUND 1,400+N,10
31083 SOUND 2,420+N,10
31085 SOUND 3,440+N,10
31090 CIRCLE N,2
31100 NEXT
31110 WAIT 100
31120 FOR N=1 TO 199 STEP 5
31082 CURSET 128,100,3
31140 DRAW -128,N-100,2
31150 CURSET 128,100,3
31160 DRAW 128,N-100,2
31170 SOUND 1,N,10
31180 NEXT
31190 PLAY 0,0,0,0
31195 GOSUB 32000
31200 TEXT:CLS
31205 FOR N=47864 TO 47871:POKE N,63:NEXT
31210 PRINT
31220 PRINT CHR$(27);CHR$(6);"2          POWER MAZE OF AGRENON"
31230 PRINT:PRINT
31240 PRINT CHR$(8);
31250 PRINT CHR$(27);"G";CHR$(27);"????????????????????????????????????
????"
31260 PRINT
31270 PRINT CHR$(27);"G";CHR$(27);"????????????????????????????????????
??????"
31272 PRINT CHR$(8);
31275 PRINT:PRINT:PRINT CHR$(27);"J                YOUR RATING"
31276 PRINT CHR$(8);
31277 WAIT 200
31280 VV=((N+UU*10)+Q*10)/200
31290 FORN=1TO5:PRINTCHR$(11);:;:NEXT
31299 PRINT CHR$(27);"a";CHR$(27);"1";
31310 FOR N=1 TO VV
31320 PRINT " ";
31325 WAIT 100
31330 NEXT
31340 FOR N=1 TO 16:PRINTCHR$(10);:NEXT
31342 PRINT CHR$(13)
31345 VV=INT(VV*100)
31350 IF VV<0 THEN PRINT"LOW LEVEL LIFE FORM"
31360 IF VV>0 AND VV<200 THEN PRINT"NOT OF AGRENON DESCENT"
31370 IF VV>199AND VV<500 THEN PRINT"LARGELY BELOW AVERAGE AGRENON
CITIZEN"
31380 IF VV>499 AND VV<1000 THEN PRINT"SLIGHTLY BELOW AVERAGE AGREN
ON CITIZEN"
31390 IF VV>959 AND VV<1000 THEN PRINT"AVERAGE AGRENON CITIZEN"
31395 IF VV>2999 THEN PRINT"YOU HAVE ACHIEVED
31400 IFVV>2999ANDVV<4888THENPRINT"AGRENON TECHNICAL POWER ASSISTAN
T";PRINT"STANDARD"
31430 IF VV>3999 AND VV<5000 THEN PRINT"AGRENON SPATIAL ENGINEER ST
ANDARD"
31430 IF VV>4999 AND VV<6000 THEN PRINT"AGRENON POWER BATTLE MASTER
 STANDARD"
31430 IF VV>5999 THEN PRINT"BATTLE LORD OF AGRENON STANDARD"
31440 WAIT 500
31450 PRINT
31460 IF Q=1 THEN B$=" " ELSE B$="S"
31470 PRINT"YOU HAD ";O;"HIGH POWER ENERGY ZONE";B$;CHR$(11)
31475 PRINT"LEFT
31480 PRINT"YOU COLLECTED _";R;"WORTH OF TREASURE"
31490 IF PP=1 THEN B$="LIFE" ELSE B$="LIVES"
31495 IF PP=0 THEN PRINT"THE GEOMITRIX IS DEAD":GOTO 31510
31500 PRINT"THE GEOMITRIX HAD ";PP;" ";B$;" LEFT"
31510 PRINT"YOU VISITED ";UU;"SPATIAL SECTORS          OUT OF 625"
31515 PRINT
31520 INPUT"DO YOU WANT TO REENTER THE MAZE";B$
31530 IF LEFT$(B$,1)="Y" THEN RUN
31540 CLS
31565 CLEAR
31570 END
32000 MIKES
32002 PLAY 1,0,0,0
32005 CURSET 21,0,3
32010 FOR N=198 TO 2 STEP-2
32020 DRAW N,0,1
32030 DRAW 0,N,1
32040 DRAW -N,0,1
32050 DRAW 0,-N,1
32060 SOUND 1,N,10
32070 NEXT
32100 CURSET 219,0,3
32110 FOR N=198 TO 2 STEP-2
32120 DRAW -N,0,2
32130 DRAW 0,N,3
32140 DRAW N,0,2
32150 DRAW 0,-N,3
32160 SOUND 1,198-N,10
32170 NEXT
32180 PLAY 0,0,0,0
32185 EXPLODE
32190 RETURN
```

3D Bar Chart

By M. J. Hall



This program is designed to give a graphical result in the form of a 3D bar chart which can have a maximum of 12 items. It gives the user the choice of how many items there are provided that there are no more than 12. Also in the program for a title where if the user wants a scale then he/she must put the scale in there and remember to make sure that the largest actual number is less than or equal to 1500. The program also asks the user to enter labels for the x and y axis's, and labels for each bar which can only be 2 characters in length (max) and so it could be made better by having the labels on the bars, which I haven't done because it was meant for the initials of the months etc. Also this program works out a scale for the y axis and at the end asks the user if he/she wishes to display any more information. If the user does then the program starts again otherwise it ends.

```
10 PAPER0:INK1:TEXT:CLS
20 DIM M$(12),DTA(12),I$(12)
30 GOSUB5000
40 HIRES:INK1
50 FOR P=48367 TO 48127 STEP40:POKEP,2:NEXTP
60 FOR P=41007 TO 41247 STEP40:POKEP,4:NEXTP
70 FOR P=48567 TO 49127 STEP40:POKEP,7:NEXTP
80 INPUT"Please enter the title(in less than 30 letters)";
   A$:CLS
90 IF LEN(A$)>30 THEN PRINT"That is too long,try
   again":GOTO 80
100 INPUT"What do you wish to label the x axis with
   (<=than 30 letters)";X$:CLS
110 IF LEN(X$)>30 THEN PRINT"That is too long,try
   again":GOTO 100
120 INPUT"What do you wish to lable the y axis with
   (<=22 letters)";Y$:CLS
130 IF LEN(Y$)>22 THEN PRINT"That is too long,try
   again":GOTO 120
140 FOR C=1TO N
150 PRINT"Enter 1-2 initials for item";C;" on the x axis
   please":INPUTI$(C)
160 IF LEN(I$(C))>2 THEN GOTO 150
170 NEXTC
180 CURSET41,0,3:DRAW0,180,1:DRAW196,0,1
190 X=48
200 FOR C=1TO N
210 CURSETX,182,3
220 FOR A=1 TO LEN(I$(C))
230 CHAR ASC(MID$(I$(C),A,1)),0,1
240 CURMOV6,0,3
250 NEXTA
260 IF DTA(C)=0 THEN 340
270 CURSETX,180,3
280 DRAW0,-DTA(C) X SC,1:DRAW8,0,1
290 DRAW0,DTA(C) X SC,1
300 CURSETX,180-(DTA(C) X SC),3
310 DRAW4,-4,1:DRAW8,0,1:DRAW-4,4,1
320 CURSETX+12,180-(DTA(C) X SC)-4,1
330 DRAW0,DTA(C) X SC,1:DRAW-4,4,1
340 X=X+16
350 NEXTC
360 CURSET50,1,3
370 FOR S=1 TO LEN(A$)
380 CHAR ASC(MID$(A$,S,1)),0,1
390 CURMOV6,0,3
400 NEXTS
410 CURSET50,190,3
420 FORS=1TO LEN(X$)
430 CHAR ASC(MID$(X$,S,1)),0,1
440 CURMOV6,0,3
450 NEXTS
460 CURSET1,20,3
470 FOR S=1 TO LEN(Y$)
480 CHARASC(MID$(Y$,S,1)),0,1
490 CURMOV0,8,3
500 NEXTS
510 YY%=180:C=0
520 REPEAT
530 C=C+1
540 L%=L%+VAL(KK$)-WW+O
550 C$(C)=STR$(L%)
560 YY%=YY%-VAL(KK$) XSC
```

3D Bar Chart Listing

