

Here's a collection of 49 games guaranteed to make you glad you invested in a Spectrum.

Tim Hartnell, the world's most widely published computer author (and recently described by Personal Computer World magazine as 'Mr Sinclair') has drawn on the work of some of the UK's most talented young programmers to bring together this incredible collection of explosive games for your Spectrum.

The programs, many of which feature machine code subroutines, include:

**PROTECTOR**  
**WIPE-OUT**  
**CHOPPER SQUAD**  
**DODGEMS**  
**SNAKE**  
**ZOMBIES**  
**DOORS OF DOOM (Adventure)**  
**GOLD RUSH**  
**SPACE TREK**  
**3-D MAZE (with full graphics)**

Don't waste a minute longer reading the back of this book in the shop. Buy it now, and rush home, and start your Spectrum exploding.

**Another great book from  
INTERFACE PUBLICATIONS**

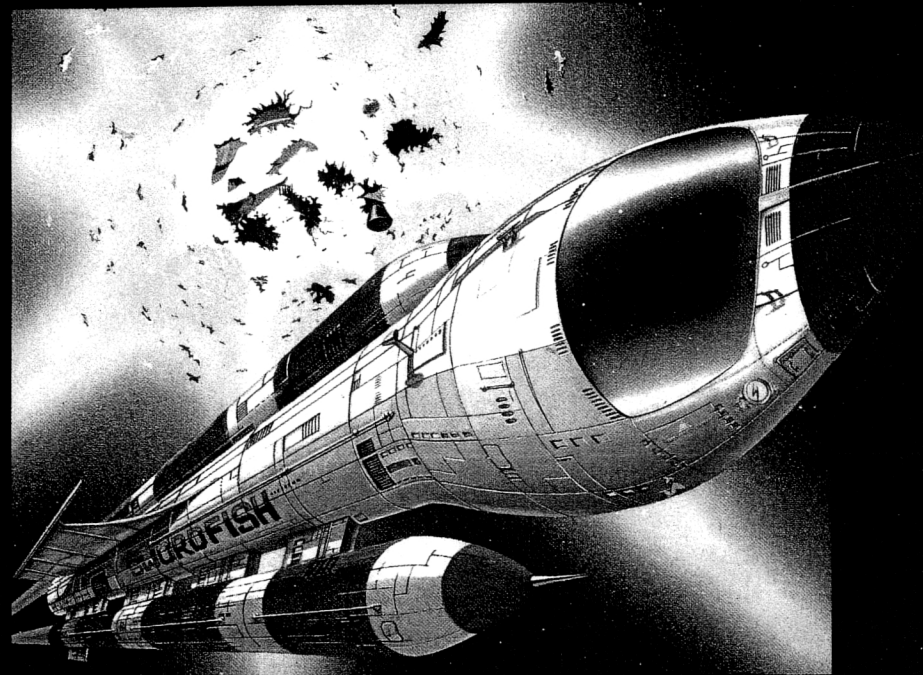
**49 EXPLOSIVE GAMES  
FOR THE**

**ZX SPECTRUM**

**TIM HARTNELL**



# 49 Explosive Games for the ZX Spectrum



**Tim Hartnell**

With David Perry, Graham Charlton, Neil Pellinacci and Malcolm Young

**49 Explosive Games  
for the ZX Spectrum**

**Tim Hartnell**

**with David Perry, Graham Charlton,  
Neil Pellinacci and Malcolm Young**

**INTERFACE**  
**PUBLICATIONS** 



28.00

**Interface Publications**

## **49 Explosive Games for the ZX Spectrum**

**Tim Hartnell**

Tim Hartnell is the most widely published author of computer books in the world. Founder of the UK's National ZX Users' Club, and founding editor of *ZX Computing* magazine, Tim has been at the forefront of developments of Sinclair computers and their application since the beginning. This book is a sequel to the highly successful work *49 Explosive Games for the ZX81*. Other work by Tim Hartnell include *The Giant Book of Computer Games* (Fontana/Interface, 1983), *Mastering the Timex Sinclair 1000* (Bantam Books, 1983) and *The Book of Listings* (BBC Publications, 1983). Tim is editor of the Virgin Books computer games series, wrote the master text and edited the 'Getting Started' series for Future and acts as computer consultant to Fontana Paperbacks.

First published in the UK by:

Interface Publications,  
9-11 Kensington High Street,  
London, W8 5NP

Copyright © Hartnell, 1983

First printing: November, 1983

ISBN 0 907563 53 8

The programs in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. Whilst every care has been taken the publishers cannot be held responsible for any running mistakes which may occur.

ALL RIGHTS RESERVED

No use whatsoever may be made of the contents of this volume — programs and/or text — except for private study by the purchaser of this volume, without the prior written permission of the copyright holder.

Reproduction in any form or for any purpose is forbidden.

Artwork reproduced, with permission, from *Marine Art Clipbook* (Spectre, Peter H. and Putz, George; Van Nostrand Reinhold Company, New York, 1980)

Books published by INTERFACE PUBLICATIONS are distributed in the UK by THE COMPUTER BOOKSHOP (021-706 5511) and in Australia by PITMAN PUBLISHING PTY LTD.

Printed and bound by Short Run Press Ltd., Exeter

# Contents

## ARCADE GAMES

Protector .....	3
Hi-Rize .....	8
Wipe-Out .....	16
Birds .....	19
Chopper Squad .....	31
Snake .....	37
Dodgems .....	41
Zombies .....	47

## ADVENTURES

TROLL and THE CITY .....	59
Doors of Doom .....	77

## MOVING GRAPHICS GAMES

Gold Rush .....	103
Tarantula .....	111
Jungle Job .....	118
Dodge .....	124
Duel Cabbageway .....	126
Ghost Chase .....	128
Frog on a Log .....	132

## LEISURE LINES

Card Pairs .....	141
Sub Search .....	145
Hangman .....	155

## SPACE GAMES

Stellar Evade .....	161
Space Trek .....	162
Stellar Probe .....	170
Lunar Storm .....	178

## THE LURE OF THE MAZE

Maze-maker .....	185
Mangled Mazes .....	187
Walls of Suspense .....	191
Wall of Suspense .....	193
Dual Level 3-D Maze .....	194
Scrolling Maze .....	196
Rollermaze .....	202
Three-D Maze .....	212
Three-D Maze II .....	219

## TWO PLAYER GAMES

Squares .....	227
Four in a Row .....	236
Tanx .....	239

## MACHINE CODE UTILITIES

Memory Monitor .....	245
Machine Code Screensave .....	250
BASIC Screensave .....	251
Ireland .....	252
Master Copier .....	255

## UTILITIES AND DEMONSTRATIONS

Biorhythms .....	261
Paint Pot .....	263
Timothy Leary .....	264
Color Test .....	266
Logic Gate Emulator .....	267
Aural Assault .....	269
Rainbird .....	270
Line of Best Fit .....	273
Hall of Fame .....	276
Spec File .....	278

## STRUCTURED PROGRAMMING

Sketching an outline .....	287
----------------------------	-----

## FOREWORD

Books as solid as this one don't just spring out of the Spectrum without a serious investment of time and effort.

And there has been plenty of both time and effort spent making sure this collection of games for your Spectrum is as good as we could make it.

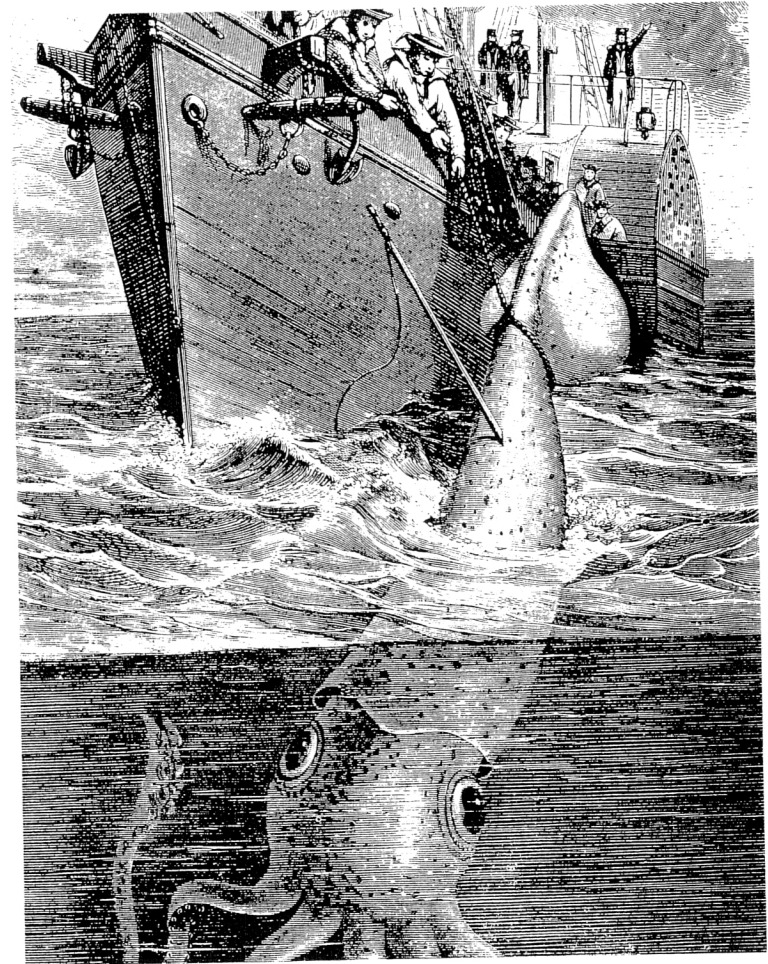
Drawing on the work of some of the UK's best young computer programmers, I've brought together a series of major programs for the Spectrum which deserve the label 'software quality'. The major contributors were Malcolm Young, David Perry, Graham Charlton and Neil Pellinacci, whose work forms the bulk of this book.

Others I contracted to produce programs for this collection include Scott Vincent, Raymond Blake, Neal Cavalier-Smith, Paul Toland, Andrew Sweetland, Martin Jones, Graham White, Tim Rogers, Damian Steele, Neville Predebon and Michael Briggs. They also deserve congratulations for the quality and 'playability' of the programs they've contributed.

Time to start your Spectrum exploding.

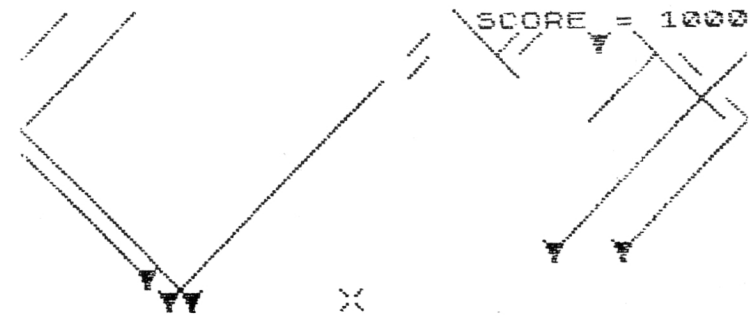
**Tim Hartnell,  
London, 1983.**

# ARCADE GAMES



# Protector

No need now to spend your money in the arcade. PROTECTOR gives you the chance to save the world (or at least a few cities) as this sample run shows:



Your controls are "M", "Q", "A", "O" and "P". Time to start protecting...

```
10 REM RUN WITH CAPS LOCK ON
20 CLS
30 FOR F=1 TO 49
40 BORDER F/7
50 BEEP 0.05,F
60 PRINT AT 5,10; INK 0; PAPER
4; FLASH 1;"PROTECTOR"
70 NEXT F
80 LET SC=0
90 LET HS=0
100 RESTORE
```



```

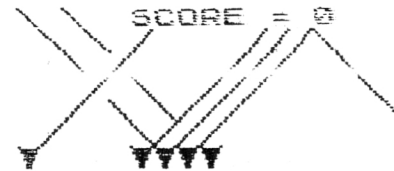
680 GO SUB 0970
690 GO SUB 0920: LET U=U+1: IF
U>15 THEN LET U=15
700 LET D=U: GO TO 0290
710 IF L(B,1)<18 THEN GO TO 036
0
720 IF SCREEN$(L(B,1),L(B,2))<
>" " THEN GO TO 0800
730 LET Z=L(B,2)
740 IF Z>1 AND Z<6 THEN LET Z$(
1,1)=" ": LET Z$(1,2)=E$: GO
TO 0840
750 IF Z>5 AND Z<10 THEN LET Z$(
2,1)=" ": LET Z$(2,2)=E$: GO
TO 0840
760 IF Z>9 AND Z<14 THEN LET Z$(
3,1)=" ": LET Z$(3,2)=E$: GO
TO 0840
770 IF Z>17 AND Z<22 THEN LET Z
$(4,1)=" ": LET Z$(4,2)=E$: G
O TO 0840
780 IF Z>21 AND Z<26 THEN LET Z
$(5,1)=" ": LET Z$(5,2)=E$: G
O TO 0840
790 IF Z>25 AND Z<30 THEN LET Z
$(6,1)=" ": LET Z$(6,2)=E$: G
O TO 0840
800 LET L(B,1)=L(D,1): LET L(B,
2)=L(D,2): LET N(B)=N(D): LET X$(
B)=X$(D): LET D=D-1
810 IF D>0 THEN GO TO 0360
820 IF W=0 THEN GO TO 0850
830 GO TO 0600
840 BEEP .15,2: GO SUB 0920: GO
SUB 0950: GO TO 0800
850 CLS : FLASH 1: FOR X=1 TO 2
2: PRINT "=====
=====": NEXT X: PRINT AT 10
,12:"GAME OVER": FLASH 0
860 PRINT
870 FLASH 0: IF SC>=HS THEN PRI
NT "WELL DONE YOU HAVE THE NEW H
IGH SCORE
": LET HS=SC: INPUT "ENTER Y
OUR NAME ";N$
880 PRINT : PRINT PAPER 0: INK
7;N$:" HAS THE HIGH SCORE"
890 PRINT : PRINT FLASH 1:"PRES
S B TO PLAY AGAIN"
900 IF INKEY#<>"B" THEN BEEP 0.
3,1: GO TO 0900

```

```

910 LET SC=0: GO TO 0020
920 LET Q$=" "+Z$(1,1)+Z$(2,1)
+Z$(3,1)+" "+Z$(4,1)+Z$(5,1)+
Z$(6,1)+" "
930 LET Q$=Q$+" "+Z$(1,2)+Z$(2
,2)+Z$(3,2)+" "+Z$(4,2)+Z$(5,
2)+Z$(6,2)+" "
940 RETURN
950 PRINT AT 19,0: INK 4;Q$( TO
2); INK 3;Q$(3 TO 14); INK 4;Q$(
15 TO 18); INK 3;Q$(19 TO 30);
INK 4;Q$(31 TO 34); INK 3;Q$(35
TO 46); INK 4;Q$(47 TO 50); INK
3;Q$(51 TO 62); INK 4;Q$(63 TO )
960 RETURN
970 PRINT AT 20,13:"LEVEL ";H
980 FOR F=1 TO 200: NEXT F: CLS
990 RETURN

```



X



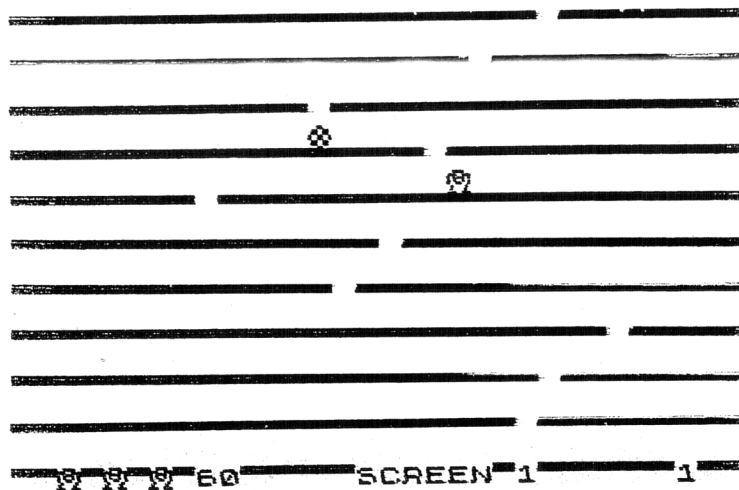


# Hi-Rize

From Paul Toland comes this exciting arcade game, which involves jumping through many screens of lines with moving gaps. There are five screen variations to be played through, and you can see them all in these screen printouts:

## HI-RIZE

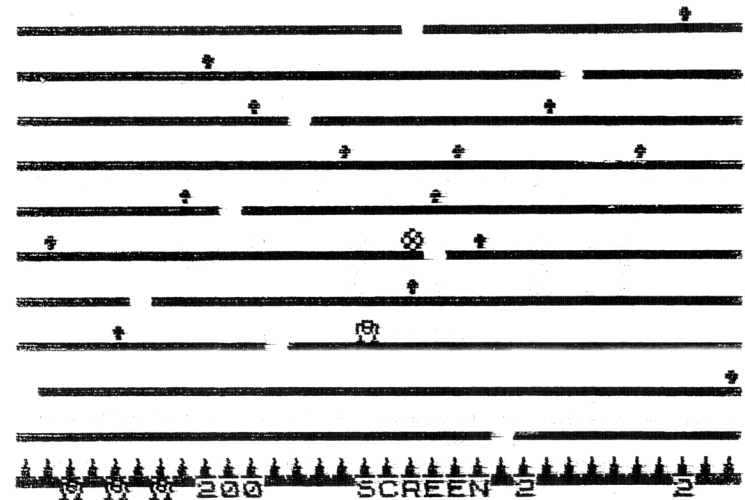
SCREEN 1



## HI-RIZE

SCREEN 2

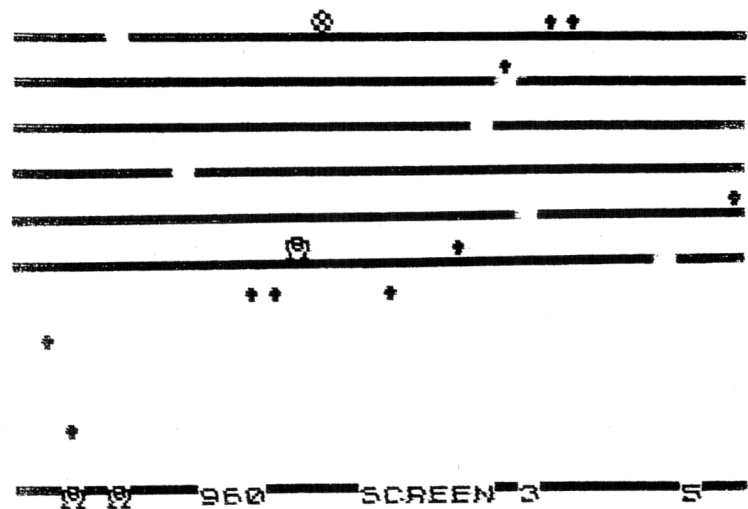
From this screen on, red dots appear on the screen and you can gain points by going over these. In this screen you may not return to the base line.



## HI-RIZE

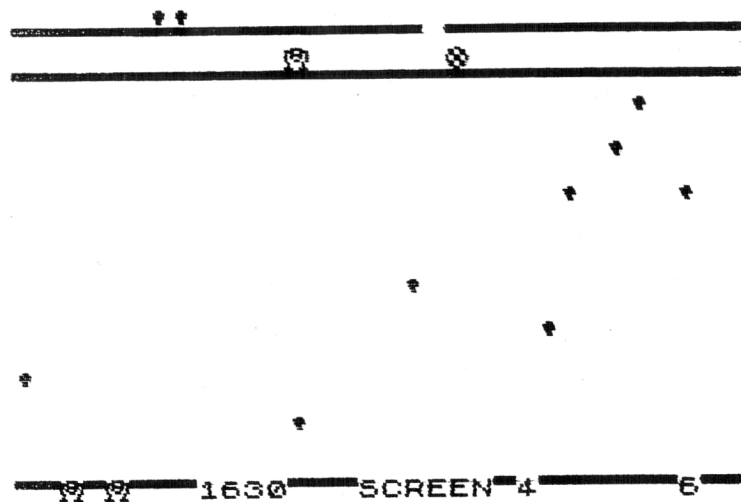
SCREEN 3

As you move up the screen the lines below you vanish, so you will lose a life if you fall.



# HI-RIZE

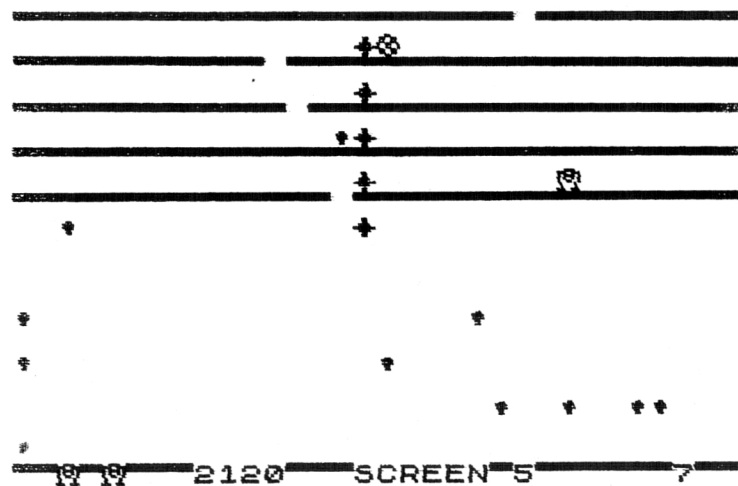
SCREEN 4



# HI-RIZE

SCREEN 5

Spikes appear at the top of the screen and their touch is lethal.



And here's the listing for HI-RIZE:

```

1 REM HI-RIZE © P. TOLAND
80 GO SUB 800: GO SUB 2500
90 LET TOT=TOT+1: LET UN=UN+2:
GO SUB 920
100 FOR I=H TO L: PRINT AT I#2-
1,H(I,1):CHR# 144: LET N=H(I,1)+
H(I,2): IF N<0 OR N>31 THEN LET
A=INT (RAND+.5): LET H(I,2)=A-(A=
0): LET N=(H(I,2)=-1)*31
110 PRINT AT I#2-1,N;" ": LET H
(I,1)=N: NEXT I

```

```

120 PRINT AT Y,X;" ": IF SCREEN#
# (Y+1,X)=" " THEN LET SC=SC-20#
G: PRINT AT 21,0;SC;CHR# 144: BE
EP .1,-20: LET Y=Y+2: LET C=146:
LET T=0: IF L<10 THEN LET L=L+1
: IF G>2 THEN GO TO 300
130 LET T=T+1: IF T=UN THEN LET
C=145: BEEP .05,30
140 IF C=146 THEN GO TO 155
145 LET I#=INKEY#: IF I#<>"1" T
HEN GO TO 150
146 IF SCREEN# (Y-1,X)=" " THEN
BEEP .05,0: LET C=146: LET T=0:
GO TO 160
147 IF Y<20 THEN LET L=L-1: IF
L<10 AND G>2 THEN PRINT AT Y+1,0
148 LET SC=SC+10#G: PRINT AT 21
0;SC;CHR# 144: BEEP .05,20: LET
Y=Y-2: LET X=X-H(Y/2+1,2)
150 LET X=X+(I#="0")-I#("5"):
LET Y=X+(X=-1)*30-(X=30)*30
154 IF SCREEN# (Y+1,X)=" " THEN
GO TO 120
155 LET N=ATTR (Y,X): IF N=50 T
HEN LET SC=SC+10#G: BEEP .05,5:
PRINT AT 21,0;SC;CHR# 144
157 IF N=40 OR N=51 THEN GO TO
300
160 IF G>6 THEN LET H=Y/2:Y/2-
0)
162 PRINT INK 0;AT Y,X;CHR# C:
IF Y=0 THEN LET G=G+1: RESTORE 2
000: FOR I=1 TO 10: READ N: BEEP
.1,N: NEXT I: GO TO 90
170 PRINT AT BY,BX;" ": LET BX=
BX+D: LET BX=BX+(BX=-1)*32-(BX=0
BY)*32: LET A#=SCREEN# (BY+1,BX):
IF SCREEN# (BY+1,BX-D)=" " OR A
#=" " THEN LET BY=BY+2: LET D=-H
(BY/2+1,2): IF BY>11 OR BY>Y THE
N LET BY=0: LET BX=16: LET D=-H(
1,2)
175 LET N=ATTR (BY,BX): IF N=40
THEN GO TO 310
177 IF N=51 THEN LET D=-D: LET
BX=BX+D
180 PRINT INK 0;AT BY,BX;CHR# 1
47
185 IF G=2 AND Y<20 THEN PRINT
INK 0;AT 20,0;N#
190 GO TO 100

```

```

300 FOR I=L*2 TO 20: PRINT AT I
,X;CHR# 146;AT I-1,X;" ": BEEP .
1,-20-I: NEXT I
310 BEEP 2,-10: LET M=M-1: IF M
=0 THEN INPUT " TRY AGAIN?";A#
: GO TO 1+(A#="N")*9998+(A#="0")
*9998
320 LET G=G-1+(G=1)
330 GO TO 90
340 RESTORE 600: LET T#="" : FOR
I=1 TO 53: READ N: LET T#=T#+CH
R# (N+120): NEXT I
350 DATA 0,14,13,0,7,11,0,12,0,
11,0,0,7,11,0,0,0,11,0,11,3,0,10
360 DATA 0,5,10,0,0,0,10,9,0,5,10,0,0,
10,0,0,11,0,0,10,5,0,13,14,0,0,0,
10,0,0,13,14,0,13,12,0,14,12,0
370 LET T#="" : +T#( TO 22)+
" : +T#(23 TO 42)+
" : +T#(43 TO )
380 RESTORE 900: FOR I=0 TO 55:
READ N: POKE USR "A"+I,N: NEXT
I
390 LET N=0: DIM B#(32): RANDOM
IZE: LET TOT=0: LET UN=5: LET S
C=0: LET M=4: LET G=1: RETURN
400 DATA 255,255,255,0,0,0,0,0
410 DATA 60,231,189,165,153,60,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
420 DATA 195,165,129,189,189,18
9,0,0,1,24
430 DATA 24,102,102,153,153,102
,102,24
440 DATA 0,0,24,60,60,24,24,0,0
450 DATA 16,16,16,56,56,56,254,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
460 DATA 24,24,60,255,60,24,24,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
470 LET N#="" : LET L#="" : FOR I
=1 TO 32: LET N#=N#+CHR# 149: LE
T L#=L#+CHR# 144: NEXT I
480 IF P#="Y" THEN GO SUB 2550
490 LET H=1: LET L=10: LET T=10
0: LET X=16: LET Y=20: LET C=145
4950 INK 1: PAPER 6: BORDER 6: C
LS
500 LET BX=16: LET BY=0
5070 IF G>1 THEN FOR I=1 TO 20:
PRINT INK 2;AT INT (RAND*10)*2,RAN
D*31;CHR# 148: NEXT I
510 IF G>4 THEN FOR I=2 TO 10 S
TEP 2: PRINT AT I,15; INK 3;CHR#
150: NEXT I

