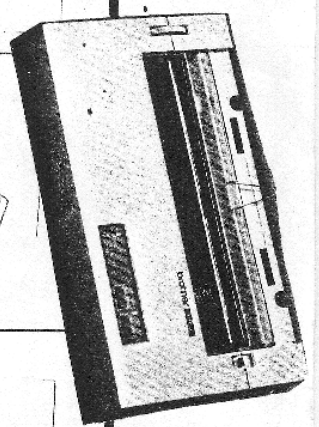
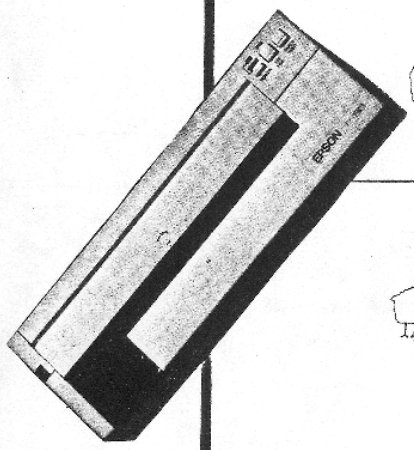
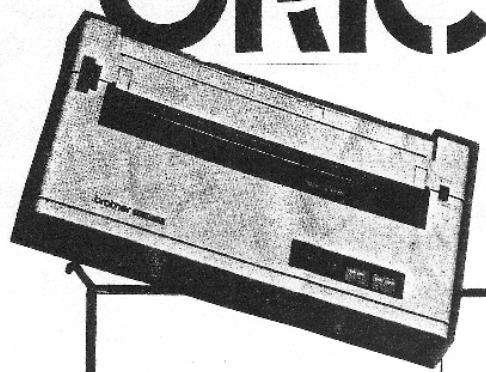


# ERIC

# NOTES

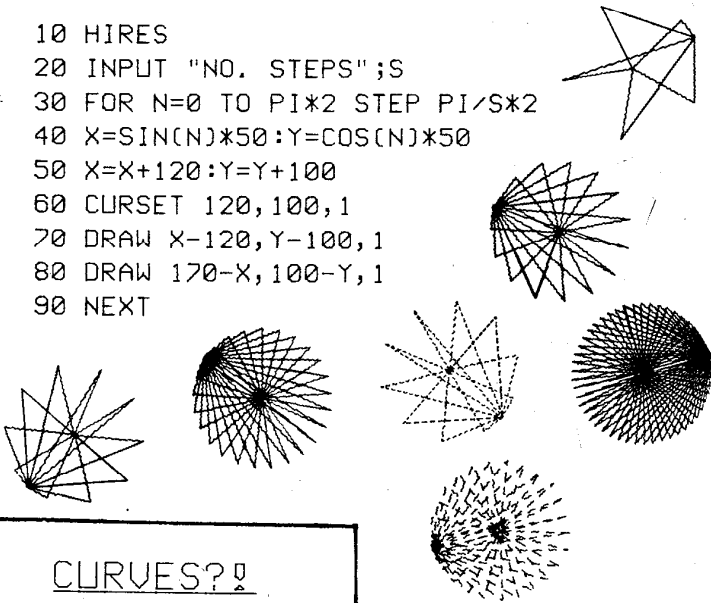



The Chicken & The Egg

# hires page

```

10 HIRES
20 INPUT "NO. STEPS";S
30 FOR N=0 TO PI*2 STEP PI/S*2
40 X=SIN(N)*50:Y=COS(N)*50
50 X=X+120:Y=Y+100
60 CURSET 120,100,1
70 DRAW X-120,Y-100,1
80 DRAW 170-X,100-Y,1
90 NEXT
    
```



```

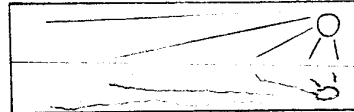
5 S=INT(RND(1)*50)
10 HIRES
20 FORN=0TOPI*2-.1STEPPI/S
30 X=120+(-SIN(N)*40)
40 Y=70+(COS(N+PI)*45)
50 CURSETX,Y,1
60 G=150+(COS(N-PI)*50)
70 H=H+120/S
80 DRAWH-X,G-Y,1
90 NEXT
100 GETA$:RUN
    
```