```
570 O=WW
580 UNTIL YY%<10
590 YY%=100:C=0
600 REPEAT
610 C=C+1
620 CURSET14,YY%,3
630 FOR S=2. TO LEN(C$(C))
640 CHAR ASC(MID$(C$(C),S.1))0,1
650 CURMOV6,0,3
660 NEXTS
670 YY%=YY%—VAL(KK$) X SC
680 UNTIL YY%<10
690 YY%=100
700 REPEAT
710 CURSET36,YY%,3:DRAW4,0,1
720 YY%=YY%—VAL(KK$) X SC
730 UNTIL YY%<10
740 YY%=100
750 REPEAT
760 YY%=YY%—VAL(KK$) X (SC XM)
770 CURSET38,YY%,3:DRAW2,0,1
780 UNTIL YY%<10
790 CLS
800 INPUT"DOyou wish to display anymore          information (Y/N)";O$
810 IF LEFTS(Q$,1)="Y" OR LEFTS(Q$.1)="N" THEN 820 E1SE
    CLS:GOTO 800
820 IF Q$="Y" THEN RUN
830 TEXT
840 PLOT8,10,"«XXXXXXXXXXXXXXXXXXXXXXXXX"
850 PLOT8,11,"X                          X"
860 PLOT8,12,"X                          X"
870 PLOT8,13,"X                          X"
880 PLOT8,14,"X                          X"
890 PLOT8,15,"XXXXXXXXXXXXXXXXXXXXXXXXX"
900 PLAY,0,0,0
910 FOR C=1 TO 6
920 SOUND1,500,10
930 PLOT11,12,"THIS PROGRAM HAS"
940 PLDT15,13,"NOW ENDED"
950 WAIT20
960 SOUND1,300,10:WAIT20
970 SOUND1,400,10
980 PLOT11,12,"                "
990 PLOT15,13,"                "
1000 WAIT20:SOUND1,600,10
1010 NEXT C
1020 PLAY0,0,0,0 POKE#26A,10:CLS:PRINT CHR$(19):INK0:END
2000 REM INITIALIZATION
2010 FOR C=1 TON
2020 READ M$(C)
2030 NEXTC
2040 RETURN
2050 DATA FIRST, SECOND, THIRD, FOURTH, FIFTH, SIXTH,
    SEVENTH, EIGHTH, NINTH, TENTH
2060 DATA ELEVENTH, TWELTH
5000 REM X START OF PROGRAM X
5010 PRINT"Please enter the number of items              (MAX is 12)";
5020 INPUT'N
5030 IFN<1   ORN>12   ORN<>INT(N)1HENPRINT"INCORRECT
    ENTRY,TRY AGAIN":GOTO 5020
5040 GOSUB 2000
5050 FORC=1TON
5060 PRINT"PLEASE ENTER";M$(C);" ITEM OF DATA";
5070 INPUT DTA(C)
5080 IF MA<DTA(C) THEN MA=DTA(C)
5090 IF DTA(C)<0THEN PRINT"INCORRECT ENTRY,TRY AGAIN":
    GOTO 5060
5100 I F  MA>1500  THEN,  PRINT"OVERFLOW  ERROR,  START
    AGAIN", WAIT400:RUN
5110 NEXTC
5120 IF MA<=100THEN SC=1.5:KK$="20":WW=20:M=.33:RETURN
5130 IF MA<=200 THEN SC=.8:KK$="40":WW=40:M=.25:RETURN
5140 IF MA<=300THEN SC=.55:KK$="60":WW=60:M=.33:RETURN
5150 IF MA<=400THEN SC=.4:KK$="80":WW=80:M=.25:RETURN
5160 IFMA>=401 THEN SC=.1:KK$="300":WW=300:M=.33:RETURN
5170 RETURN
```

This is a Hires program that draws the outline of the British Isles.

It also draws on the screen a few randomly chosen cities throughout England, Scotland and Wales.

This program allows for others to embellish or change it as desired.

```
10 REM ********************************
20 REM *  MAP OF ENGLAND  C.KINGHAM   *
30 REM ********************************
40 RESTORE
50 CLS
60 HIRES
70 PAPER 7:INK 0
80 PRINT CHR$(17)
90 WAIT 100
100 CURSET 60,20,3
110 L$="THE BRITISH ISLES"
120 FOR L=1 TO LEN(L$)
130 CHAR ASC(MID$(L$,L,1)),0,1
140 CURMOV 6,0,0:NEXT L:WAIT 100
150 CURSET 119,40,3
160 REPEAT
170 READ A,B
180 WAIT 10
190 DRAW A,B,1
200 UNTIL A=-.1
210 DATA 1,7,-9,5,3,2,-5,4,5,1,6,-4,12,5,-6,14,-7,3,5,0,-6,5,1,2,6,-2,8,7
220 DATA 4,16,5,2,11,13,-1,2,-8,-3,8,7,-2,7,2,1,2,-5,9,1,1,9,-12,16,1,2,1,-2
230 DATA 3,2,3,-2,3,2,-8,8,-2,-2,-4,4,-6,-3,-11,1,-11,3,-4,-3,-5,3,1,3,-7,1
240 DATA -3,-2,-7,4,-1,2,-3,1,-1,-2,-3,1,-1,-3,6,-2,1,-3,4,-1,3,-7,5,-2,9,-1
250 DATA 5,-7,-9,5,-6,-4,-2,2,-1,-2,-3,-2,-7,4,-2,-4,2,-2,13,-7,1,-10,-5,1
260 DATA 0,-3,5,-5,10,-1,1,-5,-2,-2,3,-4,-4,1,-4,-9,4,-6,-7,3,-3,1,-1,2
270 DATA -5,-2,-3,1,-1,-4,5,-7,-1,-7,-1,-5,3,-1,-2,-3,-4,3,-2,-2,-1,-4,-2,0
280 DATA -3,11,-3,2,2,-8,2,-9,5,-1,1,-2,-3,1,-4,-3,4,-1,-1,-3,3,-1,1,-5
290 DATA -3,-1,1,-4,6,-2,-1,-4,4,1,1,-3,5,-2,3,2,8,-3,3,4,-.1,0
300 WAIT 50
310 CURSET 104,168,3
320 CIRCLE 1.5,1
330 CURSET 60,165,3
340 A$="Plymouth"
350 FOR A=1 TO LEN(A$)
360 CHAR ASC(MID$(A$,A,1)),0,1
370 CURMOV 5,0,0:NEXT A
380 WAIT 50
390 CURDSET 123,155,3
400 CIRCLE 1.5,1
410 CURSET 85,153,3
420 B$="Bristol"
430 FOR B=1 TO LEN(B$)
440 CHAR ASC(MID$(B$,B,1)),0,1
450 CURMOV 5,0,0:NEXT B
460 WAIT 50
470 CURSET 115,150,3
480 CIRCLE 1.5,1
490 CURSET 77,143,3
500 C$="Cardiff"
510 FOR C=1 TO LEN(C$)
520 CHAR ASC(MID$(C$,C,1)),0,1
530 CURMOV 5,0,0:NEXT C
540 WAIT 50
550 CURSET 129,135,3
560 CIRCLE 1.5,1
570 CURSET 75,132,3
580 D$="Birmingham"
590 FOR D=1 TO LEN(D$)
600 CHAR ASC(MID$(D$,D,1)),0,1
610 CURMOV 5,0,0:NEXT D
620 WAIT 50
630 CURSET 125,119,3
640 CIRCLE 1.5,1
650 CURSET 71,116,3
660 E$="Manchester"
670 FOR E=1 TO LEN(E$)
680 CHAR ASC(MID$(E$,E,1)),0,1
690 CURMOV 5,0,0:NEXT E
700 WAIT 50
710 CURSET 103,82,3
720 CIRCLE 1.5,1
730 CURSET 65,79,3
740 F$="Glasgow"
750 FOR F=1 TO LEN(F$)
760 CHAR ASC(MID$(F$,F,1)),0,1
770 CURMOV 5,0,0:NEXT F
780 WAIT 50
790 CURSET 130,95,3
800 CIRCLE 1.5,1
810 CURSET 134,91,3
820 G$="Newcastle"
830 FOR G=1 TO LEN(G$)
840 CHAR ASC(MD$(G$,G,1)),0,1
850 CURMOV 5,0,0:NEXT G
860 WAIT 50
870 CURSET 144,116,3
880 CIRCLE 1.5,1
890 CURSET 148,111,3
900 H$="Hull"
910 FOR H=1 TO LEN(H$)
920 CHAR ASC(MID$(H$,H,1)),0,1
930 CURMOV 5,0,0:NEXT H
940 WAIT 50
950 CURSET 159,135,3
960 CIRCLE 1.5,1
970 CURSET 163,132,3
980 I$="Norwich"
990 FOR I=1 TO LEN(I$)
1000 CHAR ASC(MID$(I$,I,1)),0,1
1010 CURMOV 5,0,0:NEXT I
1020 WAIT 50
1030 CURSET 132,163,3
1040 CIRCLE 1.5,1
1050 CURSET 132,168,3
1060 J$="Portsmouth"
1070 FOR J=1 TO LEN(J$)
1080 CHAR ASC(MID$(J$,J,1)),0,1
1090 CURMOV 5,0,0:NEXT J
1100 WAIT 50
1110 CURSET 146,154,3
1120 CIRCLE 2.0,1
1130 CURSET 150,151,3
1140 K$="London"
1150 FOR K=1.TO LEN(K$)
1160 CHAR ASC(MID$(K$,K,1)),0,1
1170 CURMOV 5,0,0:NEXT K
1180 GOTO 1180
```

Word – 1

By David Peat

"Word-1" is a word-processor in basic for the Oric-1.

It allows you to enter up to 10 screens of text (this can be changed by changing every number 10 in the program to the number of screens you want) to be entered and edited. The keyboard works normally except for the cursor keys.

EDITING

Pressing either the left or right cursor keys gives EDIT mode. In this, choose which line of text you wish to edit then move the arrow under the letter you wish to change using the cursor keys and press "RETURN" then enter the letter you wish to put in its place.

"DEL" deletes the last character you entered in the line and escape "ESC" gives you the menu. Pressing either of the up or down keys will move you onto a new screen and "RETURN" gives carriage-return.

MENU

You are shown the menu:

1. LIST SCREEN N
2. LIST SCREEN X UNTIL LINE Y
3. LIST SCREEN X TO Y
4. PRINT SCREEN N
5. PRINT SCREEN X UNTIL LINE Y
6. PRINT SCREEN X TO Y

After choosing you are asked:

Do you want:

1. a continuous listing or
2. a page by page listing.

Option 1 lists the screens chosen non-stop while option 2 pauses in between and waits for a key to be pressed.

Back to the menu. . .

1. causes screen N to be listed. You enter N.
2. causes screen N to be listed to line Y.
3. causes screens X to Y to be printed on the screen. Best used with option 2 above.

Options 4, 5 and 6 do exactly the same but print on a printer other than the screen.