```

```

1000 DIM H(10,2)
1010 FOR I=1 TO 21 STEP 2: PRINT
  AT I,0;L$: NEXT I
1020 FOR I=1 TO 10: LET H(I,1)=I
  NT (RND*31): LET A=INT (RND+.5):
  LET H(I,2)=A-(A=0): NEXT I
1030 LET D=-H(1,2)
1040 PRINT AT 21,0;SC;AT 21,15;"
SCREEN";CHR$ 144;G;AT 21,29;TOT:
  FOR I=1 TO M-1: PRINT AT 21,I*2
  ;CHR$ 145: NEXT I
0000 RETURN
0010 DATA 2,2,2,14,14,14,11,11,1
1,7,11,7,2
0020 PAPER 1: INK 7: BORDER 1: C
L$: INPUT ;: PRINT AT 10,0;"Do
you want instructions?"
0030 LET I=0
0040 LET I#=INKEY#: PRINT AT 0,0
  INK I;T#: IF I#="" THEN LET I=
  H+1-(I=0)*0: GO TO 2510
0050 LET P#="N": IF I#="Y" OR I#
  THEN LET P#="Y"
0060 RETURN
0070 LET I=0: INK 7: PAPER 1: CL
  ; BORDER 1: INPUT ;: PRINT AT
  4,10;INK 6;"SCREEN ";G: GO SUB
  0*100+2500
0080 LET I#=INKEY#: PRINT AT 0,0
  INK I;T#: IF I#="" THEN LET I=
  H+1-(I=0)*0: GO TO 2550
0090 BEEP .5,10: RETURN
0100 PRINT AT 6,0;"Your aim is t
o reach the top of each of the
screens in order to advance t
o the next. You move up b
y jumping through moving gaps
the lines above whilst avoid
ing falling down the gap in t
he line you are standing on.
You also have to avoid the ba
rrel which rolls about on the
higher lines; you will lose a
life if it hits you."
0110 PRINT "Move left&right with
keys S&D, jump with key I": R
ETURN
0120 PRINT AT 6,0;"From this scr
een on, red dots appear on th
e screen and you can gain poi
nts by going over these."
0130 PRINT "In this screen you m
ay not return to the base

```

```

line.": RETURN
0140 PRINT AT 6,0;"As you move U
p the screen the lines below
you vanish, so you will lose a
life if you fall.": RETURN
0150 PRINT AT 10,0;"Things speed
up a bit from now on.": RETUR
N
0160 PRINT AT 10,0;"Spikes appea
r at the top of the screen and
their touch is lethal.": R
ETURN
0170 RETURN
0180 INK 0: PAPER 7: BORDER 7

```

# Wipe-Out

No collection of computer games would be complete without a 'city bomber' type program. This is a much stronger version of the game than many I've seen, and provides a real challenge to the player. Here it is in action (at the end of a run):



The scenario is familiar. You are in a space ship which is coming down. You do not have any means of ascending, so must blast a landing strip for yourself by dropping bombs on the buildings. Use the "H" key to release your bombs.

```

10 PAPER 0: BORDER 0: CLS
20 DATA 0,0,0,03,255,0,2,0,1,0
,7,255,255,63,255,2,08,0,0,0,255
,255,255,255,255,70,255,255,102,
255,255,2,0,64,71,116,39,120,2

```

```

30 FOR c=111 TO 116: FOR n=0 T
0 7
40 READ X: POKE USR CHR$ c+n,X
50 NEXT n: NEXT c
60 LET be=0
70 PRINT AT 9,5; INK 5; "***WIP
E-OUT***- H to drop bomb.
Wipe the city off the earth
with H-bombs to give yourse
lf a landing strip.";AT 15,6; BR
IGHT 1; INK 3;"DIFFICULTY ?(1 TO
7)"; INPUT D: LET D=7-D
80 PAPER 0: CLS : BORDER 0
90 LET s=-1
100 FOR a=5 TO 25
110 INK 1+INT (RND*7): PAPER 1+
INT (RND*7): LET q=11+INT (RND*9
)
120 PRINT AT q,a; PAPER 0;"■":
FOR b=q+1 TO 21: PRINT AT b,a;"■"
: NEXT b
130 NEXT a
140 FOR a=1 TO 21
150 PAPER 0: INK 1+INT (RND*7)
160 LET bp=(21-a)
170 FOR b=28 TO 1 STEP -1
180 PRINT AT a,b;"■"; BEEP .0
F, bp
190 FOR l=0 TO 4
200 LET s=s+2*(INKEY#="h")+(s=1
): GO SUB 320
210 NEXT l
220 IF SCREEN$ (a,b-1)<>" " THE
N GO TO 450
230 PRINT AT a,b;" "
240 NEXT b: PRINT AT a,1;" "
250 NEXT a
260 DATA 4,4,4,0
270 RESTORE 260: FOR a=1 TO 4:
READ X: BEEP .25*(1+a*(a=4)),X:
NEXT a
280 PRINT AT 21,0; FLASH 1; INK
3; PAPER 5;"SCORE:";be;"standby
for take-off"; PAUSE 300
290 PAPER 0: FOR a=20 TO 0 STEP
-1
300 PRINT AT a+1,a+1;" "":AT a,
a: INK 1+INT (RND*6);"■"
310 NEXT a: GO TO 70
320 IF a=21 THEN RETURN : IF s=
1 THEN LET a=a+1
330 IF s=1 THEN LET a=a+2-(a=20
)

```

```

340 IF s=1 THEN LET h=b+2: IF s
=1 THEN LET d=d/2
350 IF s<0 THEN RETURN
360 PRINT AT e-1,h;" ";AT e,h;"
*": BEEP .005,21-e: LET e=e+1
370 IF SCREEN# (e,h)<>" " THEN
GO TO 390
380 IF e<21 THEN RETURN : IF e>
21 THEN LET e=21: GO TO 400
390 LET be=be+10
400 IF e>21 THEN LET e=21
410 PRINT AT e-1,h; PAPER 0;" "
: FOR n=7 TO 0 STEP -1
420 PRINT AT e,h; INK n;"█": BE
EP .02,n-12
430 NEXT n
440 LET s=-1: LET d=d*2: RETURN

450 PRINT AT a,b-2;" "
460 DATA 0,0,0,-5,-5,-5,0
470 RESTORE 460: INK 2
480 FOR n=10 TO 90 STEP 10: PLO
T b#0,(21-a)#0: DRAW 20,n: NEXT
n
490 PRINT AT 11,10; PAPER 6; IN
K 1;"SCORE:";be
500 STOP
510 GO SUB 410: GO TO 510

```

## Birds

Written by Neil Pellinacci, this is an arcade-type game, in which you have to defend your planet from the lines of alien birdmen above.

Once by one the aliens swoop down on you, dropping large white crates in the process. The crates are used as bombs, but later in the game, the crates can smother your planet, completely covering it. If this happens, the game ends, no matter how many lives you have left. The crates have a third role, but you'll have to play the game to discover what it is.

You control your laser base using the "Z" and "X" keys, with the space key to fire. You score 50 points for each swooping birdman you hit. If you hit a birdman on the bottom row, you get just 25 points. However, a birdman hit here will not be killed, but merely stunned, and will soon return.

You start with three laser bases and lose one each time it is hit by a birdman or a crate. The number of bases is shown on the top row of the screen, together with your score (on the left) and the current high score. You gain an extra base when you've destroyed three attack waves.

At this point, you'll meet mutant aliens. These swoop, but may then rise, dropping more crates.

To give you something to aim at, you might like to know that the best score that Neil has achieved - and only after a great deal of practice - is 12,000.

The program is written entirely in BASIC with the exception of a single machine code routine located at 32500. Here is a disassembled listing of the code:

```

00500 LD HL,704
00502 LD DE,00500
00504 LD BC,02
00506 PUSH HL
00508 ADD HL,DE
0050A DEC HL
0050C CP 0
0050E OR 0
00510 LAR NZ,00500
00512 POP HL
00514 DEC HL
00516 LD A,H
00518 OR A
0051A LAR NZ,00500
0051C LD HL,00500
0051E LD B,00
00520 LD A,(HL)
00522 CP 0
00524 LAR NZ,00500
00526 DEC HL
00528 DJNZ 00500
0052A LD C,0
0052C RET
0052E LD (HL),70
00530 ADD HL,BC
00532 LD (HL),126
00534 OR 00512

```

And here is the key to the user-defined graphics:

```

BB = X
DD = X
FFG = X
HIJ = X
KL = X

```

After you have the program working, you may wish to relocate the code. If you do, remember to check lines 90, 317, 412 and 7020. The subroutine locates the position of each crate and moves it down one position. It also checks to see if the planet is completely covered by crates. The result of this test is detected on return to BASIC by line 317.

You should type the program in, and then save it, before trying to run it, in case you've made a mistake with the machine code. It is also advisable to save the program so that it starts automatically on loading.

```

10 RANDOMIZE : GO TO 90
20 PRINT AT 0,6-LEN STR# 5; IN
K 5;5: RETURN
30 LET BC=BC+(INKEY#="x" AND B
C<27)-(INKEY#="z" AND BC>0): PRI
NT AT 21,BC; PAPER 8; INK 6;
40 IF INKEY#<>" " THEN RETURN
50 GO TO 520
60 FOR X=1 TO 3: FOR Y=1 TO 10
70 IF L(X,Y)<>-1 THEN LET L1=L
(X,Y): LET C1=C(X,Y): LET C2=Y:
LET L2=X: RETURN
80 NEXT Y: NEXT X: LET UF=1: R
ETURN

```

```

80 FOR X=20 TO 7+INT (RND*10)
STEP -1: PRINT AT X,BC+8: PAPER
7;" " : BEEP .001,40-X: PRINT AT
X,BC+2: PAPER 0;" " : NEXT X: PRI
NT AT X,BC+2: INK 5: PAPER 7;" "
: RETURN
90 CLEAR 32499
95 BORDER 6: PAPER 0: CLS : 60
RDER 0
100 INK 4: BRIGHT 1: PRINT AT 0
,25: INK 5;"000000"
105 GO SUB 7000: GO SUB 9000
107 LET HS=0
110 GO SUB 3000: LET BC=13: LET
S=0: LET LB=3: LET UW=0
120 LET FF=0
130 LET HF=0
140 LET BFF=0
150 LET WF=0
200 DIM L(3,10): DIM C(3,10)
205 FOR B=0 TO 2
210 FOR A=0 TO 9
220 LET L(B+1,A+1)=B*2+1: LET C
(B+1,A+1)=A*3+1
230 NEXT A
235 NEXT B
237 PRINT AT 0,0: INK 5;"000000
";AT 0,17:LB
240 FOR B=1 TO 2
250 FOR A=1 TO 10: PRINT AT L(B
,A),C(B,A);"A": NEXT A
255 NEXT B
260 LET BH=0
290 FOR K=1 TO INT (RND*15+1)
300 LET B=INT (RND*3+1): LET A=
INT (RND*10+1)
305 IF L(B,A)=-1 THEN GO TO 315
310 PRINT AT L(B,A),C(B,A);"Y"
: BEEP .01,-10: PRINT AT L(B,A),
C(B,A);"A": BEEP .01,5
312 IF RND>.6 THEN IF ATTR (22,
BC+2)=126 THEN GO SUB 80
315 IF INKEY$("<>") THEN GO SUB 3
0
317 IF NOT USR 32500 THEN GO TO
4000
319 IF ATTR (21,BC+1)=126 OR AT
TR (21,BC+2)=126 OR ATTR (21,BC+
3)=126 THEN GO TO 1000
320 NEXT K

```

```

330 IF BH>20 THEN GO SUB 50: IF
WF=0 THEN GO TO 405
340 IF WF=1 THEN GO TO 2000
400 LET L2=INT (RND*3+1): LET C
2=INT (RND*10+1)
402 LET L1=L(L2,C2): LET C1=C(L
2,C2)
404 IF L1=-1 THEN GO TO 400
405 PRINT AT L1,C1;" "
406 LET CD=INT (RND*3-1)
410 PRINT AT L1,C1: OVER 1;"A"
: BEEP .001,L1+20: PRINT OVER 1:
AT L1,C1;"A"
411 IF RND>.6 THEN IF L1>6 THEN
PRINT AT L1+1,C1: INK 6: PAPER
7;" "
412 LET ZZ=USR 32500: IF INKEY$
("<>") THEN GO SUB 30: IF HF=1 THE
N LET HF=0: GO TO 290
413 IF ATTR (21,BC+1)=126 OR AT
TR (21,BC+2)=126 OR ATTR (21,BC+
3)=126 THEN GO TO 1000
415 LET C1=C1+CD: IF C1=32 THEN
LET C1=0
416 IF C1=-1 THEN LET C1=30
418 IF RND>.65 THEN LET CD=INT
(RND*3-1)
419 IF UW>=3 THEN IF RND>.6 THE
N IF L1=0 THEN LET L1=L1-2
420 LET L1=L1+1: IF L1<>21 THEN
GO TO 410
425 IF C1=BC+1 OR C1=BC+2 THEN
LET BFF=1: GO TO 1000
430 PRINT AT L(L2,C2),C(L2,C2);
"Y": GO TO 290
520 PLOT BC*8+20,6: DRAW INK 3,
PAPER 0,0,126: PLOT OVER 1,BC*8
+20,6: DRAW OVER 1,0,126
530 IF BC+2=C1 OR BC+1=C1 THEN
GO TO 510
540 IF SCREEN$ (5,BC+2)("<>") TH
EN PRINT AT 5,BC+1;" " : LET S=
S+25: GO SUB 20
600 RETURN
610 PRINT AT L1,C1: INK 7;"A":
AT L1,C1;" " : LET C1=-1: LET L1
=-1: BEEP .05,-30: LET L(L2,C2)=
-1: LET S=S+50: GO SUB 20
620 LET HF=1: LET BH=BH+1: RETU
RN

```



```

1000 REM BANG
1010 FOR Y=1 TO 4: FOR Z=1 TO 10
: BEEP .015,20-Z: PRINT AT 21,60
: INK RND*7;" ": NEXT Z
1020 FOR Z=10 TO 1 STEP -1: BEEP
: .015,20-Z: NEXT Z: NEXT Y
1030 LET LB=LB-1: PRINT AT 0,17;
: INK 5;LB
1040 IF LB=0 THEN GO TO 1060
1045 IF BFF=1 THEN LET BFF=0: GO
: TO 430
1050 GO TO 415
1060 PRINT AT 0,17; INK 1; PAPER
: 7; FLASH 1;LB
1070 FOR X=1 TO 20: PRINT AT X,0
: "
: " : NEXT X
1080 BEEP .2,50: BEEP .25,30
1090 FOR X=1 TO 7: PRINT AT X+7,
: 0; INK X; " GAME OVER
: GAME OVER " : NEXT X
1100 BEEP .2,50: BEEP .25,30
1270 IF S>HS THEN LET HS=S
1280 PRINT AT 0,32-LEN STR$ HS;
: INK 5;HS
1290 IF S=HS THEN FOR Z=1 TO 5:
: FOR X=20 TO -20 STEP -2: BEEP .0
: 1,X: NEXT X: NEXT Z
1300 FOR X=0 TO 20: BEEP .005,20
: NEXT X
1305 FOR X=0 TO 20: BEEP .005,0:
: NEXT X
1310 FOR X=1 TO 300: NEXT X: BOR
: DER 6: CLS : BORDER 0: PRINT AT
: 0,26; INK 5;"000000":AT 0,32-LEN
: STR$ HS; INK 5;HS: GO TO 110
1400 STOP
2000 REM WAVE OVER
2010 FOR X=1 TO 20: PRINT AT X,0
: "
: " : NEXT X
2020 FOR X=1 TO 7: PRINT AT 3+X*
: 0,2; INK X;" ATTACK WAVE DESTR
: OYED"
2030 NEXT X
2040 PAUSE 150: BEEP .2,50: FOR
: X=-30 TO 35: BEEP .01,X: BEEP .0
: 15,20: NEXT X: BEEP .2,50
2050 FOR X=1 TO 20: PRINT AT X,0
: "
: " : NEXT X

```

```

2060 LET UU=UU+1: IF UU=3 THEN 5
: ORDER 6: PAPER 0: CLS : BORDER 0
: PRINT AT 0,26; INK 5;"000000";
: AT 0,32-LEN STR$ HS; INK 5;HS: B
: EEP .2,10: BEEP .2,-10: LET LB=L
: B+1
2070 GO TO 120
3000 REM TITLE PAGE
3010 RESTORE 3500
3020 READ X: IF X=999 THEN GO TO
: 3100
3040 READ Y: PLOT X,Y
3050 READ X: IF X=-999 THEN GO T
: O 3020
3060 READ Y: DRAW X,Y: GO TO 305
: 0
3100 PRINT AT 10,4; INK 5;"@ NEI
: L PELLINACCI 1985"
3110 PRINT AT 13,7; INK 2; PAPER
: 7;"C O N T R O L S"
3120 PRINT AT 15,2; INK 6;"Z....
: .....LEFT"
3130 PRINT AT 17,2; INK 6;"X....
: .....RIGHT"
3140 PRINT AT 19,2; INK 6;"(SPAC
: E).....FIRE"
3150 PRINT AT 21,5; INK 2; PAPER
: 7;"PRESS ANY KEY TO PLAY"
3160 INK 7: PLOT 0,0: DRAW 0,55:
: DRAW 53,0: PLOT 175,70: DRAW 77
: 0: DRAW 0,-58: DRAW -45,0: PLOT
: 0,2: DRAW 37,0
3165 INK 2: PLOT 0,53: DRAW 255,
: 0: DRAW 0,16: DRAW -255,0: DRAW
: 0,-16
3170 INK 4
3200 FOR X=30 TO -30 STEP -1: BE
: EP .01,X: BEEP .01,-X: NEXT X: P
: AUSE 0
3210 FOR X=1 TO 21: PRINT AT X,0
: "
: " : NEXT X
3220 RETURN
3400 RETURN
3500 DATA 35,110,0,55,30,-14,-30
: -14,30,-14,-30,-14,-999
3510 DATA 65,110,0,55,-999
3520 DATA 105,110,0,55,30,-14,-3
: 0,-14,30,-28,-999
3530 DATA 151,110,0,55,30,-28,-3
: 0,-28,-999

```

```

3540 DATA 169,110,30,25,-30,0,30
.25,-999
3700 DATA 999
4000 REM LANDED
4010 BEEP .2,3: BEEP .15,-1: BEE
P .25,5
4020 FOR Z=1 TO 20
4030 FOR X=1 TO 3: FOR Y=1 TO 10
4040 IF L(X,Y)<0 OR L(X,Y)>20 TH
EN GO TO 4055
4050 PRINT AT L(X,Y),C(X,Y): INK
3:" ": LET L(X,Y)=L(X,Y)+1: PR
INT AT L(X,Y),C(X,Y): INK 3:"A"
4060 NEXT Y: NEXT X
4070 NEXT Z
4080 FOR Z=20 TO -30 STEP -3: BE
EP .01,Z: BEEP .01,Z*1.5: BEEP .
01,-Z+10: NEXT Z
4090 BEEP .15,-5: BEEP .15,-10:
BEEP .15,1
4095 FOR A=1 TO 5
4100 FOR Z=1 TO 19 STEP 2: PRINT
AT Z,0: INK RND*6+1:" ": BEEP .0
RDS HAVE LANDED
1,Z*2: NEXT Z
4105 NEXT A
4200 BEEP 2,-10
4300 GO TO 1000
4900 STOP
7000 REM #/C
7020 RESTORE 7030: FOR A=1 TO 46
: READ B: POKE 32499+A,B: NEXT A
7030 DATA 33,199,0,17,0,90,1,32,
0,229,25,43,126,254,126,40,22
7040 DATA 205,43,104,101,32,242
7050 DATA 36,200,90,0,32,126,254
,126,32,5,43,10,240,24,0
7060 DATA 201,54,70,0,54,126,24,
227
8000 RETURN
9000 FOR C=0 TO 95: READ B: POKE
USR "A"+C,B: NEXT C
9010 DATA 100,119,20,7,1,2,2,0
9020 DATA 3,140,66,224,100,64,64
,0
9030 DATA 0,1,0,7,25,114,105,0
9040 DATA 0,120,64,224,152,70,35
,0
9050 DATA 0,0,0,0,4,14,31,255

```

```

9060 DATA 0,0,26,20,52,127,255,0
,0
9070 DATA 0,0,0,0,16,56,250,255
9080 DATA 0,0,0,0,16,0,0,255
9090 DATA 0,0,42,126,42,201,54,0
,0
9100 DATA 0,32,0,126,36,126,0,25
,0
9110 DATA 129,72,2,16,130,0,64,1
,33
9120 DATA 129,16,64,0,64,16,2,16
,1
9300 RETURN

```

We'll now have a look at what each part of the program does:

10: this skips over the following subroutines, which are at the beginning for maximum speed

20: subroutine to print score

30: this moves the laser base. If the space key is pressed, control is sent to line 520

50: this is the beginning of a subroutine which is called when only a few birds are left. It locates the first living birdman, and selects it to swoop down. If there are no birds left, WF is set to 1 to tell the main loop what has happened

80: an 'additional hazard' subroutine

90: makes room for the machine code

110: game initialisation  
120 - 160: wave initialisation  
200 - 235: sets up the birds  
240 - 255: display birds  
290 - 320: makes a few birds flap at random  
317: checks for a whole line of crates; also scrolls them down  
318: checks for a collision between the base and a crate  
330: selects next bird without using the random selection routine, see subroutine 50  
340: checks for end of wave  
400 - 404: selects a bird at random, and sees if it exists  
410 - 420: the main loop  
411: drops a crate  
419: if a bird is a mutant, it may move upward  
520: fire routine

610 - 620: exploding bird  
1000 - 1050: base hit routine  
1060 - 1310: game over routine  
2000 - 2070: 'attack wave over' routine  
2060: start of mutant waves  
3000 - 3210: title page routine  
3500 - 3700: data for title page  
4000 - 4300: this routine is called when the planet has been completely covered by crates; it causes the birdmen to change color and slowly land  
7020: POKEs the machine code  
7030 - 7060: data for machine code  
9000: sets up UDG characters  
9010 - 9120: data for graphics characters  
Finally, here is a list of the main variables used in the program:  
HS - high score  
BC - base position  
S - score

LB - bases left  
VW - wave number  
HF - hit flag (1 - yes, 0 - no)  
WF - wave over flag (1 - yes, 0 - no)  
L(3,10), C(3,10) - positions of the 30  
birdmen  
BH - number of birdmen hit  
L1,C1 - position of swooping birdman  
CD - horizontal direction of birdman

As with most programs in this book, there are many things which can be changed to suit your own needs. For instance, the random numbers can be changed, thus changing the probability of various events taking place (for example, lines 411 and 419). The screen displays can also be changed at various stages.

## Chopper Squad

Action is the name of this game from Malcolm Young. You are in charge of a rescue helicopter flying above the desert.

You have to save as many soldiers as you can before you run out of fuel. You can either pick up a soldier by landing on him for 50 points, or for a faster and easier pickup, use the helicopter skyhook by pressing "1" when you are exactly two spaces above the man. This will get you 25 points.

You can also use the skyhook for clearing trees. However, in doing so you reduce the helicopter's cover. This cover is important as it helps shield you from the people who come along in jeeps from time to time to shoot at you. They can only shoot straight up, and you can destroy the jeeps by simply dropping bombs on them. Hardly seems fair, does it. You can only bomb them while in the sky, and you do so by pressing "1" again. (The program will know whether you want the bomb or the skyhook, which is why "1" is used to trigger both.)

After you pick up a man, the ambulance turns black and you can then gain a bonus of 100 points if you can get the man to the black ambulance.

To add to the things you have to keep track of in this program, you'll see (from time to time) a fuel truck somewhere along the bottom of the screen. If you manage to bomb this, you'll gain 20 fuel units as well as 10 points.

CHOPPER SQUAD ends when you either run out of fuel or when your lives have all been used up. There is a 'hall of fame' at the end of the program, which you can easily extend to five places.

```

100 GO SUB 550
110 REM a=# b=# c=# d=# e=# f=#
120 n=1 j=0 l=0 m=y d=0 o=#
130 DIM s$(5,10): DIM h(5)
140 LET tr=0: LET sf=0: LET dp=0
150 LET sc=0: LET pk=0: LET lives=0
160 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS
170 LET Jf=0: LET ln=100: LET f
=0: LET s=0
180 BORDER 6: PAPER 5: INK 0: C
190 LET f$="": LET r$="": L
ET a$="": LET b$="": LET p$="
": LET t$="
140 FOR g=11 TO 21: PRINT PAPER
0: AT g,0:
NEXT g
150 FOR t=0 TO 31: PAPER 8: LET
r=INT (RND*4): PRINT INK 2: AT 1
0-t, t: "Y": INK 4: AT 15-t, t: "X":
NEXT t
160 LET y=INT (RND*10)+1: LET x
=INT (RND*30): LET a=1
170 PRINT PAPER 6: AT 11, a-1: "
": INK 2-2*pk: AT 11, a: a$
175 LET fuel=fuel-1: PRINT PAPE
R 0: INK 7: BRIGHT 1: AT 0,0: "FUE
L="; fuel; " : AT 0,10: "SCORE="; sc
: AT 0,20: "LIVES="; lives

```

```

178 IF fuel<0 THEN GO TO 1000
180 LET a=a+0.5 AND (a<30): IF
NOT a THEN LET pk=a
190 PRINT OVER 0: AT y,x: " "
191 IF y=11 AND x+1=a THEN LET
sc=sc+50: LET pk=0: BEEP .5,-5
200 LET y=y+(INKEY$="6")-(INKEY
$="7"): LET x=x+(INKEY$="8")-(IN
KEY$="5")
205 IF y<11 THEN LET x=x+(INKEY
$="0")-(INKEY$="5")
210 LET y=y+(y<0)-(y>21): LET x
=x+(x<0)-(x>30)
220 LET c=ATTR (y,x): IF c<>40
AND c<>48 THEN GO TO 1000
230 LET c=ATTR (y,x+1): IF c<>4
0 AND c<>48 THEN GO TO 1000
240 PRINT OVER 0: AT y,x: h$: BEE
P .1,-10: PRINT OVER 1: AT y,x: r$
245 IF AND<.3 AND NOT pk THEN L
ET sf=1
248 IF sf THEN GO SUB 420+dp
250 IF RND<.5 THEN LET Jf=1
255 GO TO Jf*ln+170

```

```

270 LET Jy=16+INT (RND*4)+1
280 LET Jd=150
290 LET Jd=RND: LET Jd=(Jd+.5)-
(Jd*.5)
300 LET q#=p# AND Jd=1: LET q#=
q#+(J# AND Jd=-1): IF RND<.333 T
HEN LET tr=1
305 IF tr=1 THEN LET q#=(t$(2 T
0) AND Jd=1)+(t$( TO 2) AND Jd=
-1)
310 LET Jx=30 AND Jd=-1
320 PRINT INK 0: AT Jy,0: "
325 PRINT INK 1: AT Jy,Jx: q#
330 LET Jx=Jx+Jd

```

```

340 IF Jx=0 OR Jx=31 THEN LET l
n=100: LET tc=0: LET Jf=0: PRINT
PAPER 6: AT Jy,0: " : GO TO 170
350 IF y<10 AND INKEY$="1" THEN
GO TO 510
360 IF tr=1 OR NOT (Jx>=INT x-1
AND Jx<=x+1) THEN GO TO 170