## CURVES?!

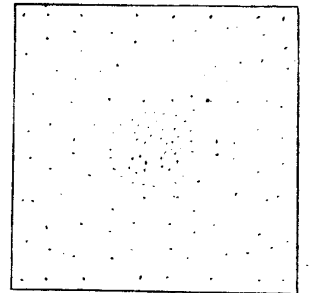
```

10 REM *** STRAIGHT CURVES ***
20 I=5:GOTO40
30 INPUT "HOW MANY LINES (I=HIGH - 40
=LOW)";I
40 X=-I
50 PAPER0:INK7
60 HIRES:CURSET50,150,1:DRAW150,0,1
65 FOR Y=150 TO 0 STEP -I
67 X=X+I
70 CURSET50,X,1
80 SOUND1,X,6
85 DRAWX,Y,1
90 NEXTY
95 SOUND1,0,0
100 GOT030
    
```

## SYMMETRY ΣYMMETRY



THIS IS A RED (WELL BLACK) HERRING!



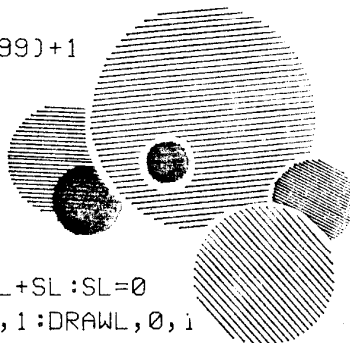
```

5 HIRES
10 CURSET45,25,1:DRAW150,0,1:DRAW0,150,
1:DRAW-150,0,1:DRAW0,-150,1
20 X=INT(RND(1)*75):Y=INT(RND(1)*75)
30 CURSET45+X,25+Y,1
40 CURSET195-X,25+Y,1
50 CURSET45+X,175-Y,1
60 CURSET195-X,175-Y,1
70 GOT020
    
```

## FILLED CIRCLES

```

10 HIRES
20 X=120:Y=100
30 S=INT(RND(1)*99)+1
50 R=S*S
60 S=S-1
70 IFS<0THENEND
80 Z=SQR(R-S*S)
90 L=Z+Z
100 SL=X-Z
110 IFSL<0THENL=L+SL:SL=0
120 CURSETSL,Y+S,1:DRAWL,0,1
140 CURSETSL,Y-S,1:DRAWL,0,1
150 GOT060
    
```



```

10 HIRES
13 C=INT(RND(1)*20)+1
15 R=INT(RND(1)*60)
17 T=INT(RND(1)*(199-2*R))+R
20 P=INT(RND(1)*7)
30 I=INT(RND(1)*7):IFI=PTHEN30
40 PAPERP:INKI:S=RND(1)
50 X=INT(RND(1)*200)+20
60 Y=INT(RND(1)*199)
70 FORN=20TO220STEP C
80 U=U+S
90 U=T+(SIN(U)*R)
100 CURSETX,Y,1
110 DRAWN-X,U-Y,1
120 NEXT
130 GETA$:RUN
    
```