```
2 CLS:CO$U$10000
5 CR$=PRINT CHR$(30);:C=#$(17)
6 PRINTCH#$(6)
10 01$A$(10,26):PO=48041
50 T$="WORD-PROCESSOR":FORF=1TO14:POKE48013+F,ASC(MID$(T$,F,1)):NEXT
70
90 FOR=1TO10:PLOT10,10,"SCREEN "+STR$(F):CLS:NEXT:CLS
95 PA=48041
96 TAB=18
100 POKE-01026:PO=A$041+(A0*L):FOR(^-#T036
110 POKEPO ,46: PX=PEEK(PO-1)
120 GET#$:IFO<>PX1,100,0:PLAY1,0,1,150
130 IF#3=CHR$(13)THEN#GOTO 2000
140 IF#5=CHR$(8)OR K$=CHR$(9) THEN#GOSUB1800:<<OUVD1,A0,0:PLAY1,0,7,2
00:GOTO120
145 IF#5=CHR$(10)ORK$=CHR$(11)THEN440
150 IFK$=CHR$(17)TH-NCOSUB4000
160 IF#5=CHR$(127)THENPOTO2500
165 IFASC(LS)%32THEN120
400 POKEPC,A$(PE):PO=PO+1:POKEPO,43:PX=PEEK(PO+1)
410 A$(P,L)=A$(P,L)+K$
420 WEITC:GOUB$0,80,0:PLAY1,0,1,200
425 PO=PO+1:PX=PEEK(PO)
430 WEXT1
440 PINC:WEITH
450 CA$U$6000
460 GOTO240
1000 CLS:PRINT"WHICH TIME.0-24":INPUTTEL:IFEL%260RTEL%0GRINT(EL)%%EI
THEN1000
1010 CLS
1020 PLOT0,5,A$(9,EL)
1030 5=0:PLOT5,6,"L"
1040 K$=KEY$:IFK$=CHR$(8)AND$%OTHENPLOT5,6," ":5=5-1
1045 PLOT5,6,"L"
1050 IFK$=CHR$(9)AND%3BTHENPLOT5,6," ":5=5+1
1060 IFK$=CHR$(13)THEN1100
1070GOTO1040
1100 PLOT5,6,"L"
1110 PLOT0,5," ":PLOT0,10,"Which character.":GETF$:PLOT5,5,F$
1115 IF$=0:THEN$3=A$(9,L):A$(9,EL)="":GOTO1227
```

Word – 1 Listing

```
2511 AS(N,L)=AS(N,L)+RIGHTS(BS,LEN(BS)-C-1)
6000 F=0
6005 SE="E"
6010 CLS
6020 PRINT:PRINT"1. LIST SCREEN N"
6030 PRINT"2. LIST SCREEN N UNTIL LINE Z"
6040 PRINT"3. LIST SCREEN X TO Y"
6060 PRINT"4. PRINT SCREEN N"
6070 PRINT"5. PRINT SCREEN N UNTIL LINE Z"
6080 PRINT"6. PRINT SCREEN X TO Y"
6100 PLOT0,14,"SELECT PROCESSING CHOICE"
6110 GET0
6120 IFD=LOAD THEN
6123 CLS
6125 IFD=3THENP=1
6130 CLS
6135 IFD=3THENP=0-1
6140 PR=620
6150 GOTOPR
6300 IFP=0THENPLOT0,2,"1. LIST SCREEN N"
6305 IFP=1THENPLOT0,2,"4. PRINT SCREEN N"
6310 PRINT:PRINT:PRINT:PRINT:INPUT"Enter N";N
6340 IFP=LORX=100RX=INT(X)THENCLS:GOTO6300
6350 I=26
6360 GOTO8000
6400 IFP=0THENPLOT0,2,"2. LIST SCREEN N UNTIL Z"
6405 IFP=0THENPLOT0,2,"5. PRINT SCREEN N UNTIL Z"
6410 PLOT0,4,"Enter N":GETX
6417 IFX=LORX=100RX=INT(X)THEN6410
6425 PLOT0,4,"Enter Z":GETZ
6440 IFZ=LORZ=INT(Z)ORZ=26THENZ=25
6450 GOTO8000
6500 IFP=0THENPLOT0,2,"3. LIST SCREEN X TO Y"
6505 IFP=1THENPLOT0,2,"7. PRINT SCREEN X TO Y"
6510 PLOT0,4,"Enter X":GETX
6515 IFX=LORX=THENGOTO6510
6520 PLOT0,4,"Enter Y":GETZ:Y=25
6525 IFZ=100RZ=XTHEN6520
6540 GOTO8000
8000 CLS
8005 IFP=1THEN8100
8010 PLOT0,2,"Do you want:"
8020 PLOT0,4,"1. A continuous listing or"
8030 IFP=0THENPLOT0,5,"2. A page by page listing."
8040 PLOT0,9,"ENTER CHOICE":GETQ:IFQ<>1ANDQ<>2THEN8040
8070 GOTO8200
8200 FORS=XTOZ
8220 CLS
8230 IFQ=2ANDP=0THENPLOT9,10,"SCREEN"+STR$(S)
8240 IFQ=2ANDP=0THENWAIT100
8241 CLS
8250 IFP=1ANDSS="y"THENLPRINT:LPRINT"----------
--------"
8270 IFP=1ANDSS="y"THENLPRINT
8275 POKE=0707
8280 PLOT0,Y,AS(X,7)
8310 IFP=0THEN8350
8312 LPRINTAS(X,7)
8350 NEXT
8360 IFP=1ANDSS="y"THENLPRINT:LPRINT"----------
--------"
8370 IFP=1ANDSS="y"THENLPRINT
8380 IFP="y"THENREPEAT:UNTILKEY$<>""
8381 NEXTT,S
8390 RETURN
10000 T$="WORD-PROCESSOR":FORN=1TO14:FORA=80TO:W,ASC(MID$(T$,N,1))
NEXT
10010 PRINT:PRINT"Use this program to create up to 10 "
10020 PRINT"screens of text."
10030 PRINT"  Editing:-"
10040 PRINT" Pressing 'DEL' will delete the last  character."
10050 PRINT" Pressing any one of the left and "
10060 PRINT"right cursor keys puts you into edit  mode."
10070 PRINT"Enter the line you want to edit(0-26)"
10080 PRINT"and then move the arrow under the  letter you want r
emoved and"
10090 PRINT"press 'RETURN'."
10100 PRINT"  Menu:"
10110 PRINT"Pressing 'ESC' gives you the menu."
10120 PRINT"Press 'RETURN' for carriage return."
10130 PRINT:PRINT"Press a key.":REPEAT:UNTILKEY$<>"":L$,KEY$:N
```

'Derby Day'

By M. J. Hall

This program is a horse racing game as you can tell from the title, where you have to enter the number of players between 2 and 8, and then the odds are printed up for the eight horses. The odds range from 1 to 8. one being the lowest and eight the highest due to the fact that if you win a race, the amount of money that you bet is multiplied by the odds, so eight will give you eight times as much and one only one times the amount. After that each player will have to enter the horse that he/she wishes to bet on and the amount that he/she wishes to bet on that horse. This must be done in turn and then that order kept until the end of that game. Once everyone has placed their bet, a table of the number of horse, bet and odds is printed up so everyone can see what they have bet and the odds and number of that horse. The program then proceeds and goes through the graphics and finally produces a winning horse. Then the program prints up any winners and any bankrupt people. It continues doing this until there is either one player left with something to bet with or until no one has any money left. At this point the program gives the users the choice of another game or not, where if they don't then the program goes to the ending routine and ends. or else it starts again.