```

```

370 LET e=y+1: FOR f=jy-1 TO e
STEP -1: LET b=ATTR (f,jx+1): IF
b=40 OR b=48 THEN PRINT PAPER 8
INK 2;AT f,jx+1;": BEEP .1,f
PRINT INK 0;AT f,jx+1;": NEX
f
380 PRINT INK 0;AT f,jx+1;": "
390 IF (jx+1=x+1 OR jx+1=x) AND
f=y THEN BEEP 1,-5: GO TO 1000
400 GO TO 170
410 REM Transport soldier
420 LET sdx=INT (RND*30)+1: LET
sdy=12+INT (RND*5): LET dp=10:
LET t=INT (RND*20)+10
430 PRINT INK 1;AT sdy,sdx;"X":
BEEP .15,0
440 LET t=t-1: PRINT AT sdy,sdx
": IF NOT t THEN LET sf=0: LE
T dp=0: RETURN
450 IF y=sdy AND x+1=sdx THEN L
ET sc=sc+50: BEEP 1,5: LET sf=0:
LET dp=0: LET pk=1: RETURN
460 IF INKEY#<>"1" THEN RETURN
470 PRINT AT y+1,x+1;"↓";AT y+2
,x+1;"↓"
480 IF sdy=y+2 AND sdx=x+1 THEN
FOR n=y+2 TO y STEP -1: PRINT A
T n,x+1;"X": BEEP .3,0: PRINT AT
n,x+1;": NEXT n: LET pk=1: LE
T sf=0: LET dp=0: LET sc=sc+25
490 PRINT AT y+1,x+1;": ";AT y+2
,x+1;": "
500 RETURN
510 LET e=jy: FOR f=y+1 TO e: L
ET b=ATTR (f,x+1): IF b=40 OR b=
48 THEN PRINT PAPER 8; INK 2;AT
f,x+1;"X";CHR$ 8: BEEP .1,30-f:
PRINT INK 0;AT f,x+1;": NEXT f
: GO TO 530
520 PRINT INK 0;AT f,x+1;": "
530 IF b=49 THEN BEEP 1,-5: LET
sc=sc+10: LET fuel=fuel+20*(tr=
1): PRINT AT jy,jx;": ": LET j
x=0: GO TO 340
540 GO TO 170
550 REM
570 RESTORE
580 DATA "a",1,1,1,255,255,195,
,219,24
590 DATA "b",15,124,6,63,63,195
,219,24

```

```

600 DATA "c",128,128,128,255,25
6,195,219,24
610 DATA "d",48,12,10,252,252,1
25,219,24
620 DATA "e",0,16,124,254,255,1
27,36,125
630 DATA "f",0,0,192,199,255,1,
0,0
640 DATA "g",255,0,0,0,0,0,0
650 DATA "h",7,0,0,0,0,0,0
660 DATA "i",126,97,97,255,255,
195,219,24
670 DATA "j",255,239,199,239,25
6,195,219,24
680 DATA "k",16,16,16,16,16,16,
04,56
690 DATA "l",138,73,118,188,114
,92,89,178
700 DATA "m",36,36,24,16,16,16,
16,16
710 DATA "n",153,90,78,56,8,20,
25,94
720 DATA "o",231,126,24,24,60,1
0,50,24
730 DATA "p",126,255,255,255,25
6,195,90,24
740 FOR c=1 TO 16: READ c$
750 FOR e=0 TO 7: READ n: POKE
1000+c*8,n: NEXT e
760 NEXT c
770 RETURN
1000 PRINT INK 2; FLASH 1;AT y,x
": BEEP 1,1
1010 PRINT INK 0; FLASH 0; PAPER
8;AT y,x;": LET y=INT (RND*1
0)+1: LET x=INT (RND*30)
1020 LET lives=lives-1
1030 IF fuel<1 THEN PRINT FLASH
1;"OUT OF FUEL! I'M AFRAID THIS
IS THE END": GO TO 1040
1040 IF lives THEN GO TO 170
1040 IF sc<h(5) THEN GO TO 1200
1050 INPUT "Please enter your na
me for the score table",n$
1060 FOR r=5 TO 1 STEP -1: BEEP
.1,r: IF sc>h(r) THEN NEXT r
1070 FOR g=5 TO r+2 STEP -1: BEE
P .1,g: LET h(g)=h(g-1): LET s$(
g)=s$(g-1): NEXT g

```

```

1080 LET h(g)=sc: LET s#(g)=n#
1090 BORDER 1: PAPER 2: INK 7: B
RIGHT 1: CLS
1110 PRINT INK 5;AT 3,2:"TODAYS
CHAMPIONS ARE-"
1120 FOR a=1 TO 5: BEEP .5,a: PR
INT INK 0;AT 7+a,8;s#(a);" ";h(a
): NEXT a
1200 PRINT FLASH 1: PAPER 4: INK
0;AT 15,3:"DO YOU WANT TO PLAY
AGAIN?"
1210 IF INKEY#="n" OR INKEY#="N"
THEN GO TO 1240
1220 IF INKEY#="" THEN GO TO 121
0
1230 GO TO 30
1240 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: STOP

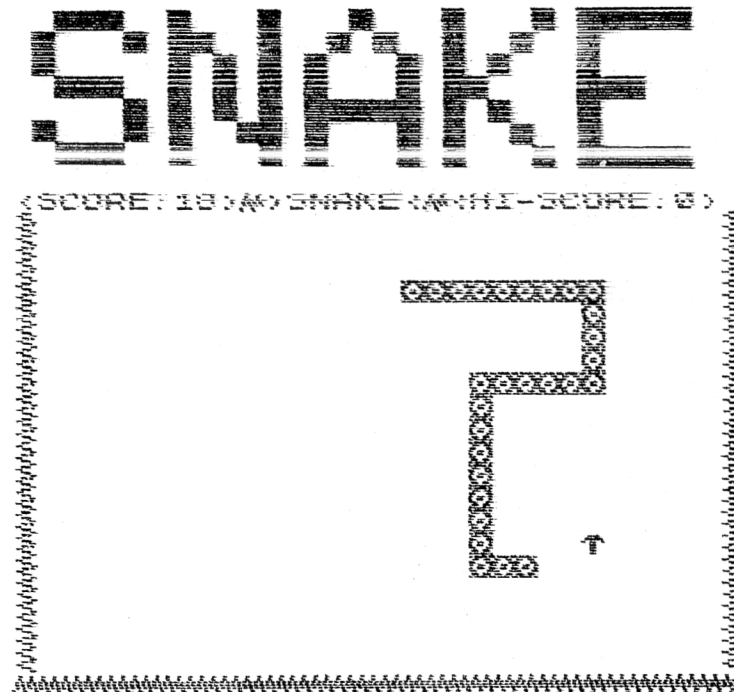
```

# Snake

Written by David Perry, SNAKE is a game demanding a steady eye and concentration.

The object of the game is to lead your snake around a garden - surrounded by an electric fence - while getting the snake to eat mushrooms. Each mushroom makes the snake grow longer, and you must make sure you do not hit the snake's tail...or the game will end.

Here's what it looks like in action:









```

620 LET C$=STR$ X
630 IF X<10 THEN LET C$=" "+STR
# X
640 LET D$=STR$ Y
650 IF Y<10 THEN LET D$=" "+STR
# Y
660 LET X1=VAL A$(1 TO 2)
670 LET Y1=VAL A$(3 TO 4)
680 IF ATTR (X1,Y1)<>122 THEN G
O TO 700
690 PRINT AT X1,Y1;" "
695 REM ***Print snake
700 PRINT BRIGHT 1; INK 2; AT X,
Y;"S"
710 LET A$=A$+C$+D$
720 IF A=0 THEN LET A$=A$(5 TO
)
730 LET A=A-(A>0)
740 GO TO 540
750 LET M=ATTR (X,Y)
760 IF M<>60 THEN RETURN
770 LET A=A+L
780 LET SC=SC+L
790 PRINT AT 0,0; INK 5;"<SCORE
";SC;">"; INK 1;"M"; INK 2;"SN
AKE"; INK 1;"M"; INK 4;"<HI-SCO
RE";HS;">" AND HS<100)
800 PRINT PAPER 7; INK 4; AT INT
(AND#10)+2, INT (AND#20)+2;"T"
810 FOR n=50 TO -20 STEP -10: B
EEP .01,n: NEXT n
820 RETURN
830 LET X=X-(X=22)+(X=-1)
840 LET Y=Y-(Y=32)+(Y=-1)
850 FOR F=50 TO -20 STEP -5: B
EEP .02,F: NEXT F
855 REM ***End of game
860 IF SC>HS THEN LET HS=SC
870 PRINT AT 10,10;"HIGH SCORE:
";HS
880 PRINT AT 12,11;"Press a key"
"
890 LET I=0
900 PRINT AT X,Y; INK I;"S": B
EEP .01,I#2
905 REM ***Print scores
910 PRINT AT 8,6; INK I; FLASH
1;">You scored ";SC;" points.<"
920 LET Q$=INKEY$: IF Q$="" THE
N LET I=I+1: IF I>7 THEN LET I=0
930 IF Q$="" THEN GO TO 900
940 CLS
950 GO TO 30

```

## Dodgems

This fast-moving action game is by Raymond Blake. You steer your car around a track with four lanes, passing over flags to gain points. At the same time, you are trying to avoid the suicidal car controlled by the computer, which travels in the opposite direction to you, trying to spoil your fun.

You can change lanes in the four gaps in the track, and you can change tracks as many times as you like. The suicide car can also change lanes, but it is limited to moving a single lane at a time. Despite this, you'll find this a very difficult game.

Near the start of a run, you'll gain points for red flags, which turn yellow as you pass over them. When you've run out of red flags (or you just feel like a change) passing over the magenta flag will allow you to gain points from yellow flags, but will also allow your Kamikaze opponent a chance to catch up!

Another magenta flag will appear in another part of the track, and passing over it will once again change the color of the flags which allow you to gain points.

```

20 FOR i=1 TO 5: READ z$: FOR
j=0 TO 7: READ z: POKE USR z#+j,
z: NEXT j: NEXT i

```



```

000 LET B#="0": PRINT B#="0": LET
B#="0": LET B#="0":
000 IF (C#="0") THEN GO TO 000
010 IF SCREEN# (X+(B#="0")-(B#="7"),
C+(B#="0")-(B#="7"))# THEN
HEN GO TO 400
020 IF B#="0" THEN LET B#="0":
LET B#="0": GO TO 400
030 IF B#="0" THEN LET B#="0":
LET B#="0": GO TO 400
040 IF B#="7" THEN LET B#="5":
LET B#="0" GO TO 400
050 IF B#="0" THEN LET B#="7":
LET B#="0":
401 IF (X>0 AND X<10) OR (Y>10
AND Y<10) THEN PRINT AT X,Y: INK
C1: GO SUB 900: GO TO 420
400 LET B=ATTR (X+(B#="0")-(B#="7"),
C+(B#="0")-(B#="7"))
400 PRINT AT X,Y: INK B:
405 IF B=100 THEN GO SUB 050
407 IF B=64+C1 THEN LET B=B+C1
410 LET X=X+(B#="0")-(B#="7"):
LET Y=Y+(B#="0")-(B#="7"):
420 PRINT AT X,Y: INK B:
430 PRINT AT 10,17: GO: BEEP .00
1.00
450 IF X=X1 AND Y=Y1 THEN GO TO
000
500 IF FLAG=0 THEN LET FLAG=10
510 IF SCREEN# (X1+(B#="0")-(B#="7"),
Y1+(B#="0")-(B#="7"))# THEN
GO TO 000
520 IF B#="0" THEN LET B#="7":
LET B#="0": GO TO 600
530 IF B#="0" THEN LET B#="5":
LET B#="0": GO TO 600
540 IF B#="7" THEN LET B#="0":
LET B#="0": GO TO 600
550 IF B#="0" THEN LET B#="0":
LET B#="0": GO TO 600
560 IF (X>0 AND X<10) OR (Y>10
AND Y<10) THEN PRINT AT X,Y:
GO SUB 900: GO TO 610
601 IF (X1=0 AND B#="0") OR (Y1
=14 AND B#="0") OR (X1=10 AND B#
="7") OR (Y1=17 AND B#="5") THEN
PRINT AT X1,Y1: INK C1: GO
SUB 700

```

```

600 IF (X1=0 AND X1<10) OR (Y1
=10 AND Y1<10) THEN PRINT AT X1,Y1:
INK C1: GO TO 610
600 LET B=ATTR (X1+(B#="0")-(B#="7"),
Y1+(B#="0")-(B#="7"))
600 PRINT AT X1,Y1: INK (Z1-64
)-(100 AND Y1-64)000: FLASH 1 AN
D Z1-64)0:
600 LET Z1=Z
610 LET X1=X1+(B#="0")-(B#="7")
LET Y1=Y1+(B#="0")-(B#="7")
600 PRINT AT X1,Y1: INK B:
600 IF X=X1 AND Y=Y1 THEN GO TO
000
600 GO TO 000
700 IF LANE=LANE1 THEN RETURN
700 PRINT AT X1,Y1: INK C1:
700 LET L=L+SGN (L-1000): L
ET LANE1=LANE1+L/2
750 LET X1=X1+L*(B#="0")-(B#="5")
750 LET Y1=Y1+L*(B#="7")-(B#="0")
700 RETURN
800 FOR I=25 TO -10 STEP -1: BE
EP .001: NEXT I
810 PRINT AT 14,0: PAPER 4: INK
0: FLASH 1: "PRESS ANY KEY FOR A
NEW GAME"
815 IF SC=SC THEN LET SC=SC
820 IF INKEY#="" THEN GO TO 82
0
830 IF INKEY#="" THEN GO TO 830
840 GO TO 05
850 LET C=C-C: LET C1=C-C1: LET
SC=SC-10: BEEP .01,30: BEEP .01
.00
855 LET FLAG=10
860 LET SC=SC+10
870 LET LIN=INT (RND*20)+1: LET
COL=INT (RND*30)+1: IF CODE SCR
EEN# (LIN,COL)<>0 THEN GO TO 870
870 PRINT AT LIN,COL: INK 0: FL
ASH 1: "4"
880 RETURN
900 LET I#=INKEY#: IF Y>10 AND
Y<10 THEN GO TO 950
905 IF Y>22 THEN GO TO 925
910 IF I#="5" AND Y>2 THEN LET
Y=Y-2: LET LANE=LANE-1
915 IF I#="0" AND Y<0 THEN LET
Y=Y+2: LET LANE=LANE+1

```

```

0000 RETURN
0001 IF I#="0" AND X>00 THEN LET
0002 -I#="1" LET LABE=LABE+1
0003 IF I#="0" AND X<00 THEN LET
0004 +I#="1" LET LABE=LABE-1
0005 RETURN
0006 IF X>10 THEN GO TO 075
0007 IF I#="0" AND X<7 THEN LET
X=X+10: LET LABE=LABE+1
0009 IF I#="7" AND X>11 THEN LET
X=X-10: LET LABE=LABE-1
0011 RETURN
0012 IF I#="0" AND X<20 THEN LET
X=X+10: LET LABE=LABE-1
0014 IF I#="7" AND X>14 THEN LET
X=X-10: LET LABE=LABE+1
0016 RETURN
0017 DATA "a",10,50,104,50,10,4,
0018 "b",04,90,106,90,04,15
0019 "c",004,70,60,055,055,
0020 "d",100,055,100,04,90,
0021 "e",7,114,06,055,055,0
0022 IF I#>0 THEN LET I#&=I#
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100

```

## Zombies

Zombies, from David Perry, is a game in which you must lure the Zombies into swamps. They are blind, and always move towards you.

Full instructions are provided within the program. Use the cursor keys to move around. Note that the pound signs in this listing should be entered as hash (#) signs.

```

1 REM
2 REM
3 REM      > ZOMBIES <
4 REM      =====
5 REM
6 REM      COPYRIGHT
7 REM      DAVID PERRY
8 REM      1983!
9 REM
10 REM
11 LET NUM=10: DIM N(NUM+1): D
IM N$(NUM+1,8): FOR N=1 TO NUM:
LET N(N)=1100-(N*100): LET N$(N)
="SPECTRUM": NEXT N
12 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS
13 LET LEV=5: GO SUB 66
14 LET SC=0
15 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS : DIM A(LEV): DIM B
(LEV): LET KILL=0

```

```

16 PRINT AT 0,3:"
   "
17 FOR N=1 TO 20: PRINT AT N,3
;" "; INK 4:"
"; INK 0;" "; NEXT n
18 PRINT AT 21,3:"
   "
19 BRIGHT 0: BORDER 7
20 PRINT AT 2,28;"7";AT 3,27;"
5"; INK 2;"X"; INK 0;"8";AT 4,28
;"6";AT 0,26;"Move.";AT 1,26;"--
--";AT 6,27;"Key";AT 7,27;"----"
21 PRINT AT 8,27;"You="; INK 6
;"a";AT 9,26; INK 0;"Zomb="; INK
2;"b";AT 10,26; INK 0;"Swamp";
INK 3;"c"
22 PRINT AT 12,27;"GOOD";AT 13
,27;"LUCK";AT 14,27;"   "
23 PRINT AT 16,28;"";LEV-KILL;
AT 17,28;"TO";AT 18,27;"KILL!"
24 FOR N=1 TO LEV
25 LET A(N)=INT (RND*20): LET
B(N)=INT (RND*20): IF ATTR (A(N)
+1,B(N)+4)<>124 THEN GO TO 25
26 PRINT AT A(N)+1,B(N)+4; INK
2; PAPER 4;"b": NEXT n
27 FOR N=1 TO LEV*5
28 LET A=INT (RND*20): LET B=I
NT (RND*20): IF ATTR (A+1,B+4)<>
124 THEN GO TO 28
29 PRINT AT A+1,B+4; INK 3; PA
PER 4;"c": NEXT n
30 LET X=INT (RND*20): LET Y=I
NT (RND*20): IF ATTR (X+1,Y+4)<>
124 THEN GO TO 30

```

```

31 PRINT AT X+1,Y+4; INK 6; PA
PER 4;"a": FOR N=0 TO 50 STEP 5:
BEEP .02,N: NEXT N
32 PRINT AT X+1,Y+4; INK 6; PA
PER 4;"a"
33 FOR N=1 TO LEV: IF A(N)=100
AND N=LEV THEN GO TO 49
34 IF A(N)=100 THEN NEXT N
35 IF KILL=LEV THEN GO TO 91
36 IF N>LEV THEN LET N=LEV
37 PRINT AT A(N)+1,B(N)+4; BRI
GHT 1; INK 4;" "
38 IF A(N)<X THEN LET A(N)=A(
N)+1
39 IF B(N)>Y THEN LET B(N)=B(
N)-1
40 IF A(N)>X THEN LET A(N)=A(
N)-1
41 IF B(N)<Y THEN LET B(N)=B(
N)+1
42 LET A=ATTR (A(N)+1,B(N)+4)
43 IF A=124 THEN PRINT AT A(N
)+1,B(N)+4; INK 2; PAPER 4; BRIG
HT 1;"b": BEEP .01,N: NEXT N
44 IF A=99 THEN LET SC=SC+10:
FOR I=7 TO 0 STEP -1: PRINT AT
A(N)+1,B(N)+4; BRIGHT 1; PAPER 4
; INK 1;"c": BEEP .02,I*7: NEXT
I: PRINT AT A(N)+1,B(N)+4; INK 3
; PAPER 4; BRIGHT 1;"c": LET KIL
L=KILL+1: PRINT AT 16,28;"";LEV-
KILL;" "; LET A(N)=100: IF N<=L
EV THEN NEXT N
45 IF N>=LEV THEN LET N=LEV

```

```

46 IF NK=LEV AND A(N)=X AND B(
N)=Y THEN FOR I=0 TO 7: PRINT A
T A(N)+1,B(N)+4; BRIGHT 1; PAPER
4; INK I; FLASH 1;"a": BEEP .1,
I*7: NEXT I: GO TO 87
47 IF KILL=LEV THEN GO TO 91
48 IF NK<LEV THEN NEXT N
49 IF LEV=KILL THEN PAUSE 30:
GO TO 91
50 FOR G=1 TO 2: PRINT AT X+1,
Y+4; INK 4; BRIGHT 1;" "
51 LET X=X+(INKEY$="6")-(INKEY
$="7"); LET Y=Y+(INKEY$="8")-(IN
KEY$="5")
52 LET A=ATTR (X+1,Y+4)
53 IF A=124 THEN PRINT AT X+1
,Y+4; INK 6; PAPER 4; BRIGHT 1;"
a": BEEP .04,20: NEXT G: GO TO 3
2
54 IF A=98 THEN GO TO 103
55 LET LEV=5: CLS : PRINT AT 1
,11;"ZOMBIES";AT 2,11;"======"
56 PRINT AT 3,1;"YOU ARE SLOWL
Y SINKING INTO THE"
57 FOR I=0 TO 7: PRINT AT I+6,
8; PAPER 7-I; INK I;" S W A M P
! ": NEXT I
58 PRINT AT 16,3;"HOWEVER YOU
HAVE MANAGED": PRINT AT 19,5;"TO
SCORE ";SC;" POINTS!"
59 PRINT £1;"(c) Copyright DAV
ID PERRY 1983"
60 LET I=0
61 PRINT AT 21,9; INK I;"PRESS
A KEY!"

```

```

62 LET I=I+1: IF I>7 THEN LET
I=1
63 IF INKEY$="" THEN GO TO 61
64 LET LEV=5: PAUSE 0
65 GO TO 106
66 CLS : PRINT AT 0,10;"ZOMBIE
S";AT 1,10;"======"
67 PRINT : PRINT " YOU HAVE J
UST LANDED ON AN"
68 PRINT " ISLAND POPULATED
WITH?"
69 RESTORE 70: FOR N=0 TO 31:
READ A: POKE USR "A"+N,A: NEXT N
70 DATA 0,24,126,24,24,24,24,0
,129,195,165,165,102,24,36,24,12
9,126,66,90,90,66,126,129,153,90
,60,231,231,60,90,153
71 FOR N=7 TO 13: PRINT AT N,1
0; INK N-7;"ZOMBIES": BEEP .5,N:
NEXT N
72 PRINT AT 14,10;"ZOMBIES": B
EEP .5,35
73 PRINT AT 17,0;"(c) COPYRIGH
T DAVID PERRY 1983!"
74 PAUSE 10: FOR N=0 TO 27: PR
INT AT 19,N; INK 2;" b "; INK 5;
"a": BEEP .02,N: NEXT N
75 PRINT AT 19,N;" ": PAUSE
10: BEEP .5,40: PAUSE 5: FOR N=
27 TO 0 STEP -1: PRINT AT 19,N;
INK 2;"b "; INK 5;"a ": BEEP .02
,N: NEXT N
76 CLS
77 PRINT AT 0,10;"ZOMBIES!";AT
1,10;"======"

```

```

78 PRINT AT 2,1;"The zombies a
re flesh-eaters         however they
are blind, that         is your only
advantage. They        come in packs
so you can use         yourself as b
ait to lure them      into the area
s of swamp.           Be careful,
if you fall in        a pit then yo
u will die too.       You only have
one life so-"

```

```

79 PRINT AT 12,4;" B E C A R
E F U L ! "

```

```

80 PRINT AT 14,1;" The movemen
t keys are 5-8."

```

```

81 PRINT AT 16,2;" YOU="; INK
5;"a"; INK 0;" ZOMBIE="; INK 2;"
b"; INK 0;" SWAMP="; INK 3;"c";
INK 0;". "

```

```

82 LET I=0: PRINT £1;"(c) Copy
right DAVID PERRY 1983"

```

```

83 PRINT AT 19,8; INK 1; PAPER
2; BRIGHT 1;" PRESS A KEY! "

```

```

84 LET I=I+1: BEEP .01,I*7: IF
I>7 THEN LET I=0

```

```

85 IF INKEY$="" THEN GO TO 83

```

```

86 CLS : RETURN

```

```

87 CLS : PRINT AT 1,11;"ZOMBIE
S";AT 2,11;"====="

```

```

88 FOR N=0 TO 7: PRINT AT 4+N,
0; PAPER N; INK 7-N;" T H E Y C
A U G H T Y O U ! ": NEXT N

```

```

89 GO TO 58

```

```

90 STOP

```

```

91 CLS : PRINT AT 1,11;"ZOMBIE
S";AT 2,11;"====="

```

```

92 FOR N=7 TO 0 STEP -1: PRINT
AT 4+N,0; PAPER N; INK 7-N;" C
O N G R A T U L A T I O N S ": N
EXT N

```

```

93 LET SC=SC+(LEV*20)

```

```

94 PRINT AT 13,1;"YOU HAVE SCO
RED ";SC;" SO FAR!"

```

```

95 LET LEV=LEV+5: FOR N=1 TO 1
0 STEP 2: BEEP .02,N: NEXT N: PR
INT AT 15,4;"NOW TRY THE NEXT LE
VEL": PRINT : PRINT "          WITH
";LEV;" ZOMBIES!"

```

```

96 PRINT AT 19,7;"BONUS POINTS
=";(LEV-5)*20;"

```

```

97 PRINT £1;"(C) Copyright DAV
ID PERRY 1983"

```

```

98 LET I=0

```

```

99 PRINT AT 21,8; INK 1;"PRESS
ANY KEY!"

```

```

100 LET I=I+1: BEEP .02,I*5: IF
I>7 THEN LET I=0

```

```

101 IF INKEY$="" THEN GO TO 99
102 GO TO 15

```

```

103 CLS : PRINT AT 1,11;"ZOMBIE
S";AT 2,11;"====="

```

```

104 FOR N=7 TO 0 STEP -1: PRINT
AT 6+N,0; PAPER N; INK 7-N;" YO
U HAVE JUST COMMITED SUICIDE ":
NEXT N

```

```

105 GO TO 58

```

```

106 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS

```

```

107 IF SC<=N(10) THEN GO TO 12
3

```

```

108 LET NUM=11: IF SC>=N(NUM) T
HEN INPUT "ENTER 8 INITIALS! ";P
#: IF LEN P#>8 THEN GO TO 108
109 PRINT AT 7,0: FLASH 1: BRIG
HT 0: INK 7: PAPER 2:"THIS WILL
ONLY TAKE A FEW SECS!"
110 IF SC>=N(NUM) THEN LET N(N
UM)=SC: LET N#(NUM)=P#
111 FOR A=1 TO (NUM-1): LET B#=
N#(A): LET C#=N#(A+1): LET B=N(A
): LET C=N(A+1): IF B<C THEN LE
T N(A)=C: LET N(A+1)=B: LET N#(A
)=C#: LET N#(A+1)=B#
112 NEXT A: FOR N=1 TO NUM-1: I
F N(N)<N(N+1) THEN GO TO 111
113 NEXT N
114 CLS
115 PRINT AT 2,4:"H A L L O F
F A M E !"
116 PRINT AT 3,4:"=====
=====
117 FOR N=1 TO NUM-1: PRINT AT
N+5,7: INK 6:"("; INK 2;N: INK
6;")";AT N+5,12: INK 7;N(N): PRI
NT AT N+5,17: INK 5;N#(N): NEXT
N
118 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
PRINT AT 17,8: PAPER 2: INK 7:"
PRESS ANY KEY! ": LET I=0
119 LET I=I+1: IF I>7 THEN LET
I=0
120 PRINT AT 2,4: INK I:"H A L
L O F F A M E !"
121 BEEP .01,1*7: PAUSE 2: IF I
NKEY#="" THEN GO TO 119

```

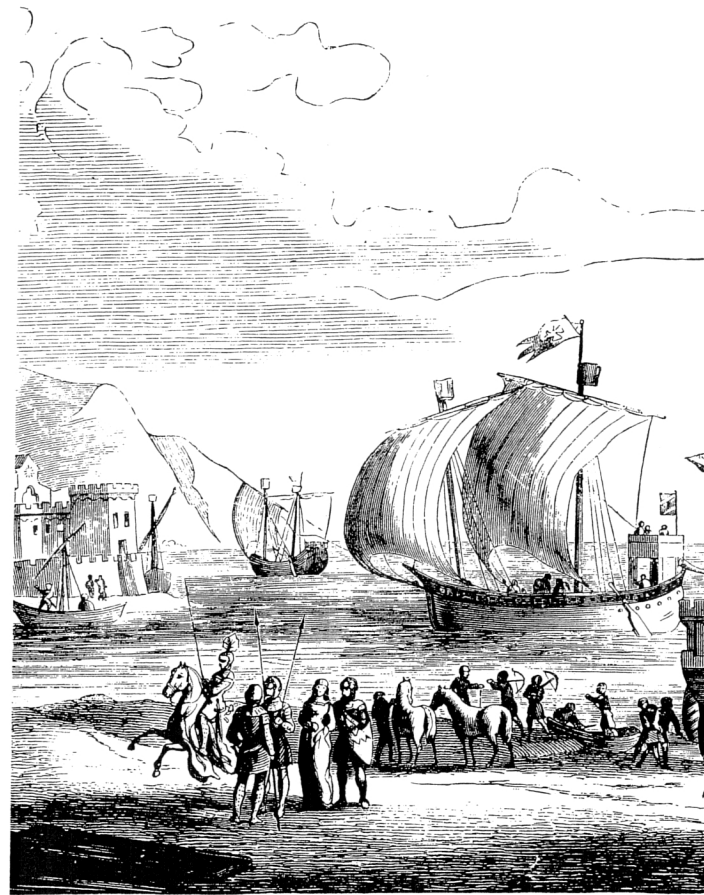
```

122 PAPER 7: INK 0: BORDER 7: B
RIGHT 0: CLS : GO TO 14
123 PRINT AT 10,4:"SORRY SCORE
TO LOW!!!"
124 PRINT AT 13,7:" PRESS A KEY
"
125 PAUSE 0: GO TO 114

```



# ADVENTURES



## TROLL and THE CITY

These two Adventure games, written by Neil Pellinacci, are played in similar ways, although the scenario and solution are - of course - different. It is worth trying to convince a friend to type the programs in for you, so you don't find out the solutions even before you run them.