# SOUND TRACK

```

100 O1=4:V1=14
110 O2=2:V2=0
115 CLS:INPUT"SPEED (1-20)";W:IFW<10RW>
20THEN115
119 SHOOT:WAIT100
120 REPEAT
130 READ D,N1,N2
140 IFN1>12THENO1=O1+1:N1=N1-12
150 IFN2>12THENO2=O2+1:N2=N2-12
160 IFD=1THENSOUND4,2,15
170 IFN1=0THENSOUND2,0,0:GOTO190
180 MUSIC2,O1,N1,V1
190 IFN2=0THENSOUND3,0,0:GOTO210
200 MUSIC3,O2,N2,V2:PLAY7,1,1,2000
210 SOUND4,0,0
220 WAIT W
224 O1=4:O2=2:C=C+1
225 IFC=188THENV1=15:V2=15
226 IFC=136THENV1=14:V2=0
227 IFC=56ORC=168THENV1=0:V2=10
228 IFC=79THENV1=1:V2=0
229 IFC=88THENV1=12:V2=12
230 UNTIL C=191
240 WAIT100:SHOOT
300 REM *DATA*
310 DATA 1,0,0,0,0,0,1,0,0,1,0,0,1,0,0,
0,0,0,1,0,0,1,0,0
320 DATA 1,0,1,0,0,0,1,0,0,1,0,5,1,0,1,
0,0,0,1,0,0,1,0,5
330 DATA 1,0,3,0,0,0,1,0,0,1,0,6,1,0,3
,0,0,0,1,0,0,1,0,6
340 DATA 1,5,1,0,0,0,1,5,0,1,5,5,1,6,6
,0,6,0,1,8,0,1,6,6
350 DATA 1,5,1,0,8,0,1,13,5,1,13,5,1,15
,3,0,10,0,1,8,5,1,5,1
360 DATA 1,5,1,0,5,0,1,0,5,1,0,0,1,1,3,
0,0,0,1,1,5,1,1,1
370 DATA 1,0,1,1,0,0,1,0,3,1,0,0,1,0,5,
1,0,0,1,0,6,1,0,0
380 DATA 1,5,1,0,0,0,1,5,0,1,5,5,1,6,6
,0,6,0,1,8,0,1,6,6
390 DATA 1,5,1,0,8,0,1,13,5,1,13,5,1,15
,3,0,10,0,1,8,5,1,5,1
400 DATA 1,5,1,0,5,0,1,0,5,1,0,0,1,1,3,
0,0,0,1,1,5,1,1,1
500 DATA 1,0,6,1,0,0,1,0,5,1,0,0,1,0,3,
1,0,0,1,0,1,1,0,0
510 DATA 0,5,13,0,5,13,0,3,12,0,5,13,0,
5,13,0,3,13,0,0,0,1,0,0
520 DATA 0,3,12,0,3,12,0,1,10,0,3,12,0,
3,12,0,2,12,0,0,0,1,0,0
530 DATA 0,5,13,0,5,13,0,3,12,0,5,13,0,
5,13,0,3,13,0,0,0,1,0,0

```

```

540 DATA 1,8,15,0,8,15,1,3,8,0,3,8,1,0,
0,1,0,0,1,0,0,1,0,0
550 DATA 0,0,13,0,0,12,0,0,10,0,0,8,0,0,
6,0,0,5,0,0,3,0,0,1
560 DATA 1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,
1,0,0,1,0,0,1,0,0
570 DATA 1,5,1,0,0,0,1,5,0,1,5,5,1,6,6
,0,6,0,1,8,0,1,6,6
580 DATA 1,5,1,0,8,0,1,13,5,1,13,5,1,15
,3,0,10,0,1,8,5,1,5,1
590 DATA 1,5,1,0,5,0,1,0,5,1,0,0,1,1,3,
0,0,0,1,1,5,1,1,1
600 DATA 1,0,1,1,0,0,1,0,3,1,0,0,1,0,5,
1,0,0,1,0,6,1,0,0
610 DATA 1,5,1,0,0,0,1,5,0,1,5,5,1,6,6
,0,6,0,1,8,0,1,6,6
620 DATA 1,5,1,0,8,0,1,13,5,1,13,5,1,15
,3,0,10,0,1,8,5,1,5,1
630 DATA 1,5,1,0,5,0,1,0,5,1,0,0,1,13,5
,0,0,0,1,13,13