```
  2 GOSUB 9000
  1 PRINT"
120 PRINT"
131 DIM M(9),B(8),H(8),P$(8,A$,W)
135 DIM W(9),B(8),H(8),P$(8),A$,W)
140 FOR X=1 TO 8:M(X)=0:NEXT
200 ML(1)+ML(2)+ML(3)+ML(4)+ML(5)+ML(6)+ML(7)+ML(8)
205 REM ** MONEY CHECKING ROUTINE **
210 LET F=1+ML(1)+ML(2)+ML(3)+ML(4)+ML(5)+ML(6)+ML(7)+ML(8)
215 IF F=1 THEN 5000
225 REM ******************
230 PRINT:PRINT:PRINT:PRINT:PRINT
235 PRINT:PRINT"IMPORTANT,YOU MUST REMEMBER THE HORSE THAT YOU ARE GOING TO";
245 PRINT" BET ON,AS THE   ODDS ARE ONLY PRINTED UP ONCE"
    PRINT:PRINT:PRINT:PRINT:PRINT
    PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
    GET A$:IF A$=CHR$(32) THEN 310 ELSE 260
320 LET T=0
330 PRINT"No.OF
    PRINT"-----
340 FOR C=1 TO 8
350 LET O(C)=INT(8 * RND(1))+2
360
370 PRINT
380 PRINT TAB(18);T;TAB(22);"1:";O(C);TAB(28);I$ T)
390 NEXTC
395 PRINT:PRINT
400 PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
405 GET A$:IF A$=CHR$(32) THEN 410 ELSE 405
410 CLS
420 PRINT" HORSE NUMBERS MUST BE <=8 AND >=1"
425 PRINT" ------
430 PRINT
440 FOR X=1 TO N
445 IF ML(X)<=0 THEN LET B(X)=0:GOTO 510
450 PRINT"PLAYER No.";X;" HAS ";M(X);" TO BET WITH"
460 IN PUT
465 IN PUT"ENTER THE HORSE NUMBER PLEASE"
```

```
470 REPEAT:GET K$ A):B X)=VAL(K$ X)):UNTIL H(A)>=1 AND H(A)<=8
490
500 IF B(X)<0 OR B(X) > ML(X) THEN GOSUB 6000:PRINT:PRINT:GOTO 490
510 PRINT
520 NEXTX
530 PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
535 GET A$:IF A$=CHR$(32) THEN 540 ELSE 535
540 CLS:T=0
545 PRINT:PRINT
550 PRINT"I----------
551 PRINT"I No.OF HORSE      BET       ODDS       I
552 PRINT"I----------
560
565 IF ML(X)=0 THEN GOTO 590
570 LET G=H(X)
575 PLOT1,7+T,126:PLOT38,7+T,124
580 PLOT7,7+T,STR$(H(X)):PLOT20,7+T,STR$(B(X)):PLOT29,7+T,STR$(O(G))
582 POKE 48328+(T * 40),0:POKE 48341+(T * 40),0:POKE 48350+(T * 40),0
585 T=T+1
590
595 PRINT"I----------I"
600
610 GOSUB5150
950 PRINT"********************************"
960 PRINT"*                              *"
970 PRINT"*  THE WINNING HORSE WAS ";W;"  *"
980 PRINT"*                              *"
990 PRINT"********************************"
1000 PRINT:PRINT
1002 PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
1005 GET A$:IF A$=CHR$(32) THEN 1010 ELSE 1005
1010 CLS
1020 FOR A=1 TO N
1030 IF H(X)=W THEN GOSUB 4000 ELSE LET ML(X)=ML(X)-B(X)
1040 IF ML(X)<=0 AND MM(X)=1 THEN MM(X)=0
1050 IF ML(X)<=0 THEN PRINT"PLAYER No.";X;" IS BANKRUPT":ML(X)=0:B(X)=0:H(X)=0
1060
1065
1070
4000
4001
4002
4010
4070 LET A=A+O(G) * B(X)
4080 PRINT" AMOUNT WON=";A:"BY PLAYER ";X;"
4090 PRINT A
4095
```

(Page reorganised for convenience)

Derby-day Listing


```
4070 PRINT"**********************************"
4080 LET ML(X)=AW(X)+ML(X)
4090 IF ML(X)<0 THEN LET ML(X)=0
4095 PRINT"PLAYER No.";X;" HAS ";ML(X);" LEFT TO BET"
4100 PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
4105 GET A$:IF A$=CHR$(32) THEN 4110 ELSE 4105
4110 RETURN
5000 REM ****************
5020 PRINT"Do you want another game (Y/N)"
5025 GET A$
5030 IF A$="Y" THEN GOTO10
5035 IF A$="N" THEN 5040 ELSE 5025
5040 LORES0
5050 FOR Q=1 TO 10
5060 PLOT8,10,"THIS PROGRAM HAS NOW ENDED"
5070 WAIT50
5080 MUSIC1,3,3,15
5090 PLAY1,0,2,50
5110 PLOT8,10,"                    "
5120 WAIT20
5125 PLAY0,0,0,0
5130 NEXTQ
5135 CLS:INK1:PRINT CHR$(19)
5140 END
5150 REM ** GRAPHICS ROUTINE **
5160 CLS:PAPER0:INK7
5170 POKE49040,17:POKE49080,17:POKE49000,17
5180 POKE48962,48:POKE49002,124:POKE49042,124
5190 PLOT10,2,"THEY'RE UNDER STARTERS ORDERS"
5195 WAIT100
5200 PLOT10,2,"                              ":PLOT15,2,"AND THEY'RE OFF"
5210 SHOOT:WAIT75
5220 PLOT15,2,"                "
5280 SOUND3,1,10
5290 PLAY4,1,3,200
5300 FOR S=1 TO 5
5310 WAIT50
5320 D=INT(7 * RND(1))+1
5330 IF D=1 THEN 5320
5340 FOR X=49033 TO 49008 STEP-5
5360 POKEX-1,D:POKEX-40,D:POKEX+38,D
5370 POKEX,127:POKEX+1,127:POKEX+2,127
5380 POKEX+39,33:POKEX+43,123
5390 POKEX+3,113:POKEX-41,122:POKEX-42,120
5395 POKEX-39,126:POKEX-40,127
5400 WAIT20
5410 POKEX,32:POKEX+1,32:POKEX+2,32
5420 POKEX+39,32:POKEX+43,32
5430 POKEX+3,32:POKEX-41,32:POKEX-42,32
5435 POKEX-39,32:POKEX-40,32
5440 POKEX,127:POKEX+1,127:POKEX+2,127
5450 POKEX+40,123:POKEX+42,33
5460 POKEX+3,113:POKEX-41,122:POKEX-42,120
5465 POKEX-39,126:POKEX-40,127
5470 WAIT20
5480 POKEX,32:POKEX+1,32:POKEX+2,32
5490 POKEX+40,32:POKEX+42,32
5500 POKEX+3,32:POKEX-41,32:POKEX-42,32
5505 POKEX-39,32:POKEX-40,32
5510 NEXTX
5520 G=INT(3 * RND(1))+1:IF G=1 THEN H$="LENGTH"ELSE H$="LENGTHS"
5530 Z=INT(8 * RND(1))+1:IF S=5 THEN 5620
5540 PLOT3,2,"                "
5550 PLOT2,2,2
5560 IF S=4 THEN W=INT(8 * RND(1))+1:GOTO 5590
5570 PLOT3,2,"NUMBER ":PLOT9,2,STR$(Z):PLOT12,2,"IS LEADING BY"
5575 PLOT25,2,STR$(G)
5580 PLOT29,2,H$:GOTO 5620
5590 PLOT10,2,"THEY'RE COMING UP TO THE":PLOT15,3,"LAST STRAIGHT"
5595 PING:WAIT50:PING:WAIT50:PING:WAIT20:SOUND3,1,10:PLAY4,1,3,200
5600 PLOT8,4,"NUMBER":PLOT15,4,STR$(W):PLOT18,4,"IS CATCHING UP":WAIT75
5610 PLOT10,2,"                    ":PLOT15,3,"                "
5615 PLOT8,4,"                        "
5620 NEXTS
5630 WAIT200
5640 INK0:PAPER1
5650 CLS
5660 RETURN
6000 REM ** INCORRECT ENTRY ROUTINE **
6010 CLS
6020 PLOT8,10,"[illegible]"
6030 PLOT8,11,"*                    *"
6040 PLOT8,12,"*   INCORRECT ENTRY  *"
6050 PLOT8,13,"*                    *"
6060 PLOT8,14,"[illegible]"
6070 FOR [illegible]
6080 MUSIC1,2,3,15
6090 PLAY1,2,3,15
6100 WAIT30
6110 PLAY0,0,0,0
6120 WAIT20
6130 NEXTC
6140 CLS
6150 RETURN
7000 REM ** INITIALIZATION **
7010 PAPER1:INK0
7020 POKE526a,10          <-- £ sign is meant to be a hash (#)
7030 CLS
7040 I$(1)="PINKFLY":I$(2)="RED RUM":I$(3)="SILVER CREAM":I$(4)="SPARTACUS"
7050 I$(5)="BLACK PRINT":I$(6)="BLACK BESS":I$(7)="KING CHARLES V"
7060 I$(8)="BESCHERS BROOK"
7070 RETURN
8000 REM ********************
8010 REM ** CHARACTER DEFINITION **
8020 REM ********************
8030 A=46000
8040 FOR N=0 TO 7:READX:POKE A+N,X:[illegible]
8050 [illegible]
8060 FOR N=0 TO 7:READX:POKE(A+264+N),X:NEXTN
8070 DATA 7,7,7,56,56,56,56,56
8080 FOR N=0 TO 7:READX:POKE(A+960+N),X:NEXTN
8090 DATA 15,15,15,15,15,0,0,0
8100 [illegible]
8110 DATA 48,48,48,48,48,15,15,15
8120 FOR N=0 TO 7:READX:POKE(A+984+N),X:NEXTN
8130 DATA 56,56,56,7,7,7,7,7
8140 FOR N=0 TO 7:READX:POKE(A+904+N),X:NEXTN
8150 DATA 63,63,0,0,0,0,0,0
8160 FOR N=0 TO 7:READX:POKE(A+848+N),X:NEXTN
8170 DATA 30,62,30,12,30,62,63,31
8180 [illegible]
8200 REM ** INSTRUCTIONS **
[illegible]
8230 [illegible]
8235 [illegible]
8236 [illegible]
8240 [illegible]
8250 [illegible]    BY MARTIN J.KALE"
8260 PRINT:PRINT
8270 [illegible]    The object of the game"
8280 [illegible]    is to use your skill and"
8290 [illegible]    luck to bet on horses,and"
8300 [illegible]    be the winner of the game"
8310 [illegible]    by being the last person"
8320 [illegible]    left there last a last"
8330 [illegible]    player in the game,then"
8340 [illegible]    there must be a winner so"
8350 PRINT"    youhave to make sure that"
8360 PRINT"    you bet carefully if you"
8370 PRINT"    want to win.During each game"
8380 PRINT"    the odds are only printed"
8390 [illegible]    up once,as the use is will"
8400 [illegible]    have to make sure  that"
8410 PRINT"    everyone is ready to"
8420 PRINT"    continue.
8430 PRINT:PRINT
8440 PRINT CHR$(27);"T   "+CHR$(27);"L   PRESS SPACE BAR TO CONTINUE"
8450 GET A$:IF A$=CHR$(32) THEN 8460 ELSE 8450
8460 CLS
[illegible]
10000 REM ** SUBTIBETO PRINT UP **
10010 REM ** WINNER OF THE GAME **
[illegible]
10030 PRINT:PRINT
10040 FOR X=1 TO 8
10050 IF ML(X)<=0 THEN 10060
10055 PRINT"PLAYER ";X;" IS THE WINNER OF THE GAME"
10056 PRINT:PRINT"WITH ";ML(X);" LEFT":PRINT
10060 NEXTX
10070 PRINT"PRESS SPACE BAR TO CONTINUE":PRINT
10080 GET A$:IF A$=CHR$(32) THEN 9000 ELSE 10080
```

Oric Demonstration Program

By Karl Williams

Lines 150–190 can be deleted to omit the Random Circles.