THE CITY includes instructions in the program which precedes the main game. To complete the Adventure, you must say the password so your contact can hear it. You don't know who the contact is, although there are not many possibilities.

In the second Adventure, TROLL, you must try and escape from the Troll's lair. He's very mean and will let you escape only over his dead body. Therefore, you have to kill him to win.

In both programs, as in most Adventure games, you enter your commands as two-word phrases (verbs and nouns) such as GO WEST or DROP AXE. You'll soon discover which words the program will recognize.

Note that each program has a loader program (which for THE CITY doubles as an instruction manual). This sets up the screen and keyboard. 48K owners can leave out the CLEAR 32767 instruction, but the

CLS must be left in. If you use the in-built SAVE instructions, note that you'll have to set the colors and POKEs beforehand.

The loaders have been used to maximise memory space for the main program. To type the games in, enter and SAVE the loader program, then type in and SAVE the main game directly after it. When saving the programs, or re-running them, remember that the start line is 1 for the loaders and 900 for the main programs.

Here is the loader for TROLL:

```

10 REM Adventure © Neil
    Pellinacci 1983
20 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS
30 CLEAR 02757
40 POKE 23609,100: POKE 23650,
6: POKE 23624,71
50 LOAD ""

```

And this is the main TROLL program:

```

12 PRINT "I AM AT THE TROLL'S
GATE."
13 IF 0(9) THEN PRINT "I HAVE
NO WEAPON AND I MUST LEAVE 0
QUICKLY."
14 LET S=P: RETURN
15 PRINT "I AM IN A SMALL SHD
P, WHICH SELLS ONLY DOG-FOOD
.": LET S=P: RETURN
33 PRINT "I AM IN A SMALL, DA
RK CAVE. THERE IS A GATE TO
THE NORTH, AND A CLEARING TO T
HE EAST."
34 LET N=P: LET S=P: LET E=P:
RETURN

```

```

36 PRINT "I AM IN A CLEARING.
THE CAVE IS TO THE WEST.": LET
E=P: LET W=P: RETURN
39 PRINT "I AM ON A PATH, WHI
CH CONTINUES TO THE SOUTH. I CAN
SEE A SHOP TO THE NORTH, AND A
FOUNTAIN TO THE EAST."
40 LET N=P: LET S=P: LET E=P:
LET W=P: RETURN
42 PRINT "THERE IS A FOUNTAIN
HERE. A ROADLEADS OFF TO THE SO
UTH.": LET S=P: LET W=P: RETURN
54 PRINT "I AM STANDING IN A
NARROW, ROCKYPASSAGE WHICH GENTL
Y RISES TO THE NORTH. I CAN SE
E DAYLIGHT ABOVE.": LET N=P: L
ET U=P: RETURN
60 PRINT "I AM STANDING ON A
GRAVEL PATH. A ROAD LIES TO THE
EAST.": LET N=P: LET E=P
61 IF 0(4)=LOC THEN PRINT "THE
RE IS A MOUND AT MY FEET."
62 RETURN
63 PRINT "A ROAD RUNS NORTH-S
OUTH. A FOUNTAIN LIES TO TH
E NORTH.": LET N=P: LET S=P: RET
URN
69 PRINT "I CAN SEE A SMALL C
LEARING WITH A LARGE TREE.": LET
S=P: LET E=P: RETURN
72 PRINT "I AM IN A THICK FOR
EST. THERE ARE CLEARINGS TO TH
E EAST AND SOUTH."
73 LET W=P: LET E=P: LET S=P:
RETURN
75 PRINT "I AM STANDING IN A
CLEARING. THERE IS A DEEP SHA
FT IN THE GROUND.": LET W=P:
LET D=P: RETURN
84 PRINT "THE ROAD ENDS HERE.
THE RIVER IS TO THE SOUTH, WITH
MOUNTAINS BEYOND.": LET N=P:
RETURN
87 PRINT "A LARGE WOLF LEAPT
OUT OF THE TREES, AND KILLED M
E.": GO TO 9000
90 PRINT "I AM STANDING IN A
CLEARING. I CAN HEAR RUSTLING N
OISES TO THE WEST.": LET N=P: LE
T E=P: LET W=P: RETURN

```

```

93 PRINT "THE FOREST ENDS HERE
E. A RIVER LIES TO THE SOUTH."
IT MAY BE POSSIBLE TO CROSS."
LET N=P: LET U=P: RETURN
114 PRINT "I AM STANDING ON THE
RIVER BANK, I CAN SEE AN OPEN G
ATE GUARDED BY A DOG, AND A MAN
SION BEYOND."
115 LET S=DOG: IF NOT S THEN PR
INT "THE DOG WILL BE A PROBLEM.."
116 RETURN
117 PRINT "I AM NEAR A MANSION
A DOOR IS AHEAD OF ME.": LET
N=P: RETURN
118 PRINT "I AM NOW INSIDE THE
MANSION. I CERTAINLY WOULDN'T L
IVE HERE. IT'S ALL DUSTY AND S
MELLY.": LET U=P: RETURN
120 PRINT "I CAN SEE...": LET
FL=0: FOR Z=P TO 9: IF D(Z)=LOC
THEN GO TO 230: REM THEN PRINT "
A ";D$(Z): LET FL=1
121 NEXT Z: IF NOT FL THEN PRIN
T "NOTHING"
122 RETURN
123 IF Z=4 AND LOC=20 THEN GO T
O 310
124 PRINT "A ";D$(Z): LET FL=1:
GO TO 310
125 IF A$(P)=" " THEN LET A#=A#
(O TO ): GO TO 350
126 IF A$(LEN A#)=" " THEN LET
A#=A$( TO LEN A#-P): GO TO 360
127 LET B#="": FOR Z=P TO LEN A
#: IF A$(Z)=" " THEN LET B#=A$(Z
+P TO ): LET A#=A$( TO Z-P): RET
URN
128 NEXT Z: RETURN
129 GO SUB 9500: CLS
130 LET N=0: LET S=0: LET E=0:
LET U=0: LET D=0: LET W=0
131 GO SUB LOC*3: PRINT "EXITS
: ("NORTH " AND N); ("SOUTH " AN
D S); ("EAST " AND E); ("WEST " AN
D W): GO SUB 800
132 INPUT INK 5; "WHAT SHALL I D
O?": LINE A#: IF A#="" THEN GO T
O C
133 GO SUB 850: PRINT INK 6;A#:
";B#: PRINT

```

```

1007 IF LOC=4 AND AND>.88 THEN P
RINT "THE TROLL LEAPT OUT AND SM
ASHED MY SKULL. I AM DEAD.": GO
TO 9000
1008 IF LOC=39 AND NOT DOG AND R
ND>.9 THEN PRINT "THE DOG ATTACK
ED ME AND KILLED ME.": GO TO 90
00
1010 IF A#="N" AND N THEN LET LO
C=LOC-F: GO TO C-P-P
1020 IF A#="S" AND S THEN LET LO
C=LOC+F: GO TO C-P-P
1030 IF A#="E" AND E THEN LET LO
C=LOC+P: GO TO C-P-P
1040 IF A#="W" AND W THEN LET LO
C=LOC-P: GO TO C-P-P
1045 IF A#="U" OR A#="E" OR A#="
N" OR A#="S" THEN PRINT "I CAN'T
GO IN THAT DIRECTION.": GO TO C
1050 IF A#="I" THEN GO SUB 2000:
GO TO C
1060 IF A#="LOOK" THEN CLS : GO
TO C-P-P
1070 IF A#="D" THEN GO TO 3000
1080 IF A#="U" THEN GO TO 3100
1090 IF A#="GET" OR A#="TAKE" TH
EN GO TO 2100
1100 IF A#="DROP" OR A#="PUT" TH
EN GO TO 2200
1110 IF A#="KILL" THEN GO TO 230
0
1120 IF A#="OPEN" THEN GO TO 240
0
1130 IF A#="CROSS" THEN GO TO 25
00
1140 IF A#="UNLOCK" THEN GO TO 2
600
1150 IF A#="FEED" THEN GO TO 270
0
1160 IF A#="BUY" THEN GO TO 2800
1170 IF A#="SAVE" THEN SAVE "AD"
LINE 998: PRINT "VERIFY...": VE
RIFY "AD": GO TO C
1180 IF A#="QUIT" THEN GO TO 897
0
1190 IF A#="HELP" THEN PRINT "TH
E TROLL WON'T LET YOU GO EVEN IF
YOU ASK NICELY...": GO TO C
1200 IF A#="DIG" THEN GO TO 2900
1200 PRINT "I DON'T UNDERSTAND."
: GO TO C

```

```

2000 PRINT "I HAVE THE FOLLOWIN
G...": LET FL=B: FOR Z=1 TO 9: I
F O(Z)=0 THEN PRINT "A ";O$(Z):
LET FL=P
2010 NEXT Z: IF NOT FL THEN PRIN
T "NOTHING."
2020 RETURN
2100 IF B$="" THEN INPUT INK 5; (
B$); "WHAT ??": LINE B$: GO TO 2
100
2105 IF OB=4 THEN PRINT "I CAN'T
CARRY ANY MORE.": GO TO C
2110 FOR Z=P TO 9: IF O$(Z) ( TO
LEN B$)=B$ AND O(Z)=LOC THEN GO
TO 2130
2120 NEXT Z: PRINT "I CAN'T DO T
HAT.": GO TO C
2130 IF B$="DOG BONE" AND LOC=6
THEN PRINT "THAT WOULD BE STEALI
NG!": GO TO C
2140 PRINT "OK..": LET O(Z)=O: L
ET OB=OB+P: GO TO C
2200 IF B$="" THEN INPUT INK 5; "
WHAT SHALL I ";(A$); " ??": LINE
B$: GO TO 2200
2210 FOR Z=P TO 9: IF O$(Z) ( TO
LEN B$)=B$ AND O(Z)=0 THEN PRINT
"OK..": LET O(Z)=LOC: LET OB=OB
-P: GO TO C
2220 NEXT Z: PRINT "I HAVE NO ";
B$; "!": GO TO C
2300 IF B$="" THEN INPUT INK 5; "
KILL WHAT?" LINE B$: GO TO 2300
2310 IF B$(">"TROLL" THEN PRINT "
I'M NOT THAT SORT OF PERSON!": G
O TO C
2315 IF LOC(">"4 THEN PRINT "I CAN
'T SEE A TROLL!": GO TO C
2320 IF O(9) THEN PRINT "YOU HAV
E NOTHING TO KILL THE TROLL W
ITH.": GO TO C
2330 PRINT "WELL DONE, YOU HAV
E KILLED THE TROLL AND EARNT YO
UR FREEDOM.": GO TO 2000
2400 IF B$="" THEN INPUT INK 5; "
WHAT SHALL I OPEN?" LINE B$: GO
TO 2400
2410 IF B$(">"DOOR" THEN PRINT "T
HAT I CANNOT DO.": GO TO C
2420 IF LOC(">"39 THEN PRINT "WHAT
DOOR?": GO TO C

```

```

2430 IF NOT DO THEN PRINT "IT'S
LOCKED.": GO TO C
2440 PRINT "OK..": LET LOC=LOC+P
: GO TO C-P-P
2500 IF B$="" THEN INPUT INK 5; "
WHAT DO YOU WANT TO CROSS?" LINE
B$: GO TO 2500
2505 IF B$(">"RIVER" THEN PRINT "
I'M NOT THAT CLEVER, MATE!": GO
TO C
2507 IF LOC(">"31 AND LOC(">"32 THEN
PRINT "SHOW ME A RIVER, AND I'L
L TRY.": GO TO C
2510 IF O(2) THEN PRINT "I'M GOI
NG TO NEED SOMETHING TO HELP ME
.": GO TO C
2520 PRINT "OK..": LET LOC=LOC+(
LOC=31)-(LOC=32): GO TO C-P-P
2600 IF B$="" THEN INPUT INK 5; "
PLEASE CONTINUE YOUR COMMAND."
LINE B$: GO TO 2600
2605 IF B$(">"DOOR" THEN PRINT "T
HAT COULD TAKE ME A WHILE.": GO
TO C
2610 IF LOC(">"39 THEN PRINT "I CA
N'T SEE A DOOR!": GO TO C
2620 IF DO THEN PRINT "YOU FOOL,
IT'S ALREADY UNLOCKED!": GO TO
C
2625 IF O(4) THEN PRINT "YOU HAV
E NO KEY...": GO TO C
2630 PRINT "OK..": LET DO=P: GO
TO C
2700 IF B$(">"DOG" THEN PRINT "I
DON'T THINK IT'S WORTH IT.": GO
TO C
2710 IF LOC(">"32 THEN PRINT "WHAT
DOG?": GO TO C
2720 IF O(P) THEN PRINT "WHAT WI
TH, YOUR FINGERS?": GO TO C
2730 PRINT "THE DOG HAS FALLEN
ASLEEP.": LET DOG=P: LET S=P: LE
T O(P)=LOC: LET OB=OB-P: GO TO C
2800 IF O(F) THEN PRINT "YOU HAV
E NO MONEY.": GO TO C
2810 IF B$="" THEN INPUT INK 5; "
ANYTHING IN PARTICULAR?" LINE B
$: GO TO 2810
2820 IF B$="NO" THEN PRINT "WATC
H IT, MATE! I'LL LOSE MY PATI
ENCE!": GO TO C

```

```

2830 IF B$(1)"DOG BONE" THEN PRINT
T "WHY BOTHER IF IT'S FREE?": GO
TO C
2840 IF LOC<>5 THEN PRINT "THERE
'S NO DOG BONE HERE.": GO TO C
2845 IF OB=4 THEN PRINT "I CAN'T
CARRY ANY MORE.": GO TO C
2850 PRINT "OK.": LET O(P)=O: L
ET OB=OB+P: GO TO C
2890 IF LOC<>20 THEN PRINT "I CA
N'T DO THAT YET.": GO TO C
2910 IF O(4)<>LOC THEN PRINT "TH
ERE'S NOTHING THERE.": GO TO C
2920 PRINT "I HAVE FOUND A KEY."
: GO TO C
2960 IF LOC<>25 THEN PRINT "DOWN
WHERE?": GO TO C
2910 IF O(5) THEN PRINT "I'M NOT
JUMPING, I'LL BREAK MY NECK!":
GO TO C
2920 PRINT "OK.": LET LOC=LOC-F
: GO TO C-P-P
3100 IF LOC<>18 THEN PRINT "I CA
N'T DO THAT YET.": GO TO C
3110 IF O(5) THEN PRINT "HOW DID
YOU GET ME DOWN IN THE FIRST P
LACE?": GO TO C
3120 PRINT "OK.": LET LOC=LOC+F
: GO TO C-P-P
2970 CLS : INPUT "ARE YOU SURE -
- Y/N:": LINE A#
2980 IF A#="Y" OR A#="YES" THEN
GO TO 9000
2990 GO TO C
3000 INPUT "DO YOU WANT ANOTHER
GO? (Y/N)": LINE A#
3010 IF A#="Y" OR A#="YES" THEN
RUN 900
3020 IF A#="N" OR A#="NO" THEN S
TOP
3030 GO TO 9000
3500 LET O=PI-PI: LET P=PI/PI
3510 LET F=VAL "7": LET LOC=VAL
"24": LET OB=0
3520 LET DO=0: LET DOG=0: LET C=
VAL "1000": DIM O$(9,8): DIM O(9
)
3530 FOR Z=P TO INT (PI*PI): REA
D O$(Z),Y$: LET O(Z)=VAL Y#
3540 NEXT Z

```

```

9550 DATA "DOG BONE", "6", "PLANK"
, "11", "RING", "14", "KEY", "20", "RO
PE", "23", "ROCK", "28", "WALLET", "3
0", "BOOK", "39", "GUN", "40"
9560 RETURN

```

This next program is the loader for THE CITY:

```

10 REM The City
20 REM An Adventure Game
30 REM By Neil Pellinacci 1983
40 REM 16k Spectrum
50 REM
60 REM
70 BORDER 0: PAPER 0: INK 7: B
RIGHT 1
80 CLEAR 32767: CLS
90 POKE 23609,100: POKE 23650,
8: POKE 23624,71
95 GO SUB 200
100 LOAD ""
200 DIM A$(32): PAPER 6: INK 0:
LET A#=""
: LET L=9: GO SUB 900
210 PAUSE 100: PAPER 0: CLS
220 LET A#="" THE CIT
Y": LET L=3: INK 6: GO SUB 900
230 LET A#="AN ADVENTURE FOR TH
E ZX SPECTRUM": LET L=5: GO SUB
900
240 LET A#="WRITTEN BY NEIL PEL
LINACCI 1983": INK 5: LET L=8:
GO SUB 900
250 LET A#="" PRESS ANY KEY TO
CONTINUE": INK 7: LET L=16: GO
SUB 900
260 LET L=1: PAUSE 1: PAUSE 0:
CLS
265 FOR Z=-20 TO 30 STEP 5: BEE
P .01,Z: BEEP .01,Z/10: BEEP .01
,Z-10: BEEP .01,Z+10: NEXT Z
270 RESTORE
280 READ A#: IF A$(1)="*" THEN
RETURN
290 IF A$(1)="+" THEN PRINT AT
21,0: INK 6: " PRESS ANY KEY TO
CONTINUE": PAUSE 1: PAUSE 0: CL
S : BEEP .1,0: BEEP .1,10: BEEP

```

```

.1,20: BEEP .3,30: LET L=1: GO T
0 280
300 GO SUB 900: LET L=L+2: GO T
0 280
400 DATA " IN THIS ADVENTURE GA
ME YOU MUST "
410 DATA "RECOVER SOME STOLEN M
ONEY. THIS"
420 DATA "MONEY IS, HOWEVER, ST
ORED IN THE"
430 DATA "BANK VAULT. THE BANK
IS OWNED BY"
440 DATA "THE GANGSTER WHO STOL
E THIS "
450 DATA "MONEY."
460 DATA " YOU MUST BREAK INTO
THE BANK,"
470 DATA "AND STEAL THE MONEY B
ACK. "
480 DATA "HOWEVER, THIS WILL NO
T BE EASY,"
490 DATA "AS YOU WILL APPEAR TO
BE A THIEF"
500 DATA "+"
510 DATA " TO COMBAT THIS, YOU
HAVE "
520 DATA "ARRANGED A BOMB SCARE
, AND SO"
530 DATA "THE CITY IS NEARLY EM
PTY. YOU "
540 DATA "MUST STILL BE CAREFUL
THAT THE"
550 DATA "POLICE DON'T GET THE
WRONG IDEA."
560 DATA "THEY WILL NOT KNOW AB
OUT YOUR"
570 DATA "SECRET MISSION."
580 DATA " WHEN YOU HAVE COLLEC
TED THE"
590 DATA "MONEY, YOU MUST FIND
YOUR"
600 DATA "CONTACT. HE WILL TAKE
YOU AWAY."
610 DATA "+"
620 DATA " TO PLAY, ENTER YOUR
WISHES IN"
630 DATA "PLAIN ENGLISH, USING
VERBS AND"
640 DATA "NOUNS. THE EXCEPTION
IS 'N' TO"
650 DATA "GO NORTH, 'S' TO GO S
OUTH, ETC."

```

```

660 DATA " IF YOU WANT TO STOP,
TYPE "
670 DATA " 'STOP' OR 'QUIT'. 'SA
VE' WILL "
680 DATA "SAVE THE CURRENT GAME
ON TAPE."
690 DATA "SO THAT YOU CAN FINIS
H IT LATER."
700 DATA "+"
710 DATA "",""/","/","/","/","/ " S
TART THE TAPE"
720 DATA "+"
800 PAPER 6: INK 7: STOP
900 FOR Z=32 TO 1 STEP -1: PRIN
T AT L,0;A$(Z TO ): NEXT Z: RETU
RN

```

Once you have that in place, you can enter and save THE CITY program itself:

```

9 PRINT "I AM IN A TUBE STAT
ION. THERE IS A TRAIN WAITING TO
LEAVE. THE TAXI WAITS OUTSIDE."
: RETURN
12 PRINT "I AM STANDING BEHIN
D A BUTCHER'S SHOP. I CAN SEE A B
UILDING SITE TO THE EAST." : LET
E=P: LET S=P: RETURN
15 PRINT "A LARGE STEEL GIRDE
R FELL ON ME. I AM DEAD." : GO TO
9000
21 PRINT "I AM OUTSIDE THE BA
NK. THE DOOR IS BEFORE ME. THE H
IGH STREET IS TO THE EAST." : LET
E=P: LET DO1=0: RETURN
24 PRINT "I AM IN THE HIGH ST
REET, WHICH IS DESERTED, APART
FROM A TAXI. A MARKET IS TO THE
SOUTH." : LET E=P: LET U=P: LET S
=P: RETURN
27 PRINT "I AM STANDING IN A
LARGE SHOP. I CAN SEE NO PEOPLE
. THERE IS A CAFE TO THE EAST." :
LET N=P: LET S=P: LET U=P: LET
E=P: RETURN
30 PRINT "I AM STANDING IN A
SMALL, DESERTED CAFE. THE
LARGE SHOP IS TO THE WEST." : LET
U=P: RETURN

```



```

36 PRINT "I AM INSIDE THE BANK. IT IS VERY QUIET. "; IF GU THEN PRINT "I CAN SEE A GUARD WITH HIS BACK TO ME.";
37 PRINT : LET N=P: LET S=NOT GU: RETURN
39 PRINT "I AM IN A STREET MARKET. A POLICE STATION IS TO THE EAST, AND A BUILDING SITE TO THE SOUTH."
40 LET N=P: LET S=P: LET E=P: RETURN
42 PRINT "I HAVE ARRIVED AT THE POLICE STATION. IT IS NOT DESERTED...."
43 IF NOT O(7) THEN PRINT "A POLICEMAN SEES THE MONEY AND ARRESTS ME.": GO TO 9000
44 PRINT "A POLICEMAN TAKES WHAT I HAVE COLLECTED AND RETURNS THEM TO WHERE I FOUND THEM.": LET N=P: LET W=P: GO TO 700
45 PRINT "I AM IN A SMALL GLOOMY TUBE STATION. THERE IS A HOLE IN THE WALL TO THE SOUTH.": LET S=P: RETURN
46 PRINT "I AM STANDING IN FRONT OF THE VAULT DOOR, WHICH IS LOCKED."
49 IF UC THEN PRINT "THERE IS A HOLE IN IT."
50 LET E=P: LET S=UC: RETURN
51 PRINT "I CAN SEE TWO DOORS, ONE TO THE WEST, ONE TO THE SOUTH. THE VAULT ENTRANCE IS TO THE NORTH.": LET N=P: LET S=P: LET W=P: RETURN
54 PRINT "I AM ON A BUILDING SITE, WHICH IS SURROUNDED BY HIGH WALLS.": LET N=P: RETURN
57 PRINT "I'M NOW STANDING IN THE FOYER OF A LARGE HOTEL. BEFORE ME IS THE RECEPTION DESK.": LET E=P: RETURN
60 PRINT "I AM IN THE BASEMENT OF A LARGE BUILDING.": LET N=P: LET W=P: RETURN
63 PRINT "I AM INSIDE THE VAULT. TO REACH THE MONEY, I HAVE TO ENTER THE KEYPANEL CODE. THERE ARE 9 BUTTONS IN FRONT OF

```

```

ME. I MUST PRESS THE RIGHT ONE."
64 LET N=P: RETURN
65 PRINT "A WALL HAS APPEARED BEHIND ME. I AM TRAPPED.": GO TO 9000
78 PRINT "I AM INSIDE THE MAIN VAULT ROOM.": LET N=P: RETURN
600 IF A$(P)=" " THEN LET A#=A$(2 TO ): GO TO 600
610 IF A$(LEN A#)=" " THEN LET A#=A$(1 TO LEN A#-P): GO TO 610
620 LET B#="": FOR Z=P TO LEN A#: IF A$(Z)=" " THEN LET B#=A$(Z+P TO ): LET A#=A$(1 TO Z-P): RETURN
630 NEXT Z: RETURN
700 RESTORE 6000: FOR Z=P TO 7: READ A#,B#: IF O(Z)=0 THEN LET O(Z)=VAL B#
710 NEXT Z: LET OB=0: RETURN
600 PRINT "EXITS: "; ("NORTH " AND N); ("SOUTH " AND S); ("EAST " AND E); ("WEST " AND W)
810 PRINT "I CAN SEE..."
820 LET FL=0: FOR Z=P TO 7: IF O(Z)=LOC THEN PRINT "A "; O$(Z): LET FL=P
830 NEXT Z: IF NOT FL THEN PRINT "NOTHING INTERESTING."
840 RETURN
900 GO SUB 2000: CLS
999 LET N=0: LET S=0: LET E=0: LET W=0: GO SUB LOC*3: GO SUB 600
1000 INPUT INK 5; "WHAT SHALL I DO NOW?": LINE A#: IF A#="" THEN GO TO C
1010 IF LOC=12 AND RND>.9 AND GU THEN PRINT "THE GUARD TURNED, SAW ME, THEN SHOT ME. I AM DEAD.": GO TO 9000
1050 GO SUB 600: PRINT " INK 6; A$; " : B#
1060 IF A#="N" AND N THEN LET LOC=LOC-F: GO TO C-P
1070 IF A#="S" AND S THEN LET LOC=LOC+F: GO TO C-P
1080 IF A#="E" AND E THEN LET LOC=LOC+P: GO TO C-P
1090 IF A#="W" AND W THEN LET LOC=LOC-P: GO TO C-P