```

Paul M.

```

0 REM MAKE-A-TUNE
2 SP=INT(RND(1)*30)+5
3 K=INT(RND(1)*12)
4 RESTORE:PH=PH+1
5 FORQ=1TO4:NN=0
6 IFPH=2ANDQ=4THEN110
8 CL=2:TL=0
10 REPEAT:NN=NN+1
15 IFNN=3ANDCL=2THENL=2:GOTO30
20 L=INT(RND(1)*CL)+1
30 READC(1),C(2),C(3)
40 R=INT(RND(1)*3)+1
50 N=C(R)
60 TL=TL+L:IFL=2THENCL=1
64 O=3:N=N+K
65 IFN>12THENN=N-12:O=O+1
67 IFN<1THENO=O-1:N=N+12
70 MUSIC1,O,N,10
75 IFQ=4THENPULL:GOTO150
80 WAITL*SP
90 UNTILTL=4
100 NEXT:SOUND1,0,0
110 MUSIC1,3,K+1,10
120 WAIT4*SP:SOUND1,0,0
130 WAIT50:RUN
150 WAIT4*SP
160 GOTO4
200 DATA1,1,1
210 DATA3,8,13
220 DATA6,8,10
230 DATA3,6,10
240 DATA8,12,13
250 DATA3,10,13
260 DATA3,6,8
270 DATA1,3,6
280 DATA-1,0,3
290 DATA1,3,5

```

Paul M

# A BIT OF FUN

```

0 REM ** USE UP AND DOWN CURSOR KEYS TO SPEED UP AND SLOW DOWN **
1 PAPER0: INK7: POKE#26A, 10
2 CLS
5 X=20
6 PLOT 11, 10, "======"
10 IFPEEK(520)=156 THEN X=X-2
20 IFPEEK(520)=180 THEN X=X+2
30 IFX<2 THEN X=2
40 MUSIC1, 3, 8, 0
50 PLAY7, 0, 3, X
60 PLOT10, 10, 126: PLOT20, 10, 126
70 WAITX/2
80 PLOT10, 10, 254: PLOT20, 10, 254
90 WAITX/2
100 GOTO10
    
```

R. H. Morgan-Jones

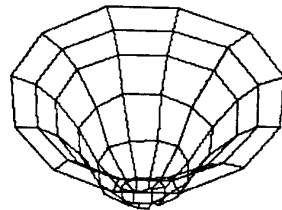
```

0 CLS
1 PRINT" SYMETRICAL CIRCLE for O.P.E.L
"
2 PRINT"PRESS CONTROL+C TO STOP AND TYPE
CONT TO CONTINUE by Lloyd Presto
n"
6 PRINT" COUNT DOWN "
7 FOR X=5 TO 1 STEP -1
8 PRINTX
9 WAIT 100:NEXT :PRINT" GO "
:WAIT 100
10 HIRES
20 PAPER 2: INK 4
30 CURSET 120, 0, 3
40 FILL 200, 20, 192
50 CURSET 120, 100, 3
60 FOR A=90 TO 1 STEP -2
70 CIRCLEA, 1
80 NEXT
90 INK 2 : PAPER 4
100 FOR B=1 TO 90 STEP 2
200 CIRCLEB, 1
300 NEXT
400 GOTO 10
    
```

by Lloyd

```

0 PLAY1, 0, 0, 0: FOR A=5 TO 30
1 LORES 1: SOUND 1, A, 15
2 PLOT 4, 15, 95: PLOT 5, 15, 36
3 PLOT 30, 15, 40: PLOT 31, 15, 95
4 PLOT A, 15, 44
5 NEXT A
6 PLOT 30, 14, 65: PLOT 32, 14, 50: PLOT 31, 1
3, 64
7 PLOT 29, 15, 34: PLOT 33, 15, 33 :EXPLODE
8 WAIT 100 :LORES0:CLS
    
```



```

100 HIRES
110 FORN=1T0100
120 FILL1, 1, 21
130 FILL1, 1, 17
140 NEXT
150 CURSET35, 30, 3
160 FILL130, 26, 170
170 PING:WAIT40
180 CURSET45, 40, 3
190 FILL130, 26, 100
200 PING:WAIT40
210 CURSET55, 50, 3
220 FILL130, 26, 127
230 PING:WAIT40
240 :
250 FORN=1T08
260 FORA=1T07
270 WAIT(8-A)*5
280 PAPER A: INK8-A
290 NEXTA, N
    
```

Huw

This quicky from HUW will use 'FILL' to produce checkered dots of two different sizes overlapping each other like paper with 'plain' paper on top.