```
10 PAPER4:INK7
20 HIRES:INK0:FORX=1TO40
30 FORA=1TO6
40 B=INT(RND(1)*235)
50 C=INT(RND(1)*168)
60 CURSETB,C,3
70 D=RND(1)*6+17
80 E=INT(RND(1)*30)+1
90 FILLE,1,D
100 F=INT(RND(1)*8)
110 G=INT(RND(1)*12)+1
120 H=INT(RND(1)*5)+10
130 MUSIC1,F,G,H
140 NEXTA
150 I=INT(RND(1)*160)+40
160 J=INT(RND(1)*120)+40
170 CURSETI,J,3
180 K=INT(RND(1)*30)+9
190 CIRCLEK,1
200 NEXTX
```

120 PAGE

A doctor and scientist's thoughts on the Oric-1

Having decided to buy the Oric-1 48K RAM I can only hope it will gain popularity so that a multitude of add-ons, programs etc. will be available. I therefore forward these ponderings in the hope they may be useful to you, being well aware that you probably are flooded with "bright ideas".

First of all, why did I buy your machine? I heard of a colleague who bought a ZX-81 with manuals and other books in order to learn basic programming. He worked his way through this and then gave the device to his son. As he remarked, the cost for doing so was small compared to what you usually pay for medical courses (A weeks course in London amounts to 350 pounds + travel expenses). Thus home computers provide a fairly cheap and comfortable way to learn programming, even when you have access to bigger computers at the job.

When deciding for the Orie I was helped by a friend, who had worked with computers since the sixties and who is now a mathematician and a doctor. The arguments for one who have access to quite a deal of hardware, and even heavy computers, runs thus in favour of Orie:

1. The keyboard looks like that of ordinary computers/terminals. The Spectrum looks more like Italian salad than like a computer. I would have willingly paid another 50 pounds for a real hardtop keyboard, but found the Commodore 64 too expensive, when considering my ready access to computer-machinery at the hospital. Give it a thought. Others sell home-computers with plug-in keyboard!
2. The Orie has ports making it suitable for someone like me, who has printers, monitors etc. available at the hospital.
3. According to my adviser the 6502 processor is well suited for Pascal, a language he recommends.
4. I am interested in music and thus may have fun with the synthesizer. But this argument was of lesser importance. Except maybe Chess games are not of interest. . . unless they have relation to medicine/research as a sort of self teaching thing. I am quite allergic to ping and pang.

If I should draw your attention to a potential market consisting of doctors, scientists, students and other serious users, you should then be aware that they will demand different things than the present buyers. If you want to reach them, you will have to take into consideration that these people are used to studying. After an initial course they will need more heavy books (with smaller letters) going into the depths of problems and handbooks, where you can read selected chapters, when solving an actual problem.

Doctors need statistical tests first of all. For instance a multivariate analysis, which I have seen advertised for Sinclair! Give them a statistical pack, this should be no problem. Many doctors though being skilled surgeons have difficulty with statistics when analysing results. But even a program leading them to choose the right test is a possibility, using questions like: big or small material? Comparison of two means? Comparison of a mean with known standard? etc.

Many medical researchers already have programmable calculators like the Texas TI-59 performing statistics. What you

can offer is more text on the screen when guiding them to the right calculations and a graphical representation of the results too.

At present home-computers are not used by doctors/scientists to any significant degree. I think a good deal of the reason is that they do not know about the capabilities of home-computers. They have, however, bought the Texas TI-59 and HP analogues . . . this with their own money, because they are too interested in the results of their research, so that a little calculating at home is unavoidable. They will buy the Oric-1 too, if they get to know of its capabilities. Doctors do not want to be ignorant, so you can even scare them with questions like: Hospitals are invaded by computers. Do you know enough about how they work?

Should you choose to write the necessary books to make the Oric suitable for medical researchers or even the general practitioner, be aware of the tax-systems in many countries. In Scandinavia (e.g. my case 85% tax of the last penny) it may be all important to sell the Oric along with tapes and books as a package with a title like "A course in computing for the medical and biological sciences". This could make the whole thing tax deductible, whercas a home-computer would be hard to justify.

Dr. Tom Buur(actually Danish),
Snickaregatan 4 C,
582 26 Linkoping, Sweden

Dear Sir,
In issue 1 of 'Oric Owner' magazine you had a program listing of 'ORIC-TREK'. In the list of instructions you mentioned that you would need a 48K Oric to run the program.

Whether or not you already realise, it is possible to run the program on a 16K ORIC, using the 'GRAB' command and with some changes in the listing you can obtain better graphics with about 4000 bytes left over.

I find it very annoying that most of the adverts for the Oric Software do not specify which model of ORIC it can be run on. I am sure that other ORIC users have the same problem.

Perhaps in future issues if a 48K program listing is printed it is possible for tips to be addcd for the conversion to the 16K model in appropriate cases. *D. T. Paton*

Dear Ed.,
Does anyone out there know how to scroll the Orics screen left or right? I know how to use strings for this purpose but it is a very slow and tedious method. Can anyone help?
I would also like to know if anyone knows anything about the commands INVERSE and NORMAL lurking inside the ROM.

Also could you explain the phrase DOWNLOADING as it is a bit of a mystery to me. Thank you for a great magazine. Keep it up.

Finally, a query about the Oric Trek program in the first issue.

Whenever I try to move up or north or into a quadrant above me the program refuses to go there.

David Peat

Clock Simulation

By C. Kingham

This is a Hires program demonstrating an anologue clock display.

It allows the user to set a time and ORIC will then display the correct time in Hr, Min and Secs.

(Only varies about a second in two hours.)

To improve the accuracy of the clock. Minor adjustments can be made to the step interval at line 580 according to whether the clock is gaining or losing time.