```



```

1100 IF A$="N" OR A$="S" OR A$="
E" OR A$="W" THEN PRINT "I CAN'T
GO IN THAT DIRECTION.": GO TO C
1110 IF A$="GET" OR A$="TAKE" TH
EN GO TO 2000
1120 IF A$="DROP" OR A$="PUT" TH
EN GO TO 2100
1130 IF A$="I" THEN GO TO 2200
1140 IF A$="LOOK" THEN CLS : GO
TO C-P
1150 IF A$="HELP" THEN GO TO 230
0
1160 IF A$="BOARD" OR A$="CATCH"
THEN GO TO 2400
1170 IF A$="UNLOCK" THEN GO TO 2
500
1180 IF A$="OPEN" THEN GO TO 260
0
1190 IF A$="D" THEN GO TO 2700
1200 IF A$="QUIT" OR A$="STOP" T
HEN GO TO 8900
1210 IF A$="READ" OR A$="EXAMINE
" THEN GO TO 2800
1220 IF A$="KILL" THEN GO TO 290
0
1230 IF A$="OUT" THEN GO TO 3000
1240 IF A$="PRESS" OR A$="PUSH"
THEN GO TO 3100
1250 IF A$="SAVE" THEN SAVE "CIT
Y" LINE 999: PRINT "VERIFY...":
VERIFY "": GO TO C
1260 IF A$="SAY" THEN GO TO 3200
1300 PRINT "I DON'T UNDERSTAND."
: GO TO C
2000 IF B$="" THEN INPUT INK 5;"
GET WHAT?": LINE B$: GO TO 200
0
2002 IF LEN B$>13 THEN PRINT "I
CAN'T": GO TO C
2005 IF OB=3 THEN PRINT "I CAN'T
CARRY ANY MORE.": GO TO C
2010 FOR Z=P TO 7: IF O$(Z) ( TO
LEN B$)=B$ AND O(Z)=LOC THEN LET
O(Z)=0: LET OB=OB+1: PRINT "OK.
": GO TO C
2020 NEXT Z: PRINT "I SEE NO ";B
$;"!": GO TO C
2100 IF B$="" THEN PRINT A$;" WH
AT?": GO TO C
2110 IF LEN B$>13 THEN PRINT "I
CAN'T.": GO TO C

```

```

2120 FOR Z=P TO 7: IF O$(Z) ( TO
LEN B$)=B$ THEN PRINT "OK.": LE
T O(Z)=LOC: LET OB=OB-1: GO TO C
2130 NEXT Z: PRINT "I HAVE NO ";
B$;"!": GO TO C
2200 PRINT "I HAVE THE FOLLOWING
.": LET FL=0: FOR Z=P TO 7: IF
O(Z)=0 THEN PRINT "A ";O$(Z): LE
T FL=P
2210 NEXT Z: IF NOT FL THEN PRIN
T "NOTHING."
2220 GO TO C
2300 IF LOC=17 THEN PRINT "ASK T
HE VILLAGE PEOPLE.": GO TO C
2310 PRINT "FOLLOW THE OLD LONDO
N - BRISTOL ROAD!": GO TO C
2400 IF B$="" THEN PRINT A$;" WH
AT?": GO TO C
2410 IF B$(<>"TRAIN" AND B$(<>"TAX
I" THEN PRINT "I CAN'T.": GO TO
C
2420 IF B$="TRAIN" AND LOC(<>3 AN
D LOC(<>15 THEN GO TO 2430
2430 IF B$="TAXI" AND LOC(<>3 AND
LOC(<>8 THEN GO TO 2430
2440 IF B$="TRAIN" THEN GO TO 24
70
2450 IF O(2) THEN PRINT "I HAVE
NO MONEY FOR THE FARE.": GO TO C
2460 PRINT "THE TAXI DRIVES OFF
WITH ME INSIDE.": LET LOC=LO
C+(F AND LOC=3)-(F AND LOC=8): G
O TO C-P
2470 IF O(3) THEN PRINT "I NEED
A TICKET!!": GO TO C
2480 PRINT "I GET IN AND THE TRA
IN MOVES AWAY.": LET LOC=(3 A
ND LOC=15)+(15 AND LOC=3): GO TO
C-P
2490 PRINT "I CAN'T DO THAT YET.
": GO TO C
2500 IF B$="" THEN PRINT "WHAT S
HALL I UNLOCK?": GO TO C
2510 IF B$(<>"DOOR" THEN PRINT "I
CAN'T DO THAT.": GO TO C
2520 IF LOC(<>7 THEN PRINT "WHAT
DOOR?": GO TO C
2530 IF DO THEN PRINT "YOU IDIOT
, IT'S ALREADY UNLOCKED!
!": GO TO C
2540 IF O(5) THEN PRINT "I NEED
THE KEY.": GO TO C

```

```

2550 PRINT "OK..": LET DO=P: GO
TO C
2560 IF B$="" THEN PRINT "OPEN W
HAT?": GO TO C
2570 IF B$("<>DOOR") THEN PRINT "I
CAN'T.": GO TO C
2580 IF LOC("<>7") THEN PRINT "I CAN
'T SEE A DOOR!": GO TO C
2590 IF NOT DO THEN PRINT "IT'S
LOCKED.": GO TO C
2600 LET DO1=P: PRINT "OK..""I
CAN SEE SOME STAIRS GOING
DOWN.": GO TO C
2700 IF LOC("<>7") THEN GO TO 2490
2710 IF NOT DO1 THEN PRINT "THER
E'S A DOOR IN THE WAY!": GO TO C
2720 PRINT "OK..": LET LOC=LOC+F
: GO TO C-P
2800 IF B$="" THEN INPUT INK 5; (
A$); "WHAT?": LINE B$: GO TO 280
0
2810 IF B$("<>CLUE") THEN PRINT "I
CAN'T DO THAT!": GO TO C
2820 IF O(6) THEN PRINT "I'M NOT
CARRYING A CLUE.": GO TO C
2830 PRINT "THE CLUE IS WRITTEN
ON A4-SIZED PAPER. IT SAYS: 'RUB
A DUB DUB...': GO TO C
2890 IF B$="" THEN PRINT "KILL W
HAT?": GO TO C
2910 IF B$("<>GUARD") THEN PRINT "
I REFUSE ON MORAL GROUNDS.": GO
TO C
2920 IF LOC("<>12") THEN PRINT "I CA
N'T SEE A GUARD!": GO TO C
2930 IF O(P) THEN PRINT "I HAVE
NO WEAPON.": GO TO C
2940 PRINT "OK..": LET GU=0: GO
TO C
3000 IF B$="" THEN INPUT INK 5; "
WHAT SHALL I CUT?": LINE B$: GO
TO 3000
3010 IF B$("<>VAULT DOOR") THEN PR
INT "I CAN'T DO THAT.": GO TO C
3020 IF LOC("<>16") THEN PRINT "I SE
E NO VAULT DOORS HERE!": GO TO C
3030 IF O(4) THEN PRINT "I HAVE
NOTHING TO CUT IT WITH.": GO TO
C
3040 PRINT "OK..": LET UC=P: GO
TO C-P

```

```

3100 IF LOC("<>21") THEN PRINT "THER
E'S NOTHING HERE TO ";A$;"!": GO
TO C
3110 IF LEN B$("<>P") THEN PRINT "I
NEED A NUMBER BETWEEN 1 AND 9.":
GO TO C
3120 IF B$("<1") OR B$(">9") THEN PR
INT "THAT'S NOT A NUMBER!!": GO
TO C
3130 LET Z=VAL B$: IF Z("<>3") THEN
LET LOC=LOC+P: GO TO C-P
3140 LET LOC=LOC+F: PRINT "THE I
RON DOOR SLIDES BACK SLOWL
Y...": GO TO C-P
3200 IF B$="" THEN PRINT "SAY WH
AT?": GO TO C
3210 IF B$("<>PASSWORD") THEN PRIN
T "I CAN'T SAY THAT!": GO TO C
3220 IF LOC=15 AND NOT O(7) THEN
PRINT ""I HAVE FOUND THE CONTRA
CT, AND HAVE SUCCESSFULLY COMP
LETED MY MISSION. THE CONTACT W
AS THE TRAIN DRIVER.": GO TO
3000
3225 IF LOC=3 OR LOC=6 THEN PRIN
T "THE TAXI DRIVER SAYS SOMETHIN
G VERY RUDE TO ME.": GO TO C
3230 PRINT ""NOBBY APPEARS TO W
ANT TO KNOW.": GO TO C
3300 LET O=PI-PI: LET P=NOT O
3310 LET F=VAL "5": LET LOC=VAL
"13"
3320 LET OB=0: LET DO=0: LET GU=
P: LET UC=0: LET C=VAL "1000"
3330 DIM O(7): DIM O$(7,13)
3340 RESTORE 3000: FOR Z=P TO 7:
READ O$(Z),A$: LET O(Z)=VAL A$:
NEXT Z
3350 DATA "KNIFE","4","WALLET","
9","TUBE TICKET","10","BLOWTORCH
","16","KEY","19","WRITTEN CLUE"
,"20","CASE OF MONEY","25"
3350 PRINT "TAB 11: INK 6; "THE
CITY"" WRITTEN BY NEIL PELLIN
ACCI"" PRESS ANY KEY TO STA
RT"
3370 PAUSE P: PAUSE O: FOR Z=50
TO -20 STEP -5: BEEP .01,Z: BEEP
.01,Z-3: BEEP .01,Z+6: NEXT Z
3380 RETURN

```

```

8900 INPUT "ARE YOU SURE?
      Y/N": LINE A$: IF A$=""
      THEN GO TO 8900
8910 IF A$(P)="Y" THEN GO TO 900
9000 INPUT "DO YOU WANT TO PLAY
AGAIN? Y/N": LINE A$: IF A$=""
      THEN GO TO 9000
9010 IF A$(P)="N" THEN STOP
9020 RUN 900

```

## Doors of Doom

This is a masterpiece program by Malcolm Young which will well reward the time it will take you to type the whole thing in.

This graphic Adventure is based around the maze routine at lines 1000 to 1400. The routine starting at line 2000 handles the graphics.

When Malcolm started writing the program, he had intended it to be simply a 3D Maze program. However, once he had that up and running he decided he could make it a lot more interesting and the result is here for you to see. Full instructions are included. A word of warning: The program generates mazes which are extremely challenging, and difficult to solve. Do not run this program if you want a game you can master in an hour or so.

```

2 REM ©1983 Malcolm Young
10 DEF FN r(x)=INT (RND*x)+1
20 LET set=0
30 GO SUB 9000: REM instructio
40 BORDER 4: PAPER 5: INK 0: C
50 GO SUB 1000: REM generate m
60 GO SUB 8000: REM initialise
  variables
70 GO TO 200
80 NEXT a
90 REM main driving routine
100 LET k$=INKEY$: IF k$="" OR
k$="a" THEN GO TO 100
110 LET f=f+(k$="r")-(k$="l")+2
*(k$="b")

```

```

120 LET f=f+4*(f<1)-4*(f>4)
130 IF k$="m" THEN GO SUB 5000
135 IF k$="f" THEN LET m$(py,px)
) = "": LET l(11)=l(11)-l(13): I
f=l(11)<0 THEN GO TO 500
140 LET px=px+((k$="f")*((f=2)-
(f=4)))
150 LET py=py+((k$="f")*((f=3)-
(f=1)))
155 IF m$(py,px)="#" THEN LET s
#=#: LET px=px-(f=2)+(f=4): LET
py=py-(f=3)+(f=1)
160 IF m$(py,px)=CHR$ 128 THEN
GO SUB 6000
170 IF m$(py,px)> "." THEN GO TO
2600
175 IF g1=py AND g2=px THEN GO
TO 600
180 IF k$="e" THEN LET re=1-re:
LET k$="f": GO TO 140
190 IF k$="y" THEN GO SUB 7000
200 IF l(7) AND k$="d" THEN GO
SUB 7200
210 IF l(8) AND k$="t" THEN GO
SUB 6800
220 IF k$="s" THEN INPUT "Do yo
u want to save this game?"; LINE
a$: IF a$="y" THEN GO SUB 7500:
CLS
230 CLS
240 PRINT #0; INVERSE 1;"NORTH"
AND f=1;"EAST" AND f=2;"SOUTH"
AND f=3;"WEST" AND f=4; INVERSE
0;" "; FLASH 1;"REPEAT" AND re;
FLASH 0;TAB 15;"Energy=";l(11)
250 BEEP .1,10: GO SUB 2000
260 IF INKEY$="" AND (re#r) THE
N LET k$="f": GO TO 135
270 IF sw THEN PRINT AT 10,11;
FLASH 1;"SOLID WALL": BEEP 1,-10
: LET sw=NOT sw
280 GO TO 100
300 REM end of game
310 CLS
320 PRINT AT 10,0;"You have sta
ved to death..."
330 PRINT "Your bones litter t
he maze with those of many other
s..."
340 GO TO 4638
350 REM finish
365 PAPER 0: INK 7: CLS

```

```

610 GO SUB 5160: PRINT AT 4,0;
620 FOR i=40 TO 45 STEP .5: BEE
P .25,i: PRINT "Well done! You m
ade it!"; NEXT i
625 FLASH 1
630 PRINT INK 5;"You accomplish
ed the task with"
635 INK 4
640 IF l(10)>200 THEN PRINT "a
great amount of skill."
650 IF l(10)<200 AND l(10)>100
THEN PRINT "well deserved merit"
655 IF l(10)<100 THEN PRINT "a
considerable amount of luck"
660 GO TO 4642
1000 REM Maze Generator
1002 PAPER 0: INK 5: CLS : PRINT
"The maze can take a considerab
letime to construct so it is
advisable to have the larger
mazes stored on cassette."
1004 PRINT "Please enter the d
imensions of your maze so that
it has a width between 30 AND 100
and a length between 20 AND 100
."
1006 INPUT "Would you like to RE
STORE a gameanswer (y/n)? "; LI
NE a$
1008 IF a$="y" THEN GO SUB 7600:
RETURN
1010 INPUT "Enter Width of Maze:
";w: LET w=INT (w/2): IF w<15 OR
w>50 THEN GO TO 1010
1020 INPUT "Enter Length of Maze
";l: LET l=INT (l/2): IF l<10 O
R l>50 THEN GO TO 1020
1030 LET s=INT (w*(l/3)): LET s=IN
T (s+s/10): DIM l(s): DIM d(s)
1032 LET s=s+INT (RND*s/3): LET
t=INT (s/60): LET t1=(s-t*60)/10
0: LET t=t+t1
1033 POKE 23674,0: POKE 23673,0:
POKE 23672,0
1035 PRINT "Please expect to
wait at least approximately ";t
;" minutes"
1040 LET m=w*2+1: LET n=l*2+1: D
IM m$(n,m)

```

```

1050 FOR i=1 TO m: LET m$(1,i)="
#": LET m$(n,i)="#": NEXT i
1060 FOR i=2 TO n-1: LET m$(i,1)
=#": LET m$(i,m)="#": NEXT i
1070 LET nn=2*INT (RAND*(l-2))+3
1080 LET mm=2*INT (RAND*(w-2))+3
1090 LET m$(nn,mm)="#": LET k=0
1100 IF m$(nn-2,mm)="" THEN GO
TO 1210
1110 IF m$(nn+2,mm)="" THEN GO
TO 1210
1120 IF m$(nn,mm-2)="" THEN GO
TO 1210
1130 IF m$(nn,mm+2)="" THEN GO
TO 1210
1140 IF k-1=0 THEN GO TO 1420
1150 IF d(k)=1 THEN LET nn=nn+l(k)
1160 IF d(k)=2 THEN LET nn=nn-l(k)
1170 IF d(k)=3 THEN LET mm=mm+l(k)
1180 IF d(k)=4 THEN LET mm=mm-l(k)
1190 LET k=k-1
1200 GO TO 1100
1210 LET l1=2*INT (RAND*3)+2
1220 IF l1=8 THEN GO TO 1210
1230 LET d1=INT (RAND*4)+1
1240 IF d1>2 THEN GO TO 1280
1250 LET s=-1+2*(d1=2)
1260 LET t=0
1270 GO TO 1300
1280 LET s=0
1290 LET t=-1+2*(d1=4)
1300 FOR i=2 TO l1 STEP 2
1310 IF m$(nn+s*i,mm+t*i)="#" THEN
EN GO TO 1210
1320 NEXT i
1330 FOR i=1 TO l1
1340 LET m$(nn+s*i,mm+t*i)="#"
1350 NEXT i
1360 LET nn=nn+s*l1
1370 LET mm=mm+t*l1
1380 LET k=k+1
1390 LET l(k)=l1
1400 LET d(k)=d1
1410 GO TO 1100
1420 REM no. of doors

```

```

1425 LET nd=INT ((l(4)*l(5)/25):
LET st=nd/14
1430 LET l(4)=m: LET l(5)=n
1435 REM hide objects
1440 FOR o=1 TO 4
1450 FOR i=1 TO st
1460 LET l=FN r(l(5)-3)+1: LET c
=FN r(l(4)-3)+1: LET kn=FN r(5)*
10+0
1470 IF m$(l,c)="#" THEN GO TO
1460
1480 LET m$(l,c)=CHR$(48+kn)
1490 NEXT i
1500 NEXT o
1510 REM hide food and villains
1520 FOR o=5 TO 7 STEP 2
1530 FOR i=1 TO st*3
1540 LET l=FN r(l(5)-3)+1: LET c
=FN r(l(4)-3)+1: LET kn=FN r(5)*
10+0
1550 IF m$(l,c)="#" THEN GO TO
1540
1560 LET m$(l,c)=CHR$(48+kn)
1570 NEXT i: NEXT o
1580 REM hide treasure
1590 FOR i=1 TO st*4
1600 LET l=FN r(l(5)-3)+1: LET c
=FN r(l(4)-3)+1: LET kn=FN r(5)*
10+0
1610 IF m$(l,c)="#" THEN GO TO
1600
1620 LET m$(l,c)=CHR$(48+kn)
1630 NEXT i
1640 LET m$(l(5)/2,l(4)/2)=CHR$(
48+FN r(5)*10+8)
1650 REM hide keys
1660 FOR i=1 TO 4*st
1670 LET l=FN r(l(5)-4)+2: LET c
=FN r(l(4)-4)+2
1680 IF m$(l,c)="#" THEN GO TO
1670
1690 LET m$(l,c)=CHR$ 128
1700 NEXT i
1705 LET set=1
1710 LET time=(PEEK 23672+256*PE
EK 23673+65536*PEEK 23674)
1720 LET time=time/50: BEEP 2,10

```

```

1730 INPUT "This maze took ";(ti
#e1);" seconds to generate,would
you like to save it for future
use?" ;a$
1740 IF a$(1)="y" THEN GO SUB 75
90
1750 CLS : RETURN
2000 REM display view
2010 BORDER 1: PAPER 0: BRIGHT 1
2020 LET x=24: LET y=16: REM wal
l angle gradient
2030 FOR i=1 TO 6
2040 LET ly=py+((f=3)-(f=1))*i
2050 LET lx=px+((f=2)-(f=4))*i
2060 IF m$(ly,lx)<"#" OR m$(ly,(
x)-chr$(128) THEN NEXT i
2065 LET dt=(m$(ly,lx)=" ")
2070 LET vw=i-1-(i>6): REM maxim
um viewing distance
2075 IF NOT vw AND m$(ly,lx)>"#"
THEN GO SUB 2500
2080 LET wx=vw*x: LET wy=vw*y
2090 FOR d=0 TO vw: REM dividing
panels
2100 LET h=255-d*x*2
2110 PLOT d*x,d*y: DRAW 0,175-(y
+d)*2: IF d=vw THEN DRAW h,0
2115 PLOT d*x+1,d*y: DRAW 0,175-
(y+d)*2
2120 PLOT 255-d*x,175-d*y: DRAW
0,-(175-y*2*d): DRAW -h,0
2125 PLOT 254-d*x,175-d*y: DRAW
0,-(175-y*2*d)
2130 NEXT d
2140 FOR i=0 TO vw-1: REM draw w
alls
2150 LET ly=py+((f=3)-(f=1))*(i
+1)
2160 LET lx=px+((f=2)-(f=4))*(i
+1)
2170 LET d$m=m$(ly-(f=2)+(f=4),lx
-(f=1)+(f=3))
2175 LET lwall=(d$m="#"): LET ldr
=(d$m>"#")
2180 LET d$m=m$(ly+(f=2)-(f=4),lx
+(f=1)-(f=3))
2190 LET rwall=(d$m="#"): LET rdr
=(d$m>"#")
2195 IF (ldr+rdr) THEN GO SUB 23
00

```

```

2197 GO TO ldr*7+2200
2200 PLOT i*x,i*y+y*NOT (wall: 0
)
2205 PLOT i*x,175-(i*y+y*NOT (wa
ll): DRAW x,-(y*lwall)
2207 GO TO rdr*20+2210
2210 PLOT 255-i*x,i*y+y*NOT rwa
ll: DRAW -x,y*(rwall)
2220 PLOT 255-i*x,175-(i*y+y*NOT
rwall): DRAW -x,-(y*rwall)
2230 NEXT i
2240 INK 5: LET p=5
2250 LET v=255/p: LET e=(255-2*w
x)/p
2260 FOR l=1 TO p-1
2270 LET h=(lx+e*l)-v*l
2280 PLOT v*l,0: DRAW h,wy
2290 NEXT l
2310 IF dt THEN PRINT FLASH 1; 0
VER 1;AT 10,15;"015";AT 11,15;"02
"
2320 IF NOT re THEN RETURN
2325 LET r=1
2330 REM check for repeat
2340 FOR i=0 TO 1
2350 LET ry=py+((f=3)-(f=1))*i:
LET rx=px+((f=2)-(f=4))*i
2360 LET r=m$(ry+(f=2)-(f=4),
rx+(f=3)-(f=1))="#")
2370 LET r=m$(ry-(f=2)+(f=4),
rx-(f=3)+(f=1))="#")
2380 NEXT i: RETURN
2390 REM draw door
2400 LET k=1/(i+1)*5
2410 FOR j=1 TO 7
2420 IF ldr THEN PLOT i*x+j*3,i*
y+l*2
2425 IF ldr THEN DRAW INK 6;0,17
5-(i*y+j*2)*2
2430 IF rdr THEN PLOT 255-(i*x+j
*3),i*y+j*2
2440 IF rdr THEN DRAW INK 6;0,17
5-(i*y+j*2)*2
2450 NEXT j
2460 RETURN
2480 PLOT 100,100+k: DRAW k,0,-P
I*1.75
2490 DRAW k/2.5,-k*1.5: DRAW -k/
0.5*2-k,0,-PI: DRAW k/2.5,k*1.5
2491 RETURN

```



```

00500 REM door
00510 PAPER 6: INK 0: CLS
00520 PRINT AT 10,4;"D O O R":
00530 .5,10
00540 PAPER 0: INK 5
00550 RETURN
00600 REM enter
00610 IF L(12) THEN GO TO 3000
00620 PRINT AT 2,4; FLASH 1;"Sor
00630 key-no entry!"
00640 LET px=px-(f=2)+(f=4): LET
00650 py=py-(f=3)+(f=1)
00660 BEEP 1,-20: GO TO 100
00700 REM found centre
00710 IF py<>INT ((15)/2) AND px<
00720 >INT ((14)/2) THEN RETURN
00730 CLS: RESTORE 2900
00740 PRINT "The door creaks slow
00750 ly open..."
00760 FOR i=-30 TO -20 STEP .5: B
00770 EEP .1,i: NEXT i
00780 FOR i=1 TO FN r(8): READ t#
00790 : NEXT i
00800 PRINT "And there in the cen
00810 tre of the room lies the ultima
00820 te goal..."TAB 2;t#;"!"
00830 PRINT "And now you have to
00840 take this""back to an exit sit
00850 uated."
00860 LET m$(py,px)=" "
00870 LET g1=FN r(15): LET g2=FN
00880 r(4)
00890 LET m$(g1,g2)=" "
00900 GO SUB 5500
00910 LET b=FN r(100): LET l(11)=
00920 l(11)+b
00930 PRINT "You have also found
00940 a vitamin""pill boosting your
00950 energy by ";b
00960 BORDER 0: PAPER 0: INK 5
00970 PAUSE 0
00980 GO TO 200
00990 DATA "a slave for life","a
01000 ticket to Timbuktu","an automati
01010 c house cleaner","the LIONS rugb
01020 y team beating the NZ ALL BLACKS
01030 ","the answer to life...42?"
01040 DATA "a free ZX83,4,5,6..."
01050 "an eternal supply of lager","1
01060 million Mars bars"

```

```

3000 REM open door
3010 LET j=CODE #$(py,px)-48: LE
3020 T ln=INT (j/10)
3030 PRINT AT 0,3; INK 6;"The do
3040 or has lock no. ";ln
3050 IF l(9) THEN PRINT "Your s#
3060 elation keys opens the door!":
3070 PAUSE 100: GO TO 3070
3080 IF l(12)<>ln THEN PRINT AT
3090 01,0; INK 5;"Your key doesn't fi
3100 t,keep going": GO TO 2630
3110 PRINT AT 20,0;"Your key fit
3120 s! Do you want to unlock this
3130 door? (y/n)": BEEP 1,10
3140 IF INKEY$="n" THEN GO TO 20
3150
3160 IF INKEY$<>"y" THEN GO TO 3
3170
3180 GO SUB 2700
3190 LET fate=j-ln*10: BORDER 4:
3200 PAPER 6: INK 1: CLS
3210 IF fate=8 THEN GO TO 2700
3220 PRINT "INSIDE,there is...."
3230 : BEEP .5,3: BEEP 1,10
3240 GO TO fate*200+3090
3250 REM empty
3260 PRINT AT 3,5;"NOTHING!";AT
3270 5,0;"Some has been here before y
3280 ou"
3290 GO TO (RAND*.5)*80+3120
3300 REM lose key
3310 PRINT "Your key is stuck i
3320 n the lock!""Do you want to spe
3330 nd some time trying to get it o
3340 ut?"
3350 PAUSE 0: IF INKEY$<>"y" THE
3360 N GO TO 3200
3370 IF l(6) THEN PRINT "OK,it s
3380 houldn't take a second using y
3390 our laser.": LET l(6)=l(6)-1: LE
3400 T i=1: GO TO 3190
3410 PRINT "OK,I'll wait a momen
3420 t while you try."
3430 PAUSE 70: FOR i=0 TO 10+FN
3440 r(50): BEEP .01,RND*30-15: NEXT
3450 i
3460 IF RND*.7 THEN LET l(13)=l(
3470 13)-1: PRINT "It won't come out,
3480 you'll have to leave it.": GO TO
3490 3190

```

```

3185 PRINT "You got it!" "but ";
3190 PRINT "You wasted ";i;" ene
rgy units in""your attempt." : L
ET L(11)=L(11)-1
3200 GO TO 4900
3290 REM laser
3300 LET U=FN R(3)+2: PRINT AT 3
,6;Q$(1) "with enough EVERYREADY
batteries to last ";U;" uses!"
3310 IF L(6) THEN PRINT AT 3,0;"
Another "
3315 PRINT TAB 12;"■■■■"TAB 1
6;"■■■■"
3320 PRINT AT 9,0;"A laser has a
n 'EC' factor of 3 Do you want t
o take it?"
3330 IF INKEY$="n" THEN GO TO 48
90
3335 IF INKEY$("<")"y" THEN GO TO 3
330
3340 IF NOT L(6) THEN LET L(13)=
L(13)+9: GO TO 3360
3350 IF L(6) THEN PRINT "Since y
ou already have a laser you jus
t have to take the batteries
es"
3360 LET L(6)=L(6)+U: BEEP .2,3:
BEEP .2,10: BEEP .2,3: BEEP .7,
10
3370 PRINT "You are now the pro
vid owner of a disposable 'LEPPY'
laser""guaranteed for ";L(6);"
uses!"
3380 GO TO 4900
3490 REM dematerialiser
3500 PRINT AT 3,5;Q$(2);AT 3,0;"
Another " AND L(7)
3510 PRINT AT 5,11;"■■";TAB 5;"=■
■■■■"TAB 12;"■■■■"
3520 PRINT "This enables you to
disintegrate solid walls in two
blasts and to charcoal any villas
n you may meet."
3530 LET U=FN R(6)+2
3540 PRINT "This one has ";U;"
cartridges" Q$(2);"has an 'EC'""
factor of 14"
3550 PRINT "Do you want to carr
y this along?"