# UTILITY TIME

## INBETWEENER

```
=====
5 DIMX(30):DIMY(30):DIMXS(30):DIMYS(30)
:DIMXN(30):DIMYN(30)
7 DIMW(30):DIMS(30)
10 DEFFNA(Z)=INT(Z-((Z-INT(Z))>.49999))
20 POKE#24E,5:POKE#24F,1
100 HIRES
110 GOSUB 500
170 INPUT"No. Steps please";ST
180 IF ST<>INT(ST)THENPRINT"No decimals
":GOTO170
185 INPUT"Single Frames or Cont. Frames
(S/C)";Q$
186 ST=ST-1
190 FORK=1TON-1
200 XS(K)=XN(K)-X(K)
210 YS(K)=YN(K)-Y(K)
220 YS(K)=YS(K)/ST
230 XS(K)=XS(K)/ST
240 NEXT
285 POKE#26A,11
290 L=0
300 FORK=1TON-1
310 W(K)=X(K)
320 S(K)=Y(K)
330 NEXT
335 HIRES
340 REPEAT
345 IF Q$="S"THENHIRES
350 L=L+1:PRINT"Step No.";L
355 CURSETW(1),S(1),1
360 FORM=2TON-1
365 A=W(M)-W(M-1):Q=S(M)-S(M-1)
370 DRAWA,Q,1
375 NEXT
380 A=W(1)-W(N-1):Q=S(1)-S(N-1)
381 DRAWA,Q,1
382 GETA$
390 FORK=1TON-1
400 W(K)=XS(K)*L+X(K):W(K)=FNA(W(K))
410 S(K)=YS(K)*L+Y(K):S(K)=FNA(S(K))
420 NEXT
430 UNTILL=ST+1
450 INPUT"Do you want a re-run? (Y/N)";
A$
455 IF A$="Y"THEN170
460 INPUT"Want to enter new data? (Y/N)
";A$
465 IF A$="Y" THEN RUN
470 GOTO 450
```

```
500 REM ** INPUT SHAPES **
502 POKE#26A,10
505 C=0:CURSET0,0,1
510 N=0
520 REPEAT
530 N=N+1
540 REPEAT
545 CURSET Z,T,2
550 K$=KEY$
560 Z=Z+(K$=CHR$(8))-(K$=CHR$(9))
570 T=T+(K$=CHR$(11))-(K$=CHR$(10))
572 IF Z>239THENZ=239
573 IF Z<0THENZ=0
574 IFT<0THENT=0
575 IFT>199THENT=199
580 CURSET Z,T,2
590 UNTIL K$="E"
595 CURSET Z,T,0
600 IF C=0 THEN X(N)=Z:Y(N)=T
610 IF C=1 THEN XN(N)=Z:YN(N)=T
620 PRINT"End of shape? (Y/N)";
630 GETA$:CLS
635 IF A$<>"Y"ANDA$<>"N"THENPRINT"Re-ent
er Please":PING:GOTO620
640 UNTIL A$="Y"
645 N=N+1
650 IF C=0 THEN C=1:B=N:GOTO510
660 IF N<>B THEN PRINT"Unequal no. poin
ts. PLEASE RE-ENTER":ZAP:GOTO 510
670 RETURN
```

Inbetweening is often used in cartoon animation nowadays, and allows the operator to enter two or more shapes and the computer will then work out the frames inbetween. This means a great saving in time as you don't have to draw anything like as many shapes.

The following program is a simple form of this, allowing you to enter two separate shapes and the computer will then work out the stages inbetween and print them up one by one. eg. you may enter an egg, and a chicken, and the computer will then (in a specified number of frames) turn the egg into the chicken.

When you 'run' the program, a flashing dot should appear in the top left hand side of the screen. Each shape is entered as a series of dots, which the computer will then join up.

# Cont...

You use the cursor keys (↑,↓,→,←) to move the dot around the screen. So to set the first point, you move the 'dot' to where you want the point, and press 'E'. The message "End of shape? (Y/N)" will then appear on the screen if that point is the last point you want to build up your shape, then enter 'Y', but if you want to enter more points, then type 'N'. Use the same method (cursor keys + 'E') to build up the rest of the dots in the shape.