```
40 PAPER 0
50 HIRES
60 PAPER 0:INK 3
70 M=0:H=0:F=0
110 CIRCLE 98,1
120 CIRCLE 78,1
130 CURSET 28,98,1:DRAW 103,0,1
140 DRAW 0,165,1:DRAW -183,0,1
170 A$="2345678 9"
200 CURSET 66+COS(F)*120,66+SIN(F)*100,3
220 C$="012 "
230 B$="||||"
240 FOR B=1 TO LEN(B$)
260 CURMOV 5,0,3
310 PRINT @5,17,"TIME (Hr,Min,Sec)";
320 INPUT H,M
330 IF M>60 OR M<0 OR H>12 OR H<0 THEN 340 ELSE 360
340 CLS:PRINT @5,10,"INVALID TIME"
360 M=M+1
370 CURSET 123,103,1
410 CURSET 123,103,1
450 REPEAT
460 CURSET 123,103,1
520 CURSET 123,103,1
540 CURSET 123,103,1
570 CURSET 123,103,1
610 CURSET 123,103,1
630 CURSET 123,103,1
640 DRAW 56*COS(Y),56*SIN(Y),0
650 M=M+1
660 IF M=60 THEN M=0:H=H+1
670 X=.1*(M=0 OR M=30)*(M=0)+.8-1.38
680 Y=.1*(M=15 OR M=45)*.55
690 Z=.1*(M=12) +.04
700 GOTO 470
1030 RETURN
1110 CURSET 24,14,3:DRAW 191,0,1
1120 DRAW 0,173,1
1130 CURSET 24,14,3:DRAW 4,4,1
1150 CURSET 24,186,3:DRAW 4,-4,1
1170 RETURN
```

Sine-patterns

By Carsten Skjerk

Sine-patterns should run on the 16K Oric without any changes. Line 30 sets the seed of the random number generator by looking at the interval timer, so it should be fairly random.

```
10 REM SINE-PATTERNS BY CARSTEN SKJERK
20 INK7:PAPER0
30 X=RND(-DEEK(630))
40 DEFFNA(X)=INT(RND(1)*X)
50 B=FNA(45)+1
60 HIRES
70 P=FNA(8):I=FNA(8)
80 IFI=PTHEN70
90 INKI:PAPERP
100 POKE618,10
110 Y=FNA(10)+11
120 FORX=-PITOPISTEP.2
130 CURSETX*Y+120,SIN(X)*20+99,3
140 CIRCLER,1
150 NEXTX
160 WAIT500
170 GOTO50
```

Interference

By J Reberg

```
10 HIRES
20 FORI=0TO239STEP2
30 CURSET0,199,3
40 DRAWI,-199,2
50 NEXTI
```

Figures

By J Reberg

```
10 HIRES
20 A=1.2
30 K=0.1
40 X1=120
50 X2=100
60 INK0
70 PAPER7
100 FORI=KTO1000STEPK
110 X=SIN(I)*100+120
120 Y=SIN(I*A)*90+100
130 CURSETX,Y,1
140 IFABS(X-X1)+ABS(Y-Y1)<1THEN160
150 DRAWX1-X+0.5,Y1-Y+0.5,1
160 X1=X
170 Y1=Y
180 NEXTI
```

Oric's Sound Capabilities

By S. Haigh

```
10 READ A,B,C: IFC=-1 THEN END
20 MUSIC 1,A,B,0: PLAY 1,0,1,1110*C
30 WAIT 14*C: GOTO 10
50 DATA 3,10,2,4,1,4,4,3,2,4,5,3,4,7,1,4,5,2,4,3,4,3,12,2,3,8,4,3,10,1
60 DATA 3,12,1,4,1,4,3,10,2,3,10,3,3,9,1,3,10,2,3,12,4,3,9,2,3,5,4,3,10,2
70 DATA 4,1,4,4,3,2,4,5,3,4,7,1,4,5,2,4,3,4,3,12,2,3,8,4,3,10,1,3,12,1
80 DATA 4,1,3,3,12,1,3,10,2,3,9,2,3,7,2,3,9,2,3,10,4,3,10,2,3,10,6,0,0,-1
```

The program above is an example of the Oric's sound capabilities which will play an easily recognised tune. All music can be converted as below with only a minimum knowledge of reading sheet music.

```
10 PAPER$:INK0
20 POKE#26A,10
30 CLS
35 PLOT7,10,"PLEASE WAIT A MINUTE,WHILE"
40 PLOT12,11,"YOU ARE BEING ARMED"
45 GOSUB 420:CLS:PRINT:PRINT
46 PRINT:PRINT" DO YOU REQUIRE INSTRUCTIONS (Y/N)"
47 REPEAT:GET A$:UNTIL A$="Y" OR A$="N":IF A$="Y" THEN GOSUB 1470
48 PRINT:PRINT
49 PRINT"ENTER THE LEVEL PLEASE 1(EASY) 2(HARD)"
50 REPEAT:GET LEVEL$:UNTIL LEVEL$="1" OR LEVEL$="2"
55 IF LEVEL$="1" THEN LEVEL=1
60 IF LEVEL$="2" THEN LEVEL=2
65 CLS:FOR C=1 TO 4
70 ZAP:PLOT10,10,"ACTION STATIONS":WAIT50
75 PLOT10,10,"                    ":WAIT30:NEXT:WAIT100
80 PLOT7,15,"THE ENEMY ARE COMING":WAIT150
85 GOSUB 1200
90 GOSUB 1320
100 X=15
110 PLOTX,Y,O$
120 REPEAT
125 IF C<1000 THEN POKE48060,32
130 C=Q-1:IF C<100 THEN POKE48059,32:IF C<10 THEN POKE48058,32
140 O$=GTR$(O):PLOT15,0,O$
150 SC$=STR$(S):PLOT20,0,SC$
160 IF D1=1 AND A<=5 THEN PLOT A,B,R$:GOSUB 9000
165 IF D1=2 AND A>=30 THEN PLOT A,B,R$:GOSUB 9000

170 IF R=3 OR R=6 THEN Q=2 ELSE Q=1
180 PLOTA,B,R$
190 IF D1=1 THEN A=A-Q
195 IF D1=2 THEN A=A+Q
200 PLOTA,B,R$(R):WAIT2
210 K$=KEY$:IF K$="" THEN K$=L$
220 IF K$=CHR$(8) THEN PLOTX,Y,T$:X=X-2:L$=K$
230 IF K$=CHR$(9) THEN PLOTX,Y,T$:X=X+2:L$=K$
240 IF K$=CHR$(10) AND LEVEL=1 THEN CO=X:L$=K$
250 IF X<=1 THEN X=1
260 IF X>=38 THEN X=38
270 PLOTX,Y,O$
280 IF K$=CHR$(13) THEN GOSUB 710
290 UNTIL C<=0
300 WAIT100
310 POKE48057,32:POKE48056,32
320 CLS
330 PRINT:PRINT
340 PRINT" YOUR SCORE WAS ";S
350  IF  S>W THEN PRINT" THE HIGHEST SO FAR"ELSE PRINT" THE  BEST
WAS ";W;" BY ";N$
360 IF W=0 THEN W=S:INPUT"ENTER YOUR NAME PLEASE";N$
370 PRINT:PRINT
375 IF LEVEL=1 AND S>20000 THEN PRINT"YOU OUGHT TO BE ON LEVEL 2"
376 PRINT:PRINT
380 PRINT"DO YOU WANT ANOTHER GAME (Y/N)"
```

(Page reorganised for convenience)

Torpedo Attack

By Martyn J. Hall

This program uses LORES graphics and has user defined characters. The program gives the user the choice of seeing the instructions, and then gives the user the choice of two levels. The difference between the two is that on level one the speed of the torpedo is faster than the speed of the torpedo on level two. Once the actual game has started the user has to destroy as many ships in the time given which can be increased on every 300 points scored. There are three different types of ships each of different points. When the time runs out your score is printed up and if you have the highest score so far then you will be asked to enter your name. Then you will be asked if you want another game or not, if you do then the game starts again otherwise it will end.

Torpedo Attack Listing