```

```

3560 IF INKEY$="n" THEN GO TO 48
90
3565 IF INKEY$("<")"y" THEN GO TO 3
560
3570 CLS : FOR I=0 TO 10: BEEP .
05,I: NEXT I
3580 PRINT FLASH 1;AT 2,9;"CONGR
ATULATIONS !"
3590 PRINT "You didn't need muc
h convincing for this formidable
weapon,as it's the hottest thi
ng around in armourments."
3600 IF L(7) THEN PRINT "You wo
n't want to carry around two of
these so you can have the carri
dges for free."
3610 PRINT "To use this weapon,
aim at your target and squeeze
the key 'd'"
3620 PRINT "You can use this we
apon ";L(7)+U;" times"
3630 PRINT INVERSE 1;"GOVERNME
NT WARNING: THIS WEAPON CAN BE D
ANGEROUS FOR YOUR HEALTH"
3670 LET L(13)=L(13)+14: LET L(7
)=L(7)+U
3680 GO TO 4900
3690 REM transporter belt
3700 PRINT AT 2,6;Q$(3);AT 2,0;"
Another " AND L(6)
3710 PRINT INK 4; PAPER 2;AT 4,1
5;"■■■■";AT 5,7; INVERSE 1;"+=:±
==■"; FLASH 1;"T"; FLASH 0; INU
ERSE 1;"■==:±==";AT 6,15;"■■■■"
3720 PRINT "This enables you to
hyper-travel to a different part
of the maze but you have no cont
rol where and if you stop in a
wall you will be instantly suff
ocated"
3730 LET U=FN R(3)+2
3740 PRINT "This is another mod
ern""disposable product and mus
t be discarded after ";U;" uses
or""transport failure may occu
r"
3750 PRINT Q$(3);" has an 'EC'
factor of 5"
3760 PRINT AT 20,3;"Do you want
to put this on?"

```



```

3770 IF INKEY$="n" THEN GO TO 49
3780 IF INKEY$<>"y" THEN GO TO 3
3790 CLS : LET L(8)=0: LET L(13)
=L(13)+5
3800 PRINT AT 1,0;"SOLD! To that
daring person sitting behind
the ZX SPECTRUM."
3810 PRINT "Use this belt no mo
re than ";L(8)"times." "To jump t
o hyperdrive press the button ma
rked 't'."
3820 PRINT INVERSE 1;"THE MANU
FACTURER WILL NOT BE RESPONSIB
LE FOR IMPROPER USE OF THIS TRA
NSPORTER BELT"
3830 FOR I=1 TO 5: BEEP .75,I: N
EXT I
3840 GO TO 4900

```

```

3890 REM skeleton key
3900 PRINT AT 2,0;Q$(4);"!"
3910 PRINT "TAB 15;";TAB 11;
"██████";TAB 15;";
3920 PRINT "This key gives you
access to anydoor in the maze!"
3940 PRINT "You can use it as m
any times as you like." "This ke
y has an 'EC' factor of 3"
3950 PRINT FLASH 1;"Do you want
to carry this along?"
3960 IF INKEY$="n" THEN GO TO 48
90
3970 IF INKEY$<>"y" THEN GO TO 3
980
3980 FOR I=1 TO 10: BEEP .1,5: P
AUSE 10: NEXT I
3990 CLS
4010 IF NOT L(9) THEN LET L(13)=
L(13)+3
4020 LET L(9)=1
4030 GO TO 4900
4090 REM food(energy)
4100 PRINT AT 2,10;"Food!"
4110 PRINT "You quickly rush to
the table" "and gobble up all t
he food."
4120 LET am=FN r(25)+10

```

```

4130 PRINT AT 10,3; INVERSE 1;"P
ress any key to find out";AT 7,0
;"The food has given you ";PAUS
E 0
4140 IF am>25+FN r(5) THEN PRINT
"Indigestion! You don't get any
energy points for that!"; BEEP
1,-20: GO TO 4900
4150 PRINT am;" valuable energy
points."
4160 LET L(11)=L(11)+am
4200 GO TO 4900
4290 REM treasure
4300 RESTORE 4480
4310 FOR I=1 TO FN r(5): READ t$
am: NEXT I
4320 PRINT AT 2,3;Q$(5);AT 2,17;
"!" "You have ";t$
4330 LET am=FN r(am)+5: LET en=I
NT (am/5)
4340 PRINT "It comes to the valu
e of ";am" treasure points and h
as an 'EC' factor of ";en
4350 PRINT "The treasure may be
useful to bribe the evil beas
ts that hide in these passages."
4360 PRINT "Do you want to carr
y this along?"
4370 IF INKEY$="n" THEN GO TO 48
90

```

```

4380 IF INKEY$<>"y" THEN GO TO 4
390
4390 LET L(13)=L(13)+en: LET L(1
0)=L(10)+am
4400 PRINT "You have now gained
";L(10);" treasure points."
4410 PRINT "Your 'EC' factor no
w adds up to ";L(13);". Each mov
e you make uses up ";L(13);" ene
rgy units."
4420 GO TO 4900
4490 DATA "struck gold!",60,"A c
hest of silver",30,"A sack of g
emming diamonds",40,"A plain
DUM sword",20,"A jar of rubies",
20
4490 REM evil villain
4500 RESTORE 4500

```

```

4510 DATA "An angry", "A crazy", "
A fierce", "An ugly", "A horrible",
"A giant", "A smelly", "A hungry"
4520 FOR i=1 TO FN r(8): READ a#
: NEXT i: RESTORE 4530
4530 DATA "gorilla", "demon", "dra
gon", "troll", "werewolf", "android
"
4540 FOR i=1 TO FN r(7): READ t#
: NEXT i
4550 PRINT INK 3; AT 2,5;a#;" "; t
#
4560 PRINT INK 1;"that's going t
o rip your arms off for waking
him up!"
4570 PRINT INK 0;"What are you
going to do?" "Are you going to:
4580 PRINT INK 1;"1/ fight""2/
bribe""3/ run""4/ pray he'll g
o away"
4590 IF INKEY#<"1" OR INKEY#>"4"
THEN GO TO 4590
4595 CLS
4600 GO TO VAL INKEY#*50+4560
4610 PRINT "You have ";
4612 IF l(6) THEN PRINT "a laser
";
4614 IF l(7) THEN PRINT "an anti
matter ray.";
4616 IF NOT (l(6)+l(7)) THEN PRI
NT "nothing.";
4618 PRINT "for a weapon."
4620 LET pr=((l(7)>0)+.8*(l(6)>0
)): IF NOT pr THEN LET pr=RND/2
4622 PRINT AT 21,1;"Press a key
to begin the fight": PAUSE 0
4624 FOR i=1 TO 5*(1/pr)
4626 LET cl=FN r(8)-1: BORDER cl
: PAPER cl: INK 7-cl: LET mc=USR
invert: BEEP .05,10-FN r(20)
4628 IF l(6) THEN BORDER 2: PAPER
6: BEEP .075,30: LET mc=USR in
vert
4630 IF l(7) THEN BORDER 0: PAPER
7: BEEP .1,0: LET mc=USR inver
t
4631 LET l(11)=l(11)-2
4632 NEXT i: BORDER 5: INK 4: PH
PER 0: BRIGHT 1: CLS
4634 IF RND<pr THEN GO TO 4648

```

```

4636 PRINT AT 5,1;"YOU LOST! The
";a#(3 TO );";t#";has thrown
your armless body in the corner
to join many others who had ve
ntured in this maze"
4638 FOR i=0 TO -30 STEP -1: BEE
P .5,i: NEXT i
4640 FLASH 1: CLS: GO SUB 5160
4642 POKE 23693,56: PRINT AT 12,
0;"Do you want to try again?"
4644 IF INKEY#="n" THEN CLS: ST
OP
4646 IF INKEY#="y" THEN GO TO 60
4647 GO TO 4644
4648 PRINT AT 5,10;"You won!"
4650 IF l(7) THEN PRINT "but wit
h the antimatter ray you couldn'
t lose!": LET l(7)=l(7)-1: GO TO
4900
4652 IF l(6) THEN PRINT "That wa
s handy work with the laser!"
: LET l(6)=l(6)-1: GO TO 4900
4654 PRINT "That was a pretty lo
cky blow." "the ";a#(3 TO );";
t#"; must be sick today."
4656 GO TO 4900
4660 PRINT "What are you going t
o give";a#;" ";t#;"?"
4665 PRINT "You have with you:"
: FOR a=1 TO 6: PRINT q$(a) AND
l(a+5): NEXT a
4670 DIM g$(18): INPUT g#
4672 FOR i=1 TO 6: BEEP .1,5: IF
g#=q$(i) AND l(5+i) THEN GO TO
4900
4674 FOR i=1 TO 6: BEEP .1,5: IF
g#=q$(i) AND l(5+i) THEN GO TO
4900
4676 NEXT i: PRINT AT 12,0;"Sorr
y, I didn't understand that. Try
again!": BEEP .5,5: PAUSE 0: DIM
g$(64): PRINT AT 12,0;g#: GO TO
4670
4680 IF i=5 THEN GO TO 4690
4682 PRINT "The not so ";a#(3 T
O );";t#";says thank you and t
akes the ";q$(i,3 TO );"and sudde
nly disappears"
4684 LET l(i+6)=0: LET l(13)=l(1
3)-e(i)
4686 GO TO 4900

```

```

4690 INPUT "How much treasure are
you going to give the ";(t$);
:am
4693 IF am>0 THEN GO TO 4700
4694 PRINT "The ";t$;" is enrage
d that you tried to cheat him an
d has""taken everything, except
half""your treasure."
4695 FOR i=0 TO -20 STEP -1: BEE
P .05,i: NEXT i
4696 LET t=l(11)/2: LET l=l(10):
LET m=l(4): LET n=l(5): DIM l(1
3)
4697 LET l(13)=1: LET l(4)=m: LE
T l(15)=n: LET l(10)=l: LET l(11)
=t
4698 GO TO 4900
4700 IF am>FN r(20)+10 THEN PRIN
T "The ";t$;" accepts your measu
re ""offering and tells you he wi
ll be more ""a$(3 TO ):" to you
next""time." : BEEP 1,10: GO TO
4900
4702 PRINT "The ";t$;"insulted b
y what you have""offered,gets m
ore ""a$(3 TO )""and steals your
";
4704 LET o=FN r(5): IF o=5 THEN
LET o=6
4706 LET l(13)=l(13)-e(o): LET l
(o+5+(o=6))=0
4708 PRINT q$(o,3 TO )
4709 GO TO 4900
4710 IF l(8) THEN PRINT "You act
ivate your transporter and who
osh! You're away!": BEEP 1,50: G
O TO 6500
4711 FOR i=0 TO FN r(40)+10 STEP
.5: BORDER FN r(7): BEEP .01,i:
LET l(11)=l(11)-.5: NEXT i: FOR
i=0 TO FN r(50): BEEP .01,-i: N
EXT i
4714 LET r=2-FN r(3)
4718 IF NOT r THEN GO TO 4730
4722 PRINT "Not fast enough!""t$
""s are pretty fast""sometimes
""You can only fight now."
4725 GO TO 4610
4735 PRINT "You made it! Why are
n't you""representing your coun
try in""running?"

```

```

4740 PRINT "After a quick 100 me
tre sprint around the corner yo
u find""yourself back in the sa
me place"
4745 PRINT AT 20,0:"After you ha
ve had a rest,press a key."
4750 BEEP .5,-50: BEEP .3,-40
4755 IF INKEY$="" THEN GO TO 475
0
4758 GO TO 4900
4760 FOR i=0 TO 5: PAPER FN r(7)
: BEEP .1,50: CLS : BEEP .5,-50:
NEXT i
4765 PAPER 6: INK 0: CLS
4770 PRINT "It didn't work,the ""
t$""is still there!"
4773 LET w=FN r(10): LET l(11)=l
(11)-w
4775 PRINT "You wasted ""w;"" ene
rgy units""trying though."
4780 GO TO 4570
4890 PRINT "OK,we'll leave it."
4900 GO SUB 8400: REM wait
4920 BEEP 1,5
4930 BORDER 0: PAPER 0: INK 5: 5
RIGHT 1: CLS
4940 LET m$(py,px)=" "
4950 GO TO 230
5000 REM map
5005 IF l(10)<10 THEN PRINT INK
0:AT 0,0:"You don't have enough
treasure to pay the fee,no more
y,no map.": BEEP .5,-30: GO TO 1
00
5010 BORDER 5: BRIGHT 0: CLS : P
RINT AT 4,9: INK 5:"***NORTH**
*"
5020 IF INKEY$<>"" THEN GO TO 50
00
5030 LET mx=(px-6)*(px>5)+1: LET
my=(py-6)*(py>5)+1
5040 IF mx+10>l(4) THEN LET mx=l
(4)-10
5050 IF my+10>l(5) THEN LET my=l
(5)-10
5055 LET d$=m$(py,px): LET m$(py
,px)="*"
5070 FOR l=my TO my+10: PRINT IN
K 6;TAB 9;"*";
5080 FOR c=mx TO mx+10

```

```

5000 IF CODE M$(L,C)>48 AND CODE
M$(L,C)<128 THEN PRINT "E": GO
TO 5120: REM print door
5100 IF M$(L,C)>"5" THEN PRINT F
LASH 1; INK 6;"E": GO TO 5120
5110 PRINT M$(L,C);
5120 NEXT C: PRINT INK 6;"*"
5130 NEXT L: PRINT INK 6;TAB 9;"
****SOUTH****"
5140 LET L(13)=L(13)-2: LET L(10)
=L(10)-10: REM pay $10 fee for
door
5150 LET M$(PY,PX)=d$
5160 PRINT INK 4; BRIGHT 1; AT 0,
0;"Energy left=";L(11)"Treasure
points=";L(10)"You energy cons
umption per move is ";L(13)
5170 PRINT INK 5; AT 17,0;"You ar
e carrying:"
5180 IF L(6) THEN PRINT INK 6; q$(
1);L(6);" shots left"
5190 IF L(7) THEN PRINT q$(2);L(
7);" shots left"
5200 IF L(8) THEN PRINT INK 5; q$(
3);" Life of ";L(8)
5210 IF L(9) THEN PRINT INK 4; q$(
4)
5215 IF L(12) THEN PRINT INK 5;"
Key"
5220 IF NOT (L(12)+L(6)+L(7)+L(8)
)+L(9) THEN PRINT FLASH 1;"Noth
ing"
5225 IF INKEY$<>" " THEN GO TO 52
27
5227 GO SUB 5500
5230 PAUSE 500: RETURN
5235 IF NOT a1 THEN RETURN
5240 PRINT "The EXIT is "; "NORTH
" AND PY>g1;"SOUTH-" AND PY<g1;
5245 PRINT "WEST" AND PX>g2;"EAS
T" AND PX<g2;" of here."
5250 RETURN
5255 REM key
5260 PRINT AT 1,5;"You have found
a key!"
5265 BEEP .75,10
5270 IF NOT L(12) THEN PRINT AT
1,3;"Do you want to pick it up?"
5275 IF L(12) THEN PRINT AT 21,3
;"Do you want to swap keys?"

```

```

6050 LET M$(PY,PX)=" ": REM spac
e
6060 IF INKEY$="n" THEN RETURN
6070 IF INKEY$<>"y" THEN GO TO 6
060
6075 IF L(13)<1 OR NOT L(12) THE
N LET L(13)=L(13)+1
6080 BORDER 2: PAPER 4: INK 0: C
LS
6090 LET key=FN r(5): IF key=L(1
2) THEN GO TO 6090
6095 LET L(12)=key
6100 BEEP .1,3: BEEP .4,10: BEEP
.2,3: BEEP .5,10
6110 PRINT AT 10,2;"YOU NOW HAVE
A KEY MARKED ";L(12)" Guard
it closely your life may
depend on it."
6120 GO SUB 8400
6140 BORDER 0: PAPER 0: INK 5: C
LS
6150 RETURN
6200 REM transporter
6210 LET PX=FN r(L(4)): LET PY=F
N r(L(5)): LET F=FN r(4)
6220 FOR I=0 TO 10: LET CL=FN r(
0)-1: BORDER CL: PAPER 7-CL: BEE
P .01,10-FN r(20): CLS : NEXT I
6230 FOR I=1 TO 5
6240 LET CL=FN r(8)-1: BEEP .1,4
0
6250 BORDER CL: PAPER 7-CL: CLS
6260 PAUSE 20: POKE 23693,56: CL
S
6270 NEXT I: BORDER 1
6280 PRINT AT 6,0;"You have now
been transported" "to a differen
t part of the maze"
6290 LET L(8)=L(8)-1: IF NOT L(8)
THEN LET L(13)=L(13)-5
6300 GO SUB 8400: RETURN
7000 REM drop object
7005 PAPER 6: INK 0: BORDER 4: C
LS
7010 INPUT ("What do you want to
leave" n$;"? "); g$
7020 IF g$(1)=" " THEN RETURN
7030 FOR I=1 TO 6
7040 IF g$=q$(I) THEN GO TO 7070
7050 BEEP .1,10-FN r(20): NEXT I

```

```

7060 PRINT "Sorry,I don't know w
hat ";g$;"is,try again.": GO TO
7070
7070 IF l(i+5) THEN GO TO 7090
7080 PRINT "But you don't have "
;g$;"Try again": GO TO 7070
7090 IF i=5 THEN GO TO 7140
7100 PRINT "OK,say goodbye to yo
ur " ;g$(3 TO )
7110 LET l(13)=l(13)-e(i)
7120 LET l(i+5)=0
7130 BEEP 1,10: GO TO 7190
7140 INPUT "How much treasure do
you want toleave? ";tr
7150 IF tr>l(10) THEN PRINT "You
don't have that much""treasure
";n$: BEEP 1,-10: GO TO 7140
7160 LET l(13)=l(13)-tr/5: LET l
(10)=l(10)-tr
7170 PRINT "Say goodbye to ";tr;
" points of""treasure"
7180 IF l(13)<=0 THEN LET l(13)=
0
7190 BORDER 0: PAPER 0: INK 5: C
LS: RETURN
7200 REM dematerialisers
7210 LET dy=py+(f=3)-(f=1): LET
dx=px+(f=2)-(f=4)
7212 IF dy>l(5)-1 OR dy<2 OR dx<
2 OR dx>l(4)-1 THEN PRINT "Sorry
you can dematerialise thisouter
wall": PAUSE 0: RETURN
7215 LET m$(dy,dx)=" "
7220 FOR i=40 TO 55: LET cl=8-FN
r(6)
7230 PAPER cl: BORDER 7-cl: CLS
7240 BEEP .05,i: NEXT i
7250 LET l(7)=l(7)-1
7260 BEEP .5,-20: RETURN
7500 REM save game
7505 LET l(1)=py: LET l(2)=px: L
ET l(3)=f
7510 PRINT FLASH 1;AT 16,7;"Savi
ng Maze Data"
7520 PRINT INVERSE 1;"Filename "
"maze""";AT 20,0;"Please keep pr
essing a key untilthe screen cle
ars"
7530 SAVE "maze" DATA l(): CLS:
SAVE "maze" DATA m$()
7540 INPUT "Verify?";a$: IF a$<>
"y" THEN RETURN

```

```

7550 PRINT "Rewind cassette,then
PLAY""If loading error then t
ype:GO TO 7500"
7560 VERIFY "maze" DATA l(): VER
IFY "maze" DATA m$()
7570 RETURN
7600 PRINT FLASH 1;AT 10,9;"Load
ing Data"
7620 PRINT #0; FLASH 1;AT 1,4;"S
tart cassette recorder"
7630 LOAD "maze" DATA l(): LOAD
"maze" DATA m$()
7640 LET py=l(1): LET px=l(2): L
ET f=l(3)
7650 CLS
7660 RETURN
8000 REM variables
8010 IF NOT set THEN GO TO 8070
8020 LET px=2: LET py=px: LET f=
0: REM player position in maze
8030 DIM l(13): LET l(4)=m: LET
l(5)=n
8040 LET l(10)=200: REM treasure
8050 LET l(13)=1: REM energy con
sumption
8060 LET l(11)=INT (l(4)*l(5)/5)
8070 LET r=0: LET re=r: REM repe
at
8080 LET g1=0: LET sw=0
8090 REM define graphics
8100 RESTORE 8250
8110 FOR i=1 TO 7: READ c$
8120 FOR e=0 TO 7: READ n: POKE
USR c#+e,n
8130 NEXT e: NEXT i
8140 FOR i=USR "s" TO USR "s"+18
8150 READ b: POKE i,b: NEXT i
8160 LET invert=USR "s"
8170 DIM q$(6,18): DIM e(6)
8180 FOR i=1 TO 6: READ a$,e: LE
T q$(i)=a$: LET e(i)=e: NEXT i
8190 BORDER 0: PAPER 0: INK 5: 0
RIGHT 1: CLS
8200 RETURN
8250 DATA "a",0,0,3,15,12,12,12,
0
8260 DATA "b",0,0,126,224,96,48,
40,96
8270 DATA "c",1,1,1,1,0,1,1,0
8280 DATA "d",224,192,126,126,0,
126,126,0

```

```

0000 DATA "e",255,129,129,129,12
0001 DATA "k",0,2,3,255,67,226,6
0010 DATA "h",24,60,126,60,24,24
0020 DATA 33,0,64,6,24,197,6,0,1
0030 DATA 255,119,35,16,249,193,16,
0040 DATA "a laser",9,"an anti-ma
0050 "ter ray",14,"a transporter belt",
0060 "a skeleton key",3,"some trea
0070 "sure",0,"a key",1
0080 PRINT AT 21,5; FLASH 1; INK
0090 "PRESS 'c' TO CONTINUE."
0100 IF INKEY$ <> "c" THEN GO TO 8
0110 BEEP 1,10: RETURN
0120 REM instructions
0130 POKE 23609,100
0140 INPUT "Hullo! Before we sta
0150 rt, please "enter your name:"; L
0160 NE n$: IF n$="" OR n$="no" OR n
0170 $="yes" THEN GO TO 9005
0180 IF n$(1) > "Z" THEN LET n$(1)
0190 =CHR$(CODE n$-32)
0200 BORDER 1: INK 1: PAPER 6: C
0210 FOR b=0 TO 10
0220 PLOT 127,175: DRAW b*25-125
0230 PLOT 127,0: DRAW b*25-125,1
0240 NEXT b: POKE 23609,50
0250 PLOT 0,112: DRAW 255,0
0260 PLOT 0,70: DRAW 255,0
0270 PRINT AT 3,0:
0280 OVER 1: INK 2
0290 PRINT TAB 7;" "
0300 PRINT TAB 7;"D_o_o_r_"
0310 PRINT TAB 7;"|_|_|_|_|"
0320 PRINT TAB 7;"_|_|_|_|"
0330 PRINT
0340 PRINT TAB 10;"O_f_"
0350 PRINT TAB 10;"O_f_"
0360 PRINT TAB 10;"O_f_"
0370 PRINT TAB 10;"O_f_"
0380 PRINT

```

```

0120 PRINT TAB 7;" "
0130 PRINT TAB 7;"D_o_o_r_"
0140 PRINT TAB 7;"|_|_|_|_|"
0150 PRINT TAB 7;"_|_|_|_|"
0160 INK 1: OVER 0: BEEP .4,20:
0170 BEEP .5,5: BEEP .5,10
0180 INPUT "Do You Need Instruct
0190 ions (y/n)?"; LINE a$: IF a$="n"
0200 THEN RETURN
0210 BORDER 5: PAPER 1: INK 4: B
0220 RIGHT 1: CLS
0230 PRINT TAB 8; FLASH 1;"DOO
0240 RS OF DOOM"
0250 PRINT "GREETINGS to the DOO
0260 RS OF DOOM" FLASH 1;"?";CHR$ 8;
0270 PRINT n$;" "
0280 PRINT "You have come here i
0290 n search of the Ultimate Goal wh
0300 ich lies beyond one of the DO
0310 ORS OF DOOM at the heart of the
0320 unwelcoming labyrinth."
0330 PRINT "You must journey th
0340 rough the "long deceiving passa
0350 ges in "search of the right rou
0360 te to the interior."
0370 PRINT "Throughout the maze
0380 there are "doors leading to an
0390 adjacent "passage, and behind th
0400 ese doors may lie DANGER...or h
0410 ope?"
0420 GO SUB 8400: PAPER 0: CLS
0430 PRINT "The doors are locked
0440 and can "only be opened if you
0450 possess a matching key, which ar
0460 e littered around the labyrinth."
0470 PRINT INVERSE 1 "DIRECTIONS
0480 " : INK 5
0490 PRINT "You may use a map w
0500 hich shows "the surrounding wal
0510 ls and the "path you have taken
0520 . It will also show up any keys an
0530 d doors "nearby"
0540 PRINT "Each time you ask fo
0550 r a map you lose 10 treasure poi
0560 nts"
0570 GO SUB 8400: BORDER 0: CLS

```

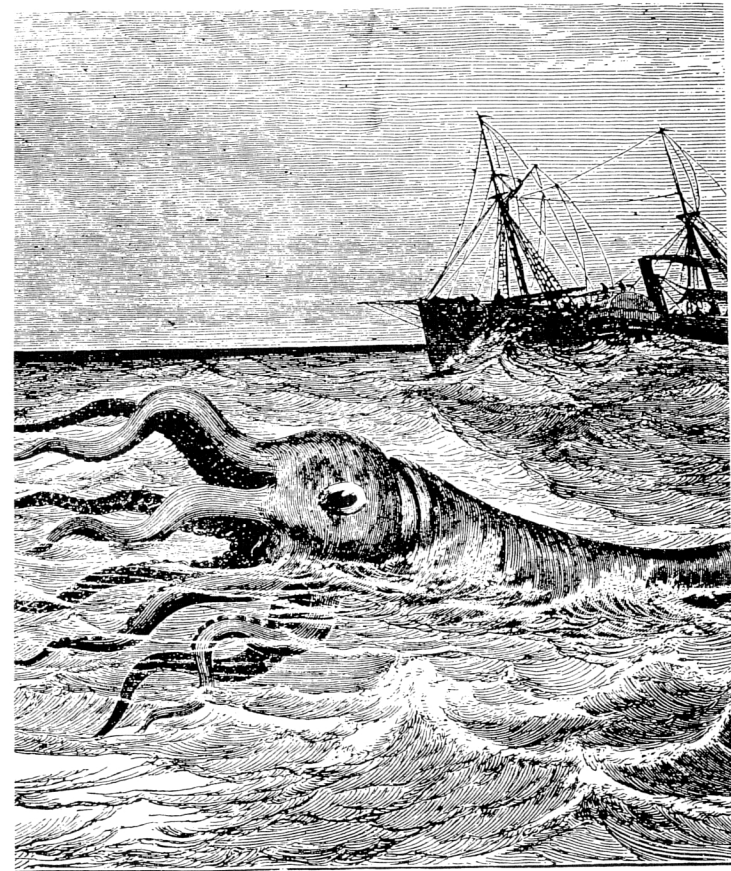


```

0320 PRINT TAB 8; "KEY words": P
LOT 70,167: DRAW INK 4;24,0
0330 PRINT "Keys 'L' for left,
'R' for right and 'B' for back."
0340 PRINT "Key 'F' to step form
ard."
0350 PRINT "Key 'S' to save game
Key 'E' to repeat.(if you com
e to a long corridor?"
0360 PRINT INK 5;"Each object yo
u pick up has its own Energy Con-
sumption or 'EC' factor.This is
the amount of "energy used aft
er each step."
0370 PRINT INK 7;"You may specif
y the size of mazeto be Used,the
largest maze willtake up to 20
minutes to producewhile the smal-
lest will take""about 1.2 minut
es."
0380 PRINT INK 5;"You may save t
he mazes for""future use."
0390 GO SUB 8400
0400 RETURN

```

# MOVING GRAPHICS GAMES



## Gold Rush

Your task in this game, written by Neil Pellinacci, is to carry six sacks of gold from a bank vault to a safe. The vault and safe are separated by a river. The river can be crossed via a network of small dams, which are built across it. Unfortunately, the future of this link is not certain, because occasional logs drifting down the river will destroy parts of the dam system. You must transfer all the sacks to the safe before your route is cut off.