Once you have finished your first shape and entered 'Y' when asked if it was the end of the shape, the flashing dot will appear, and you should use the same method again to enter the second shape. However this must have the same number of dots in it as the last one. If it hasn't, you will get the message 'Unequal no. points,' and be ZAPPED! You will then have to enter the second shape again. So be careful. If you wanted to change a square into a triangle however it is possible by placing two dots at the same position in the triangle so that you only see one dot.

Having entered both shapes you will be asked how many steps you wish to see of the shape as it moves from 'A to B'. The final question you will be asked before you have the excitement of seeing your shape in action is 'Single or Cont. Frames (S/C)' (No don't go and cower away in the far corner, it isn't that complicated!) if you enter 'S' for Single Frames, the screen will be cleared after each image has been displayed (and you have pressed any key), with 'C', continuous frames, they will all be displayed on the same screen as in the bottom left corner box on the cover. Now sit back and watch your images in progression.

With the program in it's present state, you may enter up to 30 dots per shape, but this may be changed by means of line 5. Any variations on this program that you try I would be pleased to hear about, and maybe print.

GET CLICKING!!

Paul Meadows

```

10 A=630
20 REPEAT
30 A=A+1
40 POKEA,20
50 UNTILA=760THENEND

```

strange results should occur. It seems to change the height of the screen, and yet again, I don't know how works! But maybe you would like to investigate it and please bring any results to the meeting!

This 'mega-quicky' from Huw will do a number of different things. On the ORIC-1 it will swap all the characters around ie, after running it will prompt you with:-

```
Uuut )4!
```

Which means 'Ready', don't ask me how it works, I don't know!

This is very useful for protecting programs as reset will not undo it.

But on the ATMOS it seems to mess about with the system 'screen variables'. When it has been run, try moving the cursor to the bottom of the screen and then listing the program. Some

Another address which I am pleased to say will work on the ATMOS and the ORIC-1 is #362. Try this:-

```
POKE #362,55
```

This will switch the REM SOCKET on from the computer, ie if you have a rem socket PLAY will work. To switch off the rem socket use POKE #362,247

I have also discovered that using a different number after the address may aswell as switch the rem socket on or off may repeat the key that you last pressed continuously.

## **ABC BASIC**

**COS** .. This is used to like the mathematical function 'COSINE' .

It can be used as follows:

```
PRINT COS (9)
```

The output will be (surprise,surprise) the cosine of nine.

It can be used in graphics programs in a FOR/NEXT loop.

eg

```
10 HIRE$
20 CURSET10,100,3
30 FOR N=-10 TO 10 STEP.3
40 CURMOD 2,COS(N)*7,1
50 NEXT
```

This can be used in conjunction with SINE (SIN).

It works in exactly the same way as SIN.

**CALL** .. This is used to CALL up a section of the memory and to execute it. Used alot in Machine Code to 'RUN' an MC program. (it does not, as I used to think 'call' someone else via an ORIC MODEM!)

A great deal of care has to be taken using it though because the computer will lock up at any bug, never ending loop or if no RTS sign is present.

It can be used with hex or decimal numbers.

**LOAD/SAVE** .. Yes, it's my favourite to! I am sure everyone knows how to work it, but a few extra commands have been added to it which I will go over. Added to it is AUTO which you add when saving a

program and, when the program is loaded again it will automatically run. In the form:

```
CSAVE"PROG",AUTO
```

The command does not need to be entered when loading. A and E which are used to save blocks of memory, eg the text screen. To save the text screen the format:

```
CSAVE"TEXT",A48000,E49119
```

This would save FROM location 48000 TO 49119.

### **CURSET and CURMOV..**

These are both used on the HIRES screen, CURSET is used to set a position on the screen. eg

```
CURSET 120 , 100 , 1
```

The 120 signifies 120 pixels horizontally, the 100 signifies 100 pixels vertically.