```
390 GETA$:IF A$="Y" THEN 46
400 IF A$="N" THEN 900 ELSE 380
420 REM ** CHARACTER DEFINITION **
430 T=46080
440 FOR N=0 TO 7:READX:POKE(T+264+N),X:NEXTN
450 DATA #0C,#0C,#1E,#1E,#3F,#3F,#3F,#3F
460 FOR N=0 TO 7:READX:POKE(T+984+N),X:NEXTN
470 DATA #0,#0,#1,#3F,#1F,#F,#07,#03
480 FOR N=0 TO 7:READX:POKE(T+1000+N),X:NEXTN
490 DATA #10,#20,#0,#0,#3F,#3F,#3F,#3F
500 FOR N=0 TO 7:READX:POKE(T+728+N),X:NEXTN
510 DATA #0,#0,#0,#0,#0,#0,#3F,#3F,#3F
520 FOR N=0 TO 7:READX:POKE(T+512+N),X:NEXTN
530 DATA #0,#0,#0,#0,#0,#1F,#0F,#07,#03
540 FOR N=0 TO 7:READX:POKE(T+976+N),X:NEXTN
550 DATA #0,#0,#0,#0,#3F,#3F,#3F,#3F
560 FOR N=0 TO 7:READX:POKE(T+944+N),X:NEXTN
570 DATA #30,#30,#30,#30,#3F,#3F,#3F,#3F
580 FOR N=0 TO 7:READX:POKE(T+920+N),X:NEXTN
590 DATA #0,#0,#0,#0,#0,#3F,#3F,#3E
600 FOR N=0 TO 7:READX:POKE(T+848+N),X:NEXTN
610 DATA #0,#0,#0,#0,#3F,#1F,#0F,#07,#03
620 FOR N=0 TO 7:READX:POKE(T+792+N),X:NEXTN
630 DATA #0,#0,#0,#0,#3F,#3F,#3F,#3F,#3F
640 FOR N=0 TO 7:READX:POKE(T+968+N),X:NEXTN
650 DATA #20,#38,#38,#3F,#3F,#3F,#3F,#3F
660 FOR N=0 TO 7:READX:POKE(T+960+N),X:NEXTN
670 DATA #0,#0,#0,#3F,#3E,#3C,#38,#30
680 FOR N=0 TO 7:READX:POKE(T+768+N),X:NEXTN
690 DATA #3F,#3F,#3F,#0,#0,#0,#0,#0
691 FOR N=0 TO 7:READX:POKE(T+776+N),X:NEXTN
692 DATA 0,0,0,0,63,63,63,31
693 FOR N=0 TO 7:READX:POKE(T+800+N),X:NEXTN
694 DATA 3,3,3,3,63,63,63,63
695 FOR N=0 TO 7:READX:POKE(T+808+N),X:NEXTN
696 DATA 0,0,0,0,62,60,56,48
697 FOR N=0 TO 7:READX:POKE(T+952+N),X:NEXTN
698 DATA 1,7,7,63,63,63,63,63
699 FOR N=0 TO 7:READX:POKE(T+912+N),X:NEXTN
700 DATA 0,0,32,63,62,60,56,48
701 FOR N=0 TO 7:READX:POKE(T+936+N),X:NEXTN
702 DATA 2,1,0,0,63,63,63,63
709 RETURN
710 REM ** SHOOTING ROUTINE **
720 O$=STR$(O)
730 IF LEVEL=1 THEN SP=2.75 ELSE SP=1.5
740 FOR C=22 TO 5 STEP-SP
750 PLOTA,B,R$
760 IF DI=1 THEN A=A-.4*Q
770 IF DI=1 AND A<=5 THEN PLOTA,B,R$:GOSUB 9000
780 IF DI=2 THEN A=A+.4*Q
790 IF DI=2 AND A>=30 THEN PLOTA,B,R$:GOSUB 9000
800 IF R=1 OR R=2 OR R=3 THEN DI=1 ELSE DI=2
810 IF R=3 OR R=6 THEN Q=2 ELSE Q=1
820 PLOTA,B,D$(R):WAIT2
```

```
[illegible]
[illegible]
NEXTC
GOSUB 3500
 IF  S=Y1  OR  S=Y2  OR S=Y3 OR S=Y4 OR  S=Y5  OR  S=Y6  THEN
300:GOSUB 1440
IF O<1000 THEN POKE48060,32
O=O-9:IF O<100 THEN POKE48059,32:IF O<10 THEN POKE48058,32
IF O<=0 THEN PLOT15,0,"0":POKE48057,32:POKE48056,32
PLOTX,C+SP,T$
PLOTX,C+SP+1,96
PLOT15,0,O$
RETURN
REM ********************************
SC$=STR$(S)
PLOT28,0,SC$
IF R=1 OR R=4 THEN S=S+20
IF R=2 OR R=5 THEN S=S+10
IF R=3 OR R=6 THEN S=S+50
PLOT28,0,SC$:PLOTX,C+SP+1,96:X=X+1:PLOTX-1,Y,T$:PLOT15,0,O$
GOSUB 9000
[illegible]
980 REM ** ENDING ROUTINE **
[illegible]
1000 PLOT8,8,"**********"
1010 PLOT8,9,"*                    *"
1020 PLOT8,10,"*                    *"
1030 PLOT8,11,"*                    *"
[illegible]
[illegible]
1070 PLOT11,10,"THIS PROGRAM WAS"
1080 PLOT15,11,"RON ENDED"
1090 PLAY2,0,2,200:MUSIC2,3,2,15:WAIT50
1100 PLOT11,10,"          "
1110 PLOT15,11,"          "
1120 PLAY0,0,0,0:WAIT40
[illegible]
[illegible]
1180 CLS:INK6:PRINT CHR$(19)
[illegible]
1200 REM ** INITIALIZATION **
1210 B=5:Y=25:L=126:S=0:O=400
1215 R=INT(6*RND(1))+1
1216 IF R=1 OR R=2 OR R=3 THEN A=30 ELSE A=5
[illegible]
1230 Q$="  "
1240 B$="  "
1250 Y1=[illegible]:Y2=[illegible]:Y3=[illegible]:Y4=[illegible]:Y5=[illegible]:Y6=[illegible]
1260 CLS
1270 D$(1)=CHR$[illegible]+CHR$(118)+CHR$(119)
1280 D$(2)=CHR$[illegible]+CHR$(99)+CHR$(120)
1290 D$(3)=CHR$(121)+CHR$(125)+CHR$(91)
[illegible]
```

```
1293 O$(6)=CHR$(8)+CHR$(117)+CHR$(114)
1300 @$=CHR$(13)
[illegible]
1320 REM ** START OF PROGRAM **
1330 FOR F=0 TO 38:PLOTF,6,96:NEXTF
1340 POKE49086,94:POKE49113,94
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
1450 T1=T1+300:T2=T2+300:T3=Y3+300:Y4=Y4+300:Y5=Y5+300:Y6=Y6+300
[illegible]
1470 REM ** INSTRUCTIONS **
1480 CLS
1490 PRINT:PRINT
1500 PRINT CHR$(14)
1510 PRINT CHR$(27);"E   "+CHR$(27);"N    TORPEDO ATTACK"
1520 PRINT CHR$(4)
1530 PRINT:PRINT
1540 PRINT"    THIS PROGRAM IS DESIGNED BY"
1550 PRINT"          [illegible] .HALL"
1560 PRINT
1570 PRINT"TO OPERATE THIS GAME USE THE LEFT AND"
1580 PRINT"RIGHT CURSORS TO MOVE LEFT AND RIGHT,"
1590 PRINT"THE UP CURSOR TO FIRE,AND IF IT IS"
1600 PRINT"ON LEVEL 1 THEN THE DOWN CURSOR"
1610 PRINT"WILL STOP YOUR DEFENDER.NOTE THAT"
1620 PRINT"THE ARROWS AT THE BOTTOM MARK THE"
1630 PRINT"FIRING RANGE,WHERE YOU CAN HIT THE"
1640 PRINT"BOATS.EACH BOAT IS WORTH A CERTAIN"
1650 PRINT"AMOUNT OF POINTS,AS LISTED BELOW"
1660 PRINT
1670 PRINT"    AIRCRAFT CARRIER  =10 POINTS"
1680 PRINT"    DESTROYER         =20 POINTS"
1690 PRINT"    SPEED BOAT        =50 POINTS"
1700 PRINT
1710 PRINT CHR$(27);"E   "+CHR$(27);"L  PRESS  SPACE  BAR  TO
CONTINUE"
1720 GET A$:IF A$=CHR$(32) THEN 1735 ELSE 1730
1730 CLS
1740 PRINT:PRINT
1750 PRINT"    YOUR JOB IS TO PREVENT"
1760 PRINT"    THE ENEMY (BOATS) FROM"
1770 PRINT"    REACHING YOUR BASE IN"
1780 PRINT"    THE TIME GIVEN.EVERY"
1790 PRINT"    100 POINTS WELL GIVE"
1800 PRINT"    YOU AN EXTRA 100 ONTO"
1810 PRINT"    YOUR TIME.EVERY TIME"
1820 PRINT"    YOU FIRE 10 BITS OF TIME"
```