You play the game using the following control keys:

Q - to move up  
Z - to move down  
I - to move left  
P - to move right

Your current task is shown by the color of your man. White indicates that you have to pick up a sack, and yellow shows you are carrying a sack. To pick up a sack, you just move onto the same square as it. You put it in the safe by moving onto the safe. In either case, a beep will sound, and the man will change color. You have three men per game, and the number left is shown at the top of the screen.



A log will remove any section of dam in its path. However, sections above and below are also damaged, and if you move off one of these, and into part of the river where a log has been, you will lose one life. This is the only way you can lose a life, so be very careful after the first few logs have appeared.

When all six sacks have been put in the safe, you move onto the next level. On this level, there are two dam formations, and these alternate. As the game progresses, the number of logs increases, until finally you'll find the game is virtually impossible.

You'll see at this point the dam is almost nonexistent, and you'll have to resort to new tricks. You can 'hop' onto a log immediately above or below. It is difficult at first, but not impossible. Remember, you don't move with the log; you are just using it as a piece of dam. If you are very careful, you can even move right and left using the logs.




The end of the game is the hardest part, and it is the main reason for getting as high a score as you can before the dam is destroyed.

Scoring is simple. You get 100 points for going from the safe to the bank vault and another 100 for successfully bringing back a sack of gold. If you lose a life while

carrying the sack, you do not score. Neil's highest score is around 3300, which gives you a target to shoot for.

The program is written mainly in BASIC, but it uses a machine code routine located at 32530, which moves the logs right and left across the screen, stopping them after a collision with a dam wall. The data for this machine code is held in the last two lines of the program.

Here's the key to the user-defined graphics:

A =   
B =   
C = 

This is the program's structure:

100: subroutine to print score  
1000 - 1030: game preparation  
1040 - 1045: initialisation for each game  
1050 - 1100: initialisation for each level  
of the game  
1510 - 1560: main loop  
1600 - 1630: detects collisions  
1700 - 1740: end of one level  
1800 - 1820: lose one man routine  
2000 - 2100: game over routine  
6000 - 6650: draw dams across river  
7000 - 7100: draw river and the banks  
8000 - 8100: title page and instructions  
9000 - 9120: POKE user graphics characters  
9200 - 9310: POKE machine code into memory

And here's the listing:

```
10 CLEAR 32529: RANDOMIZE : GO
TO 1000
99 REM SCORE
100 PRINT AT 0,11-LEN STR$ S; I
NK 7; PAPER 2; BRIGHT 1; G: RETUR
N
1000 REM SETUP
1010 BORDER 2: PAPER 4: CLS : GO
SUB 9000
1020 PRINT AT 0,0; INK 7; PAPER
0; "SCORE "; BRIGHT 1; "00000"; BR
IGHT 0; " MEN "; BRIGHT 1; "0"; BR
IGHT 0; " HI-SCORE "; BRIGHT 1; "0
0000"
1030 LET HS=0
1040 GO SUB 9000: PRINT AT 0,0;
PAPER 2; INK 7; BRIGHT 1; "00000"
; AT 0,10; "0"
1045 LET MM=0: LET MEN=0: LET S=
0: LET RN=.999
1050 LET SK=0: LET L=2: LET C=23
: LET G=7
1100 GO SUB 7000: GO SUB 6000
1500 REM LOOP
1510 LET L1=L: LET C1=C
1517 LET A#=INKEY$: IF A#="" THE
N GO TO 1555
1520 LET L=L+(A#="Z")-(A#="0")
1530 LET C=C+(A#="P")-(A#="I")
1540 LET AT=ATTR (L,C): IF AT<>2
4 THEN GO TO 1500
1545 BEEP .0025,10
1550 PRINT AT L1,C1; PAPER 3; IN
K 0; " "; AT L,C; INK 0; PAPER 0; "
*"
1555 IF AND>RN THEN PRINT AT 7+I
NT (AND*9),30; INK 0; PAPER 0; "
": LET RN=RN-.0005
1560 LET Z=USR 32530: GO TO 1510
1560 IF AT=30 THEN IF G=7 THEN L
ET G=6: BEEP .1,20: LET S=S+100:
GO SUB 100: LET SK=SK+1: GO TO
1550
1610 IF AT=31 THEN IF G=6 THEN B
EEP .1,40: LET G=7: LET S=S+100:
GO SUB 100: IF SK=6 THEN GO TO
1700
1620 IF AT=45 THEN GO TO 1600
1625 IF AT=0 THEN GO TO 1545
```

```
1630 LET L=L1: LET C=C1: GO TO 1
500
1700 FOR A=-10 TO 20 STEP 5: BEE
P .1,A: BEEP .1,A-5: NEXT A
1710 FOR A=15 TO -10 STEP -5: BE
EP .1,A: BEEP .1,A-5: NEXT A: BE
EP .1,-15
1720 FOR A=1 TO 21: PRINT AT A,0
: PAPER A/3; "
": NEXT A
1730 BEEP .1,20: BEEP .1,25: FOR
A=1 TO 21: PRINT AT A,0; PAPER
0; "
": NEXT A
1740 GO TO 1000
1800 FOR A=10 TO -25 STEP -4: PR
INT AT L,C: PAPER 5; INK 0; "A"; A
T L,C: PAPER 5; INK 5; " ": BEEP
.01,A: NEXT A
1805 PRINT AT L1,C1; PAPER 3; IN
K 0; " "
1810 LET MEN=MEN-1: PRINT AT 0,1
0; PAPER 2; INK 7; BRIGHT 1; MEN
1815 IF MEN=0 THEN GO TO 2000
1817 IF SK=6 THEN GO TO 1700
1820 LET L=2: LET C=23: LET G=7:
GO TO 1500
2000 REM GAME OVER
2010 FOR A=0 TO 7: FOR B=1 TO 21
: PRINT AT B,0; PAPER A; "
": NEXT
A
2020 FOR A=1 TO 10: BEEP .1,0: B
EEP .1,0: BEEP .1,10: NEXT A
2030 IF S>HS THEN LET HS=S: PRIN
T AT 0,30-LEN STR$ HS; PAPER 2;
INK 7; BRIGHT 1; HS: BEEP .1,2: B
EEP .2,10
2035 FOR A=20 TO 0 STEP -1: PRIN
T AT A,0; BRIGHT 1; PAPER AND*7;
INK 9; " GAME OVER GAME OVER GAM
E OVER " : NEXT A
2040 FOR A=0 TO 7: BORDER A: BEE
P .2,7-A: NEXT A: FOR A=7 TO 0 S
TEP -1: BORDER A: BEEP .2,7-A-1:
NEXT A
2100 GO TO 1040
5000 STOP
6000 REM BRIDGES
6010 IF MM=0 THEN GO TO 6600
6020 LET MM=0
```

```

6050 PAPER 3: INK 0: PRINT AT 7,
6060 PRINT AT 8,9: "":AT 8,10: "
:AT 8,15: "":AT 9,9: "":AT 9,15
6070 PRINT AT 9,9: "":AT 9,15
:AT 10,9: "":AT 10,17:
"
6080 PRINT AT 11,9: "":AT 11,10:
:AT 11,12: "":AT 11,14: "
:AT 12,14: "":AT 12,16: "":AT
12,18: "":AT 13,10: "":AT 13,14
6090 PRINT AT 13,10: "":AT 13,14
:AT 13,21: "":AT 14,6: "":A
6100 PRINT AT 14,6: "":AT 14,18: "
:AT 15,9: "":AT 15,16: "
6500 RETURN
6600 PAPER 3: INK 0: PRINT AT 7,
19: "":AT 7,20: "":AT 8,10: "
6610 PRINT AT 9,10: "":AT 9,18: "
:AT 9,16: "":AT 9,18: "":A
T 10,10: "":AT 10,10: "":AT 10,1
6: "":AT 10,20: "":AT 11,10: "":AT
6620 PRINT AT 11,10: "":AT 11,20: "
:AT 12,10: "":AT 12,20: "":AT 13,1
6: "":AT 13,20: "":AT 14,9: "
6630 PRINT AT 13,9: "":AT 13,1
0: "":AT 13,20: "":AT 14,9: "
:AT 14,26: "":AT 15,1
6640 PRINT AT 15,9: "":AT 15,1
7: "
6650 LET MM=1: RETURN
7000 REM SCREEN
7010 FOR A=1 TO 21: PRINT AT A,0
: PAPER 4: "": NEXT A
7015 FOR A=6 TO 18: PRINT AT A,0
: PAPER 5: INK 0: "": NEXT A
7020 INK 0: PAPER 6: FOR A=1 TO
3: PRINT AT A,23: "": NEXT A
7030 PRINT AT 2,9: "":AT 3,18: "":AT 4,9
:AT 4,18: "":AT 5,10: "":AT 5,21: "
:AT 6,10: "":AT 6,21: "":AT 10,10: "
7050 PRINT AT 10,10: "":AT 10,16
: "
7060 PRINT AT 17,9: "":AT 18,23: "":A
T 20,14: "

```

```

7070 FOR A=19 TO 21: PRINT AT A,
0: "": NEXT A
7080 FOR A=19 TO 21: PRINT AT A,
0: INK 6: "": NEXT A: PRINT AT
0,25: INK 7: "0"
7100 RETURN
8000 REM TITLE PAGE
8005 BORDER 3
8010 FOR A=1 TO 21: PRINT AT A,0
: PAPER 6: "": NEXT A
8012 DIM I(8): FOR A=1 TO 8: LET
I(A)=A-1: NEXT A
8013 FOR A=1 TO 8: LET II=INT (R
ND*8)+1: LET AA=I(A): LET I(A)=I
(II): LET I(II)=AA: NEXT A
8015 GO TO 8030
8016 FOR A=1 TO 8: INK I(A): BOR
DER I(A): PRINT AT 4,7: "XXXXXXXXXX
XXXXXXXXXX"
8020 PRINT AT 6,7: "XXXXXXXXXXXXXXXXXX
XXXXXXXXXX"
8022 PRINT AT 5,7: "X":AT 6,24: "X
": BEEP .4, I(A)+10
8024 IF INKEY#0 THEN BORDER 2
: RETURN
8026 NEXT A: GO TO 8016
8030 PRINT AT 10,10: PAPER 6: IN
K 6: BRIGHT 1: "GOLD PUSH "
8040 INK 0: PAPER 6: PRINT AT 12
,12: "CONTROLS": PLOT 0,75: DRAW
95,0: PLOT 255,75: DRAW -95,0
8050 INK 1: PRINT AT 14,9: "0....
.....UP":AT 15,9: "Z.....DO
WN"
8060 PRINT AT 16,9: "I.....LE
FT":AT 17,9: "P.....RIGHT"
8070 PAPER 6: INK 0: PLOT 0,24:
DRAW 255,0
8080 PRINT AT 19,6: PAPER 4: INK
7: BRIGHT 1: "PRESS ANY KEY TO P
LAY"
8090 PRINT AT 21,0: PAPER 2: INK
7: "Written by Neil Pellinacci
1980"
9100 GO TO 8016
9000 REM UCD
9005 RESTORE 9100
9010 FOR A=USR "A" TO USR "A"+23
: READ B: POKE A,B: NEXT A

```

```

0100 DATA 0,125,60,24,50,125,255
-100
0110 DATA 255,125,125,125,127,12
0,255,100
0120 DATA 15,55,15,124,15,40,55,
00
0200 REM MOODE
0210 FOR A=0 TO 47: READ B: POKE
0220+A,B: NEXT A
0300 RETURN
0300 DATA 33,102,88,5,11,14,31,3
5,105,254,0,50,25,54,45,43,105,2
54,24
0310 DATA 50,4,54,45,54,2,54,0,3
5,10,30,25,25,15,20,4,30,100,30,
17,50,0,0,14,54,45,25,20,251,201

```

## Tarantula

In Malcolm Young's great program TARANTULA you are in a dark pit about the size of a football field, fighting for your life against dreaded tarantula spiders.

The tarantulas slide down the sides of the pit looking for food, and human beings fit into the category of food. You're equipped with a torch and a laser, and you move around the pit in a fairly flimsy tank.

Instructions are within the program, and you'll find you'll quickly work out how to play it once you get the program up and running.

```

1 REM graphic characters: REM
2 B=# C=# D=# E=# F=# G=# H=#
3 I=# J=# K=#
4 DIM P$(15): DIM W$(5,15): D
5 IN 4(5)
6 PRINT "Please wait just a m
7 oment.": GO SUB 8615: REM define
8 characters
9 GO SUB 9000
10 BORDER 5: PAPER 0: INK 7: C
11
12 PRINT AT y,x:t#
13 PRINT INK 0; PAPER 5; AT 21,
14 SCORE=:score; AT 21,12: "SPIDE
15 :;ab;": LIVES=:lives
16 LET a=x: LET b=y
17 LET i#=INKEY#
18 GO TO (i#<"1" OR i#>"8")*20
19
20 BEEP .05,0: GO SUB VAL i#*1
21
22 PRINT AT b,a;" "; BRIGHT 1;
23 AT y,x: INK r;k;t#
24 GO SUB (RND<lev)*100+100
25 GO TO (d#(y+1,x+1)="*")*390
26

```

```

110 LET cn=cn*(cn<n)+1
120 GO TO (n>0)*65+40
125 IF NOT s(cn,1) THEN GO TO (
n<1)*70+40
130 LET sx=s(cn,1): LET sy=s(cn
n)
140 LET d$(sy,sx)=" "
150 LET sy=sy+(sy<y+1)-(sy>y+1)
: LET s(cn,2)=sy
160 LET sx=sx+(sx<x+1)-(sx>x+1)
: LET s(cn,1)=sx
170 LET d$(sy,sx)="*"
180 GO TO 40
9000 REM new tarantula
9010 IF n=15 THEN LET lev=.05: R
RETURN
9020 LET n=n+1: LET nb=nb+1
9030 LET sx=s(n,1): LET sy=s(n,2
)
9040 LET d$(sy,sx)="*"
9050 LET c=INT (RND*6)+2
9060 PRINT BRIGHT 1: INK c: AT sy
-1,sx-1:"*";CHR# 8: BEEP .4,1:
PRINT ;" "
9070 PRINT INK 0: PAPER 5: AT 21,
9080 "SCORE="; SCORE: AT 21,12: "SPIDE
9090 "NB="; nb: "LIVES="; LIVES
9100 RETURN
9110 PRINT PAPER 7: BRIGHT 1: FL
9120 ASH 1: INK 2: AT 10,5: "arrgh!! T
9130 HEY GOT YOU.": LET lives=lives-1
9140 FOR t=1 TO n: IF s(t,1)=x+1
AND s(t,2)=y+1 THEN GO TO 530
9150 NEXT t: STOP
9160 LET d$(s(t,2),s(t,1))=" "
9170 LET s(t,1)=0: LET nb=nb-1
9180 FOR t=1 TO 10: PRINT INK 2:
9190 AT y,x:"*";CHR# 8: BEEP .25,1:
9200 PRINT "*": BEEP .25,0: NEXT t
9210 PAPER 0: INK 7: FLASH 0: CL
9220
9230 IF n=15 AND nb=0 THEN GO SU
9240
9250 PRINT INK 0: PAPER 5: AT 21,
9260 "SCORE="; SCORE: AT 21,12: "SPIDE
9270 "NB="; nb: "LIVES="; LIVES
9280 LET rk=rk+(rk<7)
9290 IF lives THEN RETURN
9300 IF score<q(5) THEN GO TO 64

```

```

600 INPUT PAPER 5: INK 1: AT 0,0
610 ("YOU have made the "; FLASH 1:
620 TOP FIVE"; FLASH 0: PAPER 6: "Pl
630 ease enter your name. "); LINE P
640
650 FOR r=5 TO 1 STEP -1: BEEP
660 .1,r: IF score>q(r) THEN NEXT r
670 FOR g=5 TO r+2 STEP -1: BEE
680 .1,g: LET q(g)=q(g-1): LET w$(
690 )=w$(g-1): NEXT g
700 LET q(g)=score: LET w$(g)=p
710
720 BORDER 3: PAPER 5: INK 2: C
730
740 PRINT AT 3,2:"Today's Champ
750 s are:"
760 FOR r=1 TO 5: BEEP .5,r
770 PRINT AT 7+r,4: FLASH (w$(r
780 )=p# AND score=q(r));w$(r);q(r):
790 NEXT r
800 PRINT AT 18,3: FLASH 1: INK
810 5:"DO YOU WANT TO PLAY AGAIN?":
820 PAUSE 0
830 IF INKEY$="n" THEN STOP
840 GO TO 10
850 REM promote rank
860 BORDER 2: PAPER 6: INK 1: C
870
880 PRINT AT 3,7:"CONGRATULATIO
890 NS" TAB 1:"You have killed ";sc
900 ore/10;" tarantulas"
910 PRINT TAB 5:"You have now
920 been"
930 LET rk=rk-1
940 PRINT TAB 3:"promoted to t
950 he ";rk;"th" AND rk>3;"rd" AND r
960 k=3;"nd" AND rk=2;"st" AND rk=1:
970 "RANK"
980 RESTORE 8660: FOR m=1 TO 25
990 : READ n: BEEP .2,n: NEXT m
1000 PRINT TAB 2:"You will now
1010 change tanks to"
1020 PRINT TAB 5:"signify your n
1030 ew rank"
1040 PRINT "You now control the
1050 "
1060 FOR c=6 TO rk STEP -1: READ
1070 c#: NEXT c
1080 PRINT BRIGHT 1: PAPER rk: I
1090 NK 9;c#:
1100 PRINT " tank."

```

```

0330 DIM d$(22,32): LET s=7: LET
r=s/3: LET n=0: LET nb=0
0340 GO SUB 9250: PRINT AT 20,10
;"Press a key": PAUSE 500
0350 BORDER 5: PAPER 0: INK rk:
CLS
0360 LET p=0: RETURN
10000 LET UD=(T$="▲")-(T$="▼")
10010 LET HD=(T$="■")-(T$="□")
10020 LET U=Y+1: LET H=X+1: LET w
wth=1
10025 PAPER 6: INK 0:
10030 GO TO ABS vd*110+1040
10040 LET dt=(s#hd)
10050 LET d=h+dt: LET d=d OR (d<1
): LET d=d-(d>32)*(d-32)
10070 FOR t=b+bd TO d STEP bd
10080 LET ul=v-wth OR v-wth<1
10090 LET ll=(v+wth)-(v+wth>21)*(
v+wth-21)
1100 PRINT AT ul-1,t-1;d$(ul,t):
CHR$ 8: IF ul>1 THEN PRINT OVER
1;"▲" AND hd=1;"▼" AND hd=-1
1110 FOR w=ul+1 TO ll-1: PRINT F
LASH (d$(w,t)<>" ");AT w-1,t-1;d
$(w,t): NEXT w
1120 PRINT AT ll-1,t-1;d$(ll,t):
CHR$ 8: IF ll<21 THEN PRINT QUE
R 1;"▲" AND hd=1;"▼" AND hd=-1
1130 LET wth=wth+1: NEXT t
1140 GO TO 1030
1150 LET dt=(s#vd)
1160 LET d=v+dt: LET d=d OR (d<1
): LET d=d-(d>21)*(d-21)
1170 FOR t=v+vd TO d STEP vd
1180 LET ll=(h-wth) OR (h-wth<1)
1190 LET ul=(h+wth)-((h+wth-32)*
(h+wth>32))
1200 PRINT AT t-1,ll-1;d$(t,ll T
O ul):
1202 IF ll>1 THEN PRINT OVER 1:A
T t-1,ll-1;"▲" AND vd=1;"▼" AND
vd=-1
1205 IF ul<32 THEN PRINT OVER 1:
AT t-1,ul-1;"▲" AND vd=1;"▼" AND
vd=-1
1210 LET wth=wth+1
1220 NEXT t
1230 BEEP .2,5: PAPER 0: INK 7:
CLS : PRINT AT y,x:t$:

```

```

1240 PRINT INK 0; PAPER 5;AT 21,
20;"SCORE=";score;AT 21,12;"SPIDE
RS=";nb;" LIVES=";lives
1250 LET s=s-RND/3: RETURN
12600 LET UD=(T$="▲")-(T$="▼"): L
ET HD=(T$="■")-(T$="□")
12610 LET U=Y+1: LET H=X+1
12620 FOR T=1 TO R
12625 LET U=U+UD: LET H=H+HD
12630 LET V=U+(U<1)-(U>21): LET H
=H+(H<1)-(H>32)
12635 BEEP .05,10
12640 PRINT AT U-1,H-1;" "
12650 GO TO (D$(U,H)<>" ")#20+206
0
12660 PRINT INK 0;AT U-1,H-1;"■":
NEXT T
12670 LET r=r-(r>2)/10: RETURN
12680 FOR T=0 TO 7: BEEP .1,t: PR
INT INK t;AT U-1,H-1;"*";CHR$ 8:
: BEEP .1,t+7: PRINT INK 7-t;"*"
: CHR$ 8: BEEP .1,t-7: NEXT T
12690 LET SCORE=SCORE+10: LET S=S
+RND/4: LET r=INT (s/1.5)+1
12100 FOR t=1 TO n: IF v=s(t,2) A
ND b=s(t,1) THEN GO TO 2120
12110 NEXT T: STOP
12120 LET s(t,1)=0: LET nb=nb-1
12130 LET d$(v,h)=" "
12140 PRINT INK 0; PAPER 5;AT 21,
20;"SCORE=";SCORE;AT 21,12;"SPIDE
RS=";nb;" LIVES=";LIVES
12150 LET lev=lev+.01*(lev<.2)
12160 LET p=p+1: IF p=15 THEN GO
TO 700
12170 IF n=15 AND nb=0 THEN PRINT
BRIGHT 1; PAPER 4; INK 0;AT 15,
0;"Please wait 2 seconds for the
next wave.": GO SUB 9250
12190 RETURN
50000 LET T$="▲"
50010 LET X=X-(X>0)
50020 RETURN
50000 LET T$="▼"
50010 LET Y=Y+(Y<20)
50020 RETURN
70000 LET T$="■"
70010 LET Y=Y-(Y>0)
70020 RETURN
90000 LET T$="□"
90010 LET X=X+(X<31)
90020 RETURN

```



```

0000 REM DEFINE CHARACTERS
0010 DATA 66,90,126,60,126,153,6
0020 DATA 129,155,126,60,126,153
0030 DATA 0,126,60,55,55,60,126,
0040 DATA 16,16,146,254,206,254,
0050 DATA 130,254,254,230,254,14
0060 DATA 0,126,60,236,236,60,12
0070 DATA 1,3,7,15,31,63,127,255
0080 DATA 129,192,224,240,248,25
0090 DATA 255,127,63,31,15,7,3,1
0100 DATA 255,254,252,248,240,22
0110 DATA 0,0,0,24,24,0,0,0
0120 RESTORE
0130 FOR d=97 TO 107
0140 FOR e=0 TO 7
0150 READ n: POKE USR CHR$ d+e,n
0160 NEXT e: NEXT d
0170 RETURN
0180 DATA 0,-1,-2,-1,1,1,1,0,0,0
0190 DATA 1,1,1,1,0,-2,0,1,1,1,0,-1,1,0,-
0200 DATA "YELLOW", "CYAN", "GREEN",
0210 "MAGENTA", "RED", "BLUE", "BLACK"
0220 REM HEADINGS
0230 RANDOMIZE AND*65535
0240 LET H#="          "
0250 INK 1: PAPER 5: BORDER 1: C
0260 PRINT H$(2 TO 31): AT 0,0;
0270 FOR X=1 TO 11: PRINT OVER 1
0280 TAB 31; "X": NEXT X
0290 PRINT AT 21,0; H$(  TO 32)
0300 PRINT AT 10,10; BRIGHT 1; F
0310 "LASH 1; "TARANTULA"
0320 LET M#="You are in a dark p
0330 it about the size of a football
0340 field fighting for your life aga
0350 inst the dreaded tarantulas. The
0360 tarantulas slide down the side o
0370 f the pit in search of food. You
0380 must fight off these ravenous be
0390 ats if you value your life."

```

```

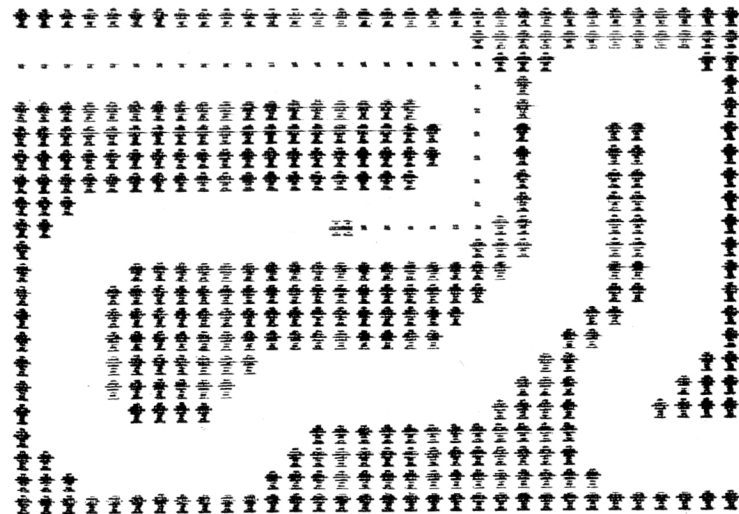
0000 LET M#="You are equipped
0010 with only a torch, a laser and y
0020 ou move around in a lightly arm
0030 ed tank which is no match for
0040 the crushing jaws of the tarantu
0050 les."
0060 LET M#="Your weapons ru
0070 n on tarantula juice and if they
0080 don't get enough they begin to
0090 wear down and their range slowly
0100 decreases."
0110 LET M#="
0120 "M#"
0130 PRINT INK 0; PAPER 6; AT 19,
0140 1; "PRESS 'e' TO END INSTRUCTIONS
0150 ; AT 20,1; "OR PRESS ANY KEY TO
0160 SPEED UP."
0170 PAUSE 500
0180 FOR L=1 TO LEN M#-27: PRINT
0190 AT 15,2; M$(L TO L+27)
0200 IF INKEY#="e" OR INKEY#="E"
0210 THEN GO TO 0160
0220 PAUSE 10: NEXT L
0230 BORDER 4: PAPER 5: BRIGHT 0
0240 CLS
0250 PRINT AT 5,2; "USE KEYS 5 TO
0260 8 TO MOVE YOUR TANK." "USE
0270 KEY '1' TO LIGHT YOUR TORCH" "U
0280 SE KEY '2' TO FIRE."
0290 DIM d$(21,32): DIM s(15,2)
0300 LET s=7: LET r=s/1.5+1
0310 LET x=INT (RND*32): LET y=I
0320 NT (RND*21)
0330 LET score=0: LET lives=3: L
0340 ET nb=0
0350 LET rk=7: LET t$="E"
0360 LET CJ=N
0370 LET lev=.05: LET p=0
0380 GO SUB 9250: INPUT " Press
0390 ENTER when you are ready": r$: RE
0400 TURN
0410 FOR z=1 TO 15
0420 LET sx=INT (RND*32)+1: LET
0430 sy=INT (RND*21)+1: LET o=RND
0440 IF o<.5 THEN LET s(z,1)=sx
0450 LET s(z,2)=20*(RND>.5)+1
0460 IF o>.5 THEN LET s(z,2)=sy
0470 LET s(z,1)=31*(RND>.5)+1
0480 NEXT z: LET n=nb
0490 RETURN

```

# Jungle Job

This is a two-stage challenge from Andrew Sweetland and Martin Jones. In the first stage, you have to guide your canoe down a fast-flowing river. There are a number of rocks blocking your way, and you must avoid them. If you hit a rock, a man-eating fish (or a faint computer version of one) will leap out of the water to consume you.