The 1 is an 'fb' code which tells the computer to print a dot. Here is a list of all the fb codes:

- 0.. delete any dot here
- 1.. print a dot here
- 2.. if there is a dot here, rub it out. If not then print one.
- 3.. go to this place but do not do anything here

CURMOV works in the same manner as with fb codes but is a 'relative' drawer ie it will move x and y position from the point it is at already.

```
CURMOV X,Y,fb
```

Both these can be seen in the first program.

By Humr (programmer for PLUG)



TRAVELLING WITH YOUR ORIC

When you go on holiday your computer can get very bored, so on long trips it is best to take him with you; but some can get car sick, if this is the case give him this every 5 hours:

```
DOKE 65251,21
POKE 103,50:CALL#EB8D
```

ORIC's travel best in shoe boxes which shop the violent explosions of car sickness dirtying your car/landrover/bus/train/hovercraft/QE2/Sinclair C5 etc. (however, if you own the latter it may be a good way of getting rid of it).

Travelling is a new thill for your ORIC and so he can get a bit restless, you can put him to sleep with:

```
5 CLS
7 POKE#26A,10
10 FORA=1T0500
20 PRINTCHR$(11);"SHEEP...";A
30 NEXT A
40 CALL#239
```

When he has finished counting ORIC will fall into deep sleep.

There are three things that will make your ORIC espacially car sick, yes, you've guessed it, a SPECTRUM, an COMMODORE or an ATARI!

If left at home, ORIC must have something to occupy his mind with e.g:

```
10 FORA=1T01000000
20 ?"I AM A RABBIT HUTCH, "A/3.5;
30 ? INT(RND(1)*26)+65
40 NEXTA
50 RUN
```

This program has no significance but it is something for him to do!

If you are at a camp site it is important that you find a mains supply, and that you do not mistake it for a water supply! At night in a tent ORIC should be put to baed in his shoe box and wrapped up warmly so he will not catch a chill!

HUW

```
0 CLS:PAPER0:INK7
1 REM
2 REM *****
4 REM ** MAZE DOT UNREMOVER **
5 REM * Will only work with spaces *
6 REM * between dots more than1 *
7 REM * character *
8 REM *****
9 REM
10 PLOT0,10,"* * * * * * * * *
* * "DOTS
20 X=37
30 X=X-1:IF X=0 THEN END
35 A$=" "
40 IF SCRNX(X-1,10) <> 32 THEN T=TRUE
45 IF T=TRUE THEN R=R+1
50 IF R=3 THEN A$="*":R=FALSE:T=FALSE
60 PLOTX,10,"+"
70 PLOTX+1,10,A$
75 WAITS
80 GOTO 30
```

LINE 40 checks whether the character just to the left of 'thing' is a dot, if it is, it remembers that there was one there.

LINE 45 asks line 40 if there was a dot, if there was it adds one to the variable 'R'.

LINE 50 checks whether variable 'R' is three, if it is then string variable A\$ is set to be a dot, which will then be printed just to the right of 'the thing'. It will then set 'R' back to zero and set 'T' (which is a pointer for 'R') back to zero.

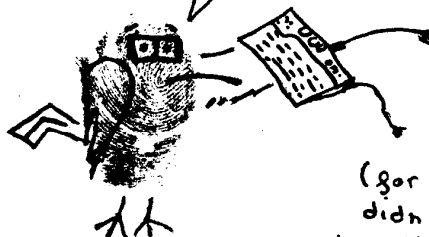
LINE 60 if you don't understand this line, read the top of the page where it says 'FOR THE MORE EXPERIENCED PROGRAMMER'!

LINE 70 prints just to the right of the 'thing' either, a space if the 'thing' has not just been over a dot, to rub the last 'thing' out. Or a dot, if it has just been over a dot.

This simple program could be extended to make a full arcade game, which is what I intend to do in the near future.

By  
HuW

ALAS! POOR ORIC,  
I KNEW IT WELL!



(for those who didn't know, that's HAMLET!)



ORIC P.L.U.G.  
104 Wood Road  
King's Cliffe  
PETERBOROUGH  
PE6 6XR  
Telephone (078 087) 665