```
1830 PRINT"    ARE TAKEN OFF YOUR TIME"
1840 PRINT"    SO THE BEST WAY TO PLAY"
1850 PRINT"    IS TO SHOOT AS LITTLE AS"
1860 PRINT"    POSSIBLE.ALSO EVERY TIME"
1870 PRINT"    YOU HIT A BOAT,YOUR DEFENDER"
1880 PRINT"    IS MOVED TO THE RIGHT"
1890 PRINT"    BY ONE PLACE,AND ALSO"
1900 PRINT"    YOUR DEFENDER MOVES IN"
1910 PRINT"    THE DIRECTION WHICH IT"
1920 PRINT"    WAS GOING IN BEFORE UNLESS"
1930 PRINT"    THE DIRECTION IS CHANGED"
1940 PRINT"    OR YOU FIRE,OR ON LEVEL ONE"
1950 PRINT"    YOU STOP THE DEFENDER."
1960 PRINT
1970 PRINT CHR$(27);"Q    "+CHR$(27);"L  PRESS  SPACE  BAR  TO
CONTINUE"
1980 GET A$:IF A$=CHR$(32) THEN 1990 ELSE 1980
1990 CLS
2000 PRINT:PRINT
2010 RETURN
3500  IF  X AND SCRN(A,5)=L THEN  EXPLODE:WAIT10:PLOTA,B,R$:GOSUB
890:RETURN
3510 IF X AND SCRN(A+1,5)=L THEN  EXPLODE:WAIT10:PLOTA,B,R$:GOSUB
890:RETURN
3520  IF X AND SCRN(A+2,5)=L THEN EXPLODE:WAIT10:PLOTA,B,R$:GOSUB
890:RETURN
3530 IF R=3 OR R=6 THEN RETURN
3540 IF X AND SCRN(A+3,5)=L THEN  EXPLODE:WAIT10:PLOTA,B,R$:GOSUB
890:RETURN
3550  IF X AND SCRN(A+4,5)=L THEN EXPLODE:WAIT10:PLOTA,B,R$:GOSUB
890:RETURN
3560 RETURN
9000 REM ** STARTING POINTS **
9010 R=INT(6*RND(1))+1
9020 IF R=1 OR R=2 OR R=3 THEN DI=1 ELSE DI=2
9030 IF DI=1 THEN A=30 ELSE A=5
9040 IF DI=1 THEN A=A-Q
9050 IF DI=2 THEN A=A+Q
9060 RETURN
```

(Listing resized for convenience)

Why you should subscribe to ORIC OWNER

Your Oric 1 is one of the most powerful low-cost computers on the market today. Oric Owner is the only magazine totally dedicated to the Oric and its supporting hardware and software.

From issue to issue Oric Owner is able to bring you in-depth information on the Oric, advance news on the latest add-ons, superb programs to try and 'behind the scenes' interviews with the engineers who designed it.

If you wish to keep ahead of the latest developments and news on the Oric then Oric Owner is essential. At only £10 for a year's subscription of 6 issues (£15 for overseas readers) can you afford to be without Oric Owner?

SUBSCRIBE TODAY!

subscription form

Please send me the next 6 issues of ORIC OWNER

Starting from issue .....

I enclose a cheque ..... for £10

Name.....

Address.....

.....

.....

PLEASE NOTE: All cheques must be made payable to: Tansoft Ltd, Unit 1 & 2, Techno Park, Newmarket Road, Cambridge

TANSOFT

ORIC Software

ORIC-MON
Includes mnemonic assembler/disassembler, block move and verify, data and text entry and cassette and printer handling. Includes manual. £15.0 inc. VAT–16K or 48K

ORICMUNCH
Munch the Bugmonsters and eat the power pills, clear the maze to reach the high score. £7.95 inc. VAT –16K or 48K.

ORICBASE
A complete database system. A sophisticatad query language allows you to search and sort your data, keep running totals and print reports. Full manual. £14.50 inc. VAT – 48K.

House of Death

An impressive follow-up to Zodiac, discover the secrets of the hauntad house, avoid the witch and the axe murderer and come away with incredible treasures. £9.99 inc. VAT – 48K.

ZODIAC
One of our most successful games. Discover the twelve cunningly disguised signs of the zodiac which will lead you to six treasures. Full of puzzles, puns and misleading clues. £9.99 inc. VAT – 48K.

TANSOFT LANGUAGE MASTERCLASS
If you are going abroad for pleasure or business these courses will give you a basic understanding of the language. Based on the new Gruneberg Link-Word method.

Spanish
Italian
German
French

supplied with program cassette and spoken vocabulary pronunciation cassette. £12.95 each inc. VAT – 48K.

FORTH

A full implementation of this exciting new computer language. Includes editor and assembler plus language extensions. £15.00 inc. VAT – 48K.

You received your first issue of Oric Owner free with your machine. Why not subscribe now and keep in touch with all the latest developments in the Oric range.

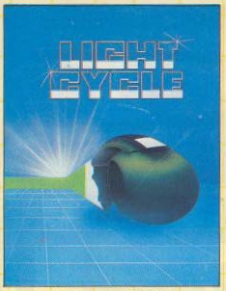
Please send me _____ TITLE _____ QUANTITY _____ PRICE _____ *Delete as appropriate.
Oricmon
Forth
Oricbase
House of Death
*French/Spanish/Italian/German
Oricmunch
Oric Owner (6 issues £10.00)

NAME _____

ADDRESS _____

I enclose cheque/P.O./Access No. _____ for £ _____

Tansoft Ltd., Unit 1 & 2, Techno Park, Newmarket Road, Cambridge Tel: Order hotline: Teversham (0223) 2261.



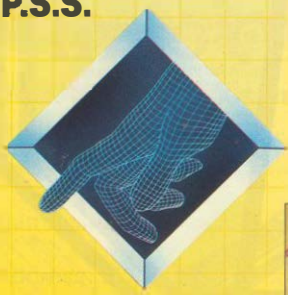
LIGHT CYCLE

CENTIPEDE

INVADERS

REAL SOFTWARE FOR YOUR ORIC 1

AVAILABLE NOW DIRECT FROM P.S.S.



CENTIPEDE

AN ALL MACHINE CODE VERSION OF THE ARCADE GAME. FEATURES — MUSHROOMS, SPIDERS, FLEAS, MUSHROOM LAYING BEETLES, SOUND, COLOUR, HI RES. GRAPHICS, HIGH SCORE TABLE, GAME RESET.

THIS PROGRAM HAS BEEN APPROVED BY ORIC PRODUCTS INTERNATIONAL

6·95

AVAILABLE SOON
LIGHT CYCLE

YOU HAVE BEEN CHOSEN BY THE M.C.P. TO RACE ON THE INFAMOUS GRID. CAN YOU SURVIVE? 100% MACHINE CODE. 1 OR 2 PLAYERS. RACE THE COMPUTER OR ANOTHER PLAYER.....SO FAST YOU WON'T BELIEVE IT!!

6·95

INVADERS

QUITE SIMPLY THE ULTIMATE VERSION OF "SPACE INVADERS" AVAILABLE TODAY. FEATURES — ALL MACHINE CODE, SMOOTH ACTION LASER BASE, "LIVE" ALIENS, CRUMBLING WALLS, EXPLOSIVE SOUND EFFECTS, FULL COLOUR, HI RES. GRAPHICS, INCREASING GAME SPEED, BONUS POINTS, MOTHER SHIPS AND MORE.....

6·95

ORIC-MON

ORIC-MON IS A POWERFUL MACHINE CODE MONITOR PLUS A FULL DISASSEMBLER FOR YOUR ORIC 1. AN ESSENTIAL SOFTWARE TOOL FOR WRITING, DEBUGGING AND RUNNING YOUR MACHINE CODE PROGRAMS OR MACHINE CODE SUBROUTINES TO BE CALLED FROM BASIC. CONTAINS ALL THE FEATURES OF MONITORS FOUND ON MUCH LARGER MACHINES. A SUPERB PROGRAMMING AID FOR YOUR ORIC 1.

8·95

HOPPER

ALL MACHINE CODE VERSION OF "FROGGER" FOR YOUR ORIC 1.

FEATURES — VARIABLE SPEED TRAFFIC, 3 LANE HIGHWAY, SNAKES, TURTLES, DIVING TURTLES, FLEAS, LOGS, DECREASING TIME LIMIT, EVER INCREASING GAME SPEED, HI SCORE TABLE, FULL COLOUR, AMAZING GRAPHICS, SUPERB SOUND. YOU HAVEN'T PLAYED "FROGGER" UNTIL YOU'VE PLAYED "HOPPER".

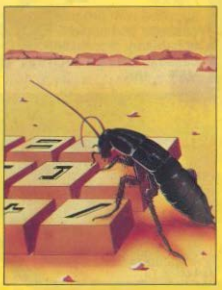
6·95

AVAILABLE SOON
THE ULTRA

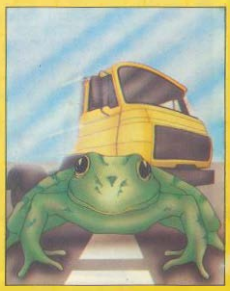
THIS IS THE GAME FOR ALL YOU ARCADE WIZARDS OUT THERE. ZAPPING ALIENS WAS EASY UNTIL "THE ULTRA" THIS GAME IS TOTALLY WRITTEN IN MACHINE CODE TO MAKE IT THE FASTEST, TRICKIEST ARCADE GAME OF THEM ALL.

16 DIFFERENT SCREENS COLOUR, SOUND AND HI RES. GRAPHICS

6·95



ORIC-MON



HOPPER

PERSONAL SOFTWARE SERVICES

FOR INSTANT CREDIT CARD SALES TEL (0203)667556

CHEQUE OR P.O. TO P.S.S.
452 STONEY STANTON RD.
COVENTRY CV6 5DG.

TRADE ENQUIRIES CONTACT JOHN FLETCHER.