In the second stage, you'll find yourself on a dirt roadway like this:



Your car has no brakes, so working your way around the track will call on all your driving skill.

Use the following keys to control your boat and your car:

- Q - to move up
- A - to move down
- O - to move right
- P - to move left

```

1 REM ***JUNGLE JOB***
©Martin Jones & Andrew Sweetland
2 REM EPISODE I
3 GO SUB 9001: LET li=3
4 POKE 23658,0
5 FOR t=1 TO 3: BORDER 0: BRI
GHT 1: PAPER 5: INK 7: CLS: BRI
GHT 0
10 FOR f=14 TO 21: PRINT AT f,
0: PAPER 1: NEXT f
15 PRINT #1; AT 0,0; PAPER 1,,,
20 PRINT AT 8,0; BRIGHT 1; INK
3;
30 FOR f=10 TO 13: PRINT BRIGHT
T 1; AT f,0; PAPER 3: NEXT f
40 PRINT AT 1,29; BRIGHT 1; IN
K 6; "DE"; AT 2,29; "FG"
45 FOR f=15 TO 18: LET p=5+(AM
D*15): PRINT AT f,p; PAPER 1; IN
K 6; "HI"; AT f,p+10; "HI": NEXT f:
PRINT AT 20,AND*30; INK 6; PAPE
R 1; "HI"
50 LET x=17: LET y=0
55 IF y<=29 THEN IF ATTR (x,y)
=14 OR ATTR (x,y+1)=14 OR ATTR (
x,y+2)=14 THEN GO SUB 130
60 PRINT AT x,y; INK 7; PAPER
1; ".AB"
70 LET y=y+1
80 IF INKEY$="q" THEN IF (x>15
AND y<29) THEN IF ATTR (x-1,y)<
>14 AND ATTR (x-1,y+1)<>14 AND A
TTR (x-1,y+2)<>14 THEN PRINT AT
x,y; PAPER 1; INK 7; " ": LET x
=x-1

```



```

83 IF INKEY#="a" THEN IF (x<20
AND y<29) THEN IF ATTR (x+1,y) <
>14 THEN IF ATTR (x+1,y+1) <>14 T
HEN IF ATTR (x+1,y+2) <>14 THEN P
RINT AT x,y; PAPER 1; INK 7;"
": LET x=x+1
85 IF INKEY#="o" AND y<28 THEN
LET y=y-.5
86 OUT 254,0: OUT 254,16
90 FOR z=1 TO 10: NEXT z: IF y
=30 THEN NEXT t: GO TO 105
95 GO TO 55
100 REM END OF EPISODE I
105 PRINT AT x,y; PAPER 1; INK
7;"AB": FOR f=50 TO 60: FOR g=0
TO 32: OUT 254,g+16: OUT 254,0:
NEXT g: NEXT f
110 GO TO 500
120 REM OOOOPS!
130 PRINT AT x,y; PAPER 1; INK
7;"JK": FOR f=0 TO 50: OUT 254,1
6: OUT 254,0: NEXT f: PRINT AT x
+1,y+1; PAPER 1; INK 4;"CC";AT x
-1,y+1; INK 7;"LM": LET li=li-1:
IF li=0 THEN GO TO 8000
140 FOR r=1 TO 500: NEXT r: GO
TO 5
500 REM EPISODE II
505 CLS : PAPER 1: INK 4: CLS :
FOR f=0 TO 21: PRINT "NNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN": NEXT f
510 FOR f=-50 TO 50: BEEP .01,f
: NEXT f
515 FOR f=1 TO 3: PRINT PAPER 4
; INK 1;AT f,0;"
": NEXT f
520 FOR f=5 TO 6: PRINT AT f,19
; PAPER 4; INK 1;" ";AT f,23;"
": AT f,28;" "; NEXT f
525 FOR f=7 TO 12: PRINT AT f,2
3; PAPER 4; INK 1;" ";AT f,28;"
": NEXT f
530 FOR f=3 TO 4: PRINT AT f,23
; PAPER 4; INK 1;" "; NEX
T f
535 FOR f=8 TO 10: PRINT AT f,3
; PAPER 4; INK 1;"
": NEXT f
540 FOR f=10 TO 18: PRINT AT f,
1; PAPER 4; INK 1;" "; NEXT f

```

```

545 FOR f=15 TO 17: PRINT AT f,
11; PAPER 4; INK 1;"
": AT f,25;" "; NEXT f
550 FOR f=18 TO 20: PRINT AT f,
3; PAPER 4; INK 1;" ";AT
f,26;" "; NEXT f
555 FOR f=13 TO 15: PRINT AT f,
20; PAPER 4; INK 1;" ";AT f,27
;" "; NEXT f
560 PRINT PAPER 4; INK 1;AT 2,2
0;" ";AT 2,24;" ";AT 3,20;"
";AT 4,18;" ";AT 7,18;"
";AT 8,20;" ";AT 9,2;" ";AT 9,8
";AT 11,22;" ";AT 12,21;" ";AT
13,23;" ";AT 13,30;" ";
565 PRINT PAPER 4; INK 1;AT 14,
19;" ";AT 14,23;" ";AT 14,26;"
";AT 16,10;" ";AT 16,21;" ";A
T 16,28;" ";AT 17,4;" ";AT 17,9;"
";AT 18,11;" ";AT 18,25;" ";
AT 19,2;" ";AT 19,11;" ";AT 19,2
5;"
570 LET x#="": LET x=2: LET y=0
: LET d#="0"
580 IF x=(18 AND y=31) OR (x=19
AND y=31) OR (x=20 AND y=31) TH
EN GO TO 7000
581 IF ATTR (x,y)=12 THEN GO TO
800
583 PRINT AT x,y; PAPER 4; INK
0;d#
585 OUT 254,0: OUT 254,16
590 IF INKEY#="o" THEN LET x#="
0": LET d#="0"
600 IF INKEY#="a" THEN LET x#="
a": LET d#="P"
610 IF INKEY#="q" THEN LET x#="
q": LET d#="R"
620 IF INKEY#="p" THEN LET x#="
p": LET d#="0"
630 PRINT AT x,y; PAPER 4; INK
0;"T"
640 IF x#="o" THEN LET y=y-1
650 IF x#="p" THEN LET y=y+1
660 IF x#="a" THEN LET x=x+1
670 IF x#="q" THEN LET x=x-1
680 GO TO 580
700 GO TO 900
800 REM OOOOPS!

```

```

810 PRINT AT X,Y; INK 2; FLASH
1;"S"; FLASH 0; AT X-1,Y; PAPER 4
; INK 0;"LM": FOR f=0 TO 50: OUT
254,15: OUT 254,0: NEXT f: LET
li=li-1: IF li>0 THEN FOR f=0 TO
500: NEXT f: GO TO 8000
8020 IF li<=0 THEN GO TO 8000
7000 REM SUCCESS!!
7010 BORDER 0: PAPER 0: INK 7: CLS
LS: PRINT AT 10,6;"CONGRATULATI
ONS!"; AT 12,0;"YOU HAVE BEATEN Y
OUR WAY THROUGH"; AT 13,10;"THE J
ANGLE!!"
7020 FOR f=0 TO 30: BEEP .01,f:
BEEP .01,-f: NEXT f
7030 GO TO 8050
7040 STOP
8000 REM FAILURE!!
8010 RESTORE : BORDER 0: PAPER 0
: INK 7: CLS
8020 PRINT AT 10,1;"HA! HA! FASC
IST! YOU'VE USED UP EVERY CHANC
E THAT I GAVE YOU!"
8030 FOR f=1 TO 12: READ a,b: BE
EP a,b: NEXT f
8040 DATA .5,-.5,.5,-.5,.5,0,.5,0,
.5,2,.5,2,.5,5,.5,10,.5,15,.5,20
,.5,25,2,-20
8050 PRINT AT 15,0;"WOULD YOU LI
KE ANOTHER GO? (Y/N)"
8060 IF INKEY#="Y" THEN LET li=3
: GO TO 5
8070 IF INKEY#="N" THEN PRINT US
R 0
8080 GO TO 8060
8090 STOP
8095 STOP
9000 REM UDG DATA
9001 RESTORE 9000: FOR f=0 TO 15
9: READ a: POKE USR "a"+f,a: NEX
T f
9010 DATA 0,0,1,1,48,96,127,63
9020 DATA 0,0,192,192,134,131,25
5,254
9030 DATA 240,56,188,252,124,63,
0,4
9040 DATA 0,0,0,3,15,15,31,31
9050 DATA 0,0,0,192,240,240,248,
240
9060 DATA 31,31,15,15,3,0,0,0

```

```

9070 DATA 248,248,240,240,192,0,
0,0
9080 DATA 0,0,0,1,3,7,31,63
9090 DATA 0,0,0,192,192,224,240,
240
9100 DATA 0,24,48,56,20,14,7,2
9110 DATA 0,12,6,14,20,50,110,64
9120 DATA 0,204,170,200,0,0,0,0
9130 DATA 0,204,140,204,0,0,0,0
9140 DATA 24,60,126,60,24,24,60,
0
9150 DATA 0,236,68,255,255,68,23
0,0
9160 DATA 90,126,90,24,90,126,90
,24
9170 DATA 0,110,34,255,255,34,11
0,0
9180 DATA 24,90,126,90,24,90,126
,90
9190 DATA 0,6,70,0,0,48,248,254
9200 DATA 0,0,0,24,24,0,0,0
9210 RETURN

```

# Dodge

In this fast-moving game of logic and skill, you are heavily disguised as an asterisk. You have to avoid the VBB's (Vicious Black Blobs) which appear at random around you to terrorise and trap you.

As you'll see when you run DODGE, you must enter a 'level of difficulty'. You need to enter a number between 1 and 100, with the higher numbers representing easier games. Level 100 is only for real experts.

You'll have to keep moving to ensure you are not trapped by the VBB's. Use the arrow keys to move around.

```
20 INPUT "Use cursor keys to m
ove,          Input Difficulty
              1(tricky)-100(easy)";
";P
30 BORDER 4: LET x=10: LET y=1
5: LET s=0: LET a=53
40 PRINT " ": POKE 23692,255
50 IF SCREEN$(x,y)="_" THEN G
O TO 160
60 PRINT AT x,y;"*": LET s=s+1
70 PRINT AT x+INT (RND*3)-1,y+
INT (RND*3)-1: INVERSE 1;"_"
80 PAUSE p
90 IF CODE INKEY$/53 OR CODE I
NKEY$>56 THEN GO TO 40
100 LET a=CODE INKEY$
110 IF a=53 AND y>0 THEN LET y=
y-1
120 IF a=54 AND y<20 THEN LET x
=x+1
130 IF a=55 AND x>0 THEN LET x=
x-1
140 IF a=56 AND y<20 THEN LET y
=y+1
```

```
150 GO TO 40
160 PRINT " OUCH! YOU SCORED "
;S: PAUSE 500: INPUT "ENTER TO P
LAY AGAIN ";S$: GOTO RUN
```

# Duel Cabbageway

The fiendish mind of Neal Cavalier-Smith thought up this strange little program (and its 'witty' title). The aim of this game is to plant cabbages. You score one point for each cabbage you plant, and lose 10 if you attempt to plant a cabbage on top of a house. Three points will be deducted from your score if you plant them next to a house or under a tree. (I told you it was strange.)

```
3 LET a=0
4 PAPER 0: INK 7: BORDER 0
5 PRINT "DUEL CABBAGEWAY"
```

## DUEL -CABBAGEWAY

WAY"

```
6 PRINT "You score 1 point for each cabbage you plant, but lose 10 if you plant one on top of a post (it will not grow there) and 3 cabbages if you plant them by a house (beware of terraces) or under a tree."
```

```
10 LET a$="": LET s=0: LET h=0
```

```
20 DIM b$(12,32)
```

```
30 LET c$=""
```

```
50 FOR x=1 TO 12
```

```
60 LET c$=c$+" "
```

```
70 LET b$(x)=c$+a$
```

```
80 NEXT x
```

```
90 LET y=13
```

```
100 LET x=(RND*12+1)
```

```
101 PRINT AT 20,12: FLASH 1;"START"
```

```
110 PRINT AT 21,0: PRINT INK 5;
```

```
b$(x)
```

```
120 LET x=x+(RND*2-1)
```

```
125 LET s=s+1
```

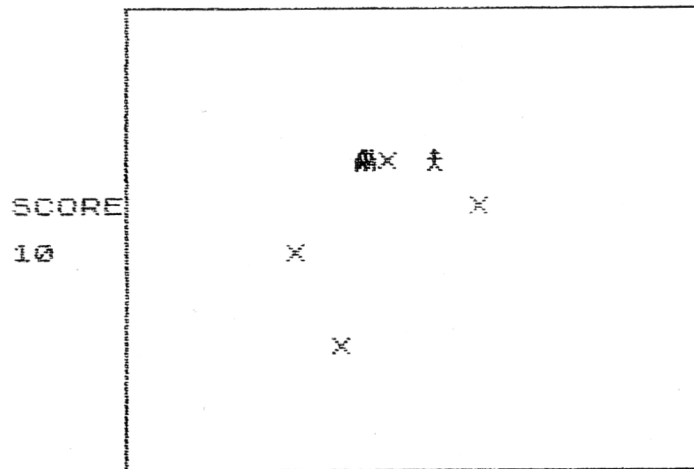
```
130 IF x>12 THEN LET x=x-1
140 IF x<1 THEN LET x=x+1
145 IF INT x=6 AND y=12 OR y=10
THEN PRINT AT 20,4: INK 2;" "
: PRINT AT 21,4: INK 7;" " : LE
T s=s-2
146 IF INT x=9 AND y=14 THEN PR
INT AT 20,20: INK 4;" " : PRINT
AT 21,20: INK 4;" " : LET s=s-3
150 IF INKEY$="5" THEN LET y=y-
1
160 IF INKEY$="6" THEN LET y=y+
1
170 IF SCREEN$(11,y)<>" " THEN
GO TO 220
180 PRINT AT 11,y: INK 4;"B"
190 POKE 23692,255
200 GO TO 110
220 BEEP 1,1: LET h=h+1: LET s=
s-10: IF h=3 THEN PRINT "START="
: s: FOR x=1 TO 10: LET z=TAN x:
NEXT x: PAUSE 50: RUN
230 GO TO 110
1450 IF INT x=6 AND y=12 OR y=10
THEN PRINT AT 20,4: INK 2;" "
: PRINT AT 21,4: INK 7;" " : LE
T s=s-2
```

# Ghost Chase

The ghost is after you. The small playing grid has a number of mines (X) on it, and you have to try and tempt the ghost to come into contact with one of the mines. At this point, the ghost will be electrocuted.

The mines also pose a danger to you, so as you try to evade the evil little ghost, you have to make sure you don't touch mines or the ghost.

Written by Neal Cavalier-Smith and Graham White, this program demonstrates a fairly intelligent ghost in action. You certainly



won't be able to evade it for long. You'll see that the program runs very

quickly indeed. If it is too fast for you the first few times you run it, insert a few dummy FOR/NEXT loops to slow it down a bit.

```

1 GO SUB 800
2 GO SUB 800
3 LET q=0: LET s=0: LET p=6:
LET h=0
5 BORDER 0: PAPER 0: INK 7: C
L5
10 LET a=INT (RAND*10)+2
15 PLOT 0,0: DRAW 200,0: DRAW
0,200: DRAW -200,0: DRAW 0,-100
200 LET b=INT (RAND*12)+10
300 LET x=2*(INT (RAND*8))+2
40 LET y=2*(INT (RAND*10))+10
40 LET e=2*(INT (RAND*6))+6: LE
T 7=2*(INT (RAND*8))+10
44 IF e=x OR e=y OR f=y OR f=b
THEN GO TO 45
45 PRINT AT e,f: PAPER 2: "X"
46 IF q=p THEN GO TO 49
47 LET q=p: FOR z=0 TO 7-p: GO
TO 48: NEXT z
48 IF c<20 THEN LET c=20
50 PRINT AT x,y: INK 5: CHR# 15
6: LET v=x: LET w=y
60 PRINT CHR#(150) " is a user
400 PRINT AT a,b: INK 6: CHR# 14
4: LET c=a: LET d=b
60 PRINT CHR#(144) " is a user
400 PRINT CHR#(150) " is a user
70 LET a=a-(x<a)+(x>a)
80 LET b=b-(y<b)+(y>b)
900 LET x=x+0*(INKEY#="0")-2*(I
NKEY#="7")
900 LET x=x+0*(x<1)-2*(x>20)
1000 LET c=c+0*(INKEY#="0")-2*(I
NKEY#="5")
1000 LET w=w+0*(w<5)-2*(w>20)
110 PRINT AT v,w: " " AT c,d: " "
120 IF a=x AND b=y THEN GO TO 4
60
130 IF SCREEN# (a,b)="X" THEN G
O TO 200
130 IF SCREEN# (x,y)="X" THEN G
O TO 300

```

```

1440 GO TO 49
1450 PRINT AT a,b: FLASH 1;CHR#
1460 REM CHR#(145) is a user
1470 FOR d=0 TO 49 STEP 4
1480 READ a:POKE USA "b"+d,d
1490 NEXT d
1500 DATA 145,62,0,195,0,74,137,
1510 FOR d=0 TO 7
1520 READ a:POKE USA "m"+d,d
1530 NEXT d
1540 DATA 8,26,6,62,6,6,20,34
1550 RETURN
1560 BORDER 0: PAPER 0: INK 7: C
1570 PRINT TAB 5; INK 6; FLASH 1
1580 H O S T " C H A S E "
1590 PRINT "You have to move y
1600 our bomb ("; INK 6;CHR# 155; INK
1610 7;") around the screen to enti
1620 ce the ghost ("; INK 6;CHR# 144;
1630 INK 7;") which is chasing you,
1640 to run into the bombs ("; PAPER
1650 0;"X"; PAPER 0;"). But be carefu
1660 l you don't run into the bomb or
1670 let the ghost catch you. You ha
1680 ve 3 lives.
1690 Look...:
1700 PRINT "PRESS KEYS ""S""-""
1710 TO MOVE IN DIRECTION SHOWN
1720 ON KEY"
1730 PRINT "FLASH 1;Press any
1740 key to continue"
1750 LET s$=INKEY$: IF INKEY$=""
1760 THEN GO TO 840
1770 RETURN

```

```

1440 GO TO 49
1450 PRINT AT a,b: FLASH 1;CHR#
1460 REM CHR#(145) is a user
1470 FOR d=0 TO 49 STEP 4
1480 READ a:POKE USA "b"+d,d
1490 NEXT d
1500 DATA 145,62,0,195,0,74,137,
1510 FOR d=0 TO 7
1520 READ a:POKE USA "m"+d,d
1530 NEXT d
1540 DATA 8,26,6,62,6,6,20,34
1550 RETURN
1560 BORDER 0: PAPER 0: INK 7: C
1570 PRINT TAB 5; INK 6; FLASH 1
1580 H O S T " C H A S E "
1590 PRINT "You have to move y
1600 our bomb ("; INK 6;CHR# 155; INK
1610 7;") around the screen to enti
1620 ce the ghost ("; INK 6;CHR# 144;
1630 INK 7;") which is chasing you,
1640 to run into the bombs ("; PAPER
1650 0;"X"; PAPER 0;"). But be carefu
1660 l you don't run into the bomb or
1670 let the ghost catch you. You ha
1680 ve 3 lives.
1690 Look...:
1700 PRINT "PRESS KEYS ""S""-""
1710 TO MOVE IN DIRECTION SHOWN
1720 ON KEY"
1730 PRINT "FLASH 1;Press any
1740 key to continue"
1750 LET s$=INKEY$: IF INKEY$=""
1760 THEN GO TO 840
1770 RETURN

```

## Frog on a Log

Frog on a Log is another great program from Malcolm Young. He says he enjoyed creating this program as with it he had broken new ground in programming techniques. "For example," he wrote, "I have been experimenting with machine code routines and produced a scroll left routine of my own...The computer cannot usually recognize user-defined graphics using SCREEN\$, and most people have got around this by using the ATTR function.

"I found that by pointing the system variables CHARS to the UDG memory, SCREEN\$ can operate in the normal way, although you must use the letters that the character represents. I have used this technique in my program from lines 300 to 490. This technique has the advantage of recognising any UDG characters, including non-flashing, non-bright black ones.

"I have made the program so it can be typed into the 16K model as well as the 48K model without any modifications, i.e. line 30 CLEAR USR "2"-43 for the machine code."

The object of Malcolm's program is to catch as many insects as possible using keys I and P for left and right, and CAPS SHIFT to make the frog jump. You must avoid going off the screen or falling off

the log and encountering the spider, as this will result in a loss of life. Each fly starts off with 200 energy units and this slowly decreases to zero. When it reaches zero, the frog starves and dies.

Butterflies may appear, and if caught are worth 10 energy points plus 50 bonus points. You have only a 50% chance of catching a butterfly, no matter how accurate you are in your jump.

You'll find this program quite addictive and you may wish to modify it to include a skill level option, which changes the size and number of the logs. You should type the program in and then save it by using SAVE "frog" LINE 5000.

```
1 REM a= b=* c=^ d=^ e=^ f=^
      gb=^ ij=^ lm=^
10 REM frog on a log
20 DEF FN r(x)=INT (RAND*x)+1
30 CLEAR USR "a"-43
40 GO SUB 9000: REM vdg
45 GO SUB 8110
50 GO SUB 8000: REM display
55 GO TO 500
70 POKE 23506,0: POKE 23507,60
80 PRINT AT 0,0; BRIGHT 1; INK
  7; PAPER 2; "SCORE: ";50;TAB 11;
  "ENERGY: ";200;TAB 24; "FROGS: ";0
  109
90 RETURN
100 REM read keyboard
110 LET k=PEEK 23556
120 LET x=x+(k=80)-(k=73)
130 IF x<1 OR x>31 THEN GO TO 3
30
140 LET k=IN 65276
145 GO TO (k=254 OR jp)*ds+150
150 PRINT AT y,x-1; " "
160 IF NOT POINT (x#8,31) THEN
GO TO 450
```

```

165 RETURN
170 REM JUMP
180 LET JP=1: LET EP=10: LET SC
=40: PRINT AT 4,X-1;" "
190 POKE 23605,10: POKE 23607,5
200 FOR I=0 TO 1
210 LET Y=Y-JP
215 GO TO (Y=17)+100+220
220 IF SCREEN$(Y,X)="" THEN G
O TO 220
230 PRINT AT Y-1,X;"a" AND JP=0
240 PRINT AT Y+1,X;"a" AT Y+2,
X;"a" AND (JP=0 AND Y<15): BEEP
250,5
260 NEXT I
270 LET EP=EP+JP
275 LET JP=JP-2+(Y<15)+(Y>15)
280 POKE 23605,0: POKE 23607,50
290 RETURN
300 LET A$=SCREEN$(Y,X)
310 IF A$="i" OR A$="j" THEN LE
T E=(Y-1)*32+X-2: LET I$(E TO E+
2)=" ": GO TO 350
320 IF A$="g" OR A$="h" THEN LE
T B=0: LET ED=ED+10: LET SC=SC+2
330 GO TO 350
335 FOR Z=20 TO 30: BEEP .05,-Z
: BEEP .1,-Z-.5: NEXT Z
340 LET F109=F109-1: LET Z=0: B
EEP 1,-10: IF NOT F109 THEN GO T
O 1200
345 LET ED=200: GO TO 6000
350 LET SC=SC+10: LET JP=-1
360 BEEP .5,10
370 GO SUB 70
380 RETURN
390 IF SCREEN$(12,X)="" THEN
GO TO 450
400 GO SUB 6000
410 PRINT AT Y-1,X;"c";AT Y,X;"
"
420 PRINT AT 16,X-1;"aaa";AT 17
,X;"b"
430 RETURN
440 BEEP .05,-5
450 PRINT AT 16,X;" ";AT 17,X;"
";AT 18,X;" "
460 BEEP .05,-5
470 PRINT AT 17,X;" ";AT 18,X;"
";BEEP .1,-5: PRINT AT 18,X;"
"

```

```

490 GO TO 330
500 REM main routine
510 IF AND(.05 THEN GO TO 560
520 LET ST=32*(FN R(15)+9)-1
530 LET I$(ST TO ST+1)="* "
540 PRINT AT 1,0: INK 1;I$
550 LET I$=I$(3 TO 7)
560 LET X=X-NOT JP: LET L=USR #
570 BEEP .005,10
580 GO SUB 100
590 GO TO (INDT B)+40+630
600 LET D=FN R(4): LET B=B-1
610 LET BY=BY+(D=1)-(D=2): LET
BX=BX+(D=3)-(D=4)
620 LET BY=BY-(BY>15)+(BY<1): L
ET BX=BX+(BX<0)-(BX>30)
630 PRINT AT BY,BX: INK 2: OVER
1;" "
640 IF NOT B THEN LET B=(AND>.9
)+(FN R(20)+5): IF B THEN LET BY
=FN R(15)+1: LET BX=FN R(30)
650 REM SP3207
660 LET S=S+(AND>.4)*((S<X)-(S>
X))
670 PRINT INK 0;AT 16,0;Q$;AT 1
6,S;" "
680 IF Y>15 AND (X=S OR X=S+1)
THEN GO TO 810
690 LET EN=EN-1
700 PRINT AT 0,10: BRIGHT 1: IN
K 7: PAPER 2;EN;" "
710 IF EN<0 THEN PRINT PAPER 2;
AT 8,7: FLASH 1;"No More Energy!
": GO SUB 330
720 GO SUB 100
730 GO TO 500
740 PRINT INK 0;AT 16,S;" ";AT
17,S;" "
750 FOR Z=0 TO 1 STEP .1
760 BEEP Z,-Z*10: NEXT Z
770 GO SUB 340
780 GO TO 500
1000 REM score table
1010 GO SUB 5000: GO SUB 70
1020 IF SC<=HS THEN GO TO 1050
1030 INPUT "You have the new hig
h score!";"Please type in your n
ame. ";LINE N$
1040 LET HS=SC
1050 DIM A$(1)
1060 INPUT "Do you want to play
again?";A$

```



```

1070 IF a$="n" OR a$="N" THEN PO
KE 23693,56: PRINT AT 20,10;"OK,
Play again soon.": PAUSE 50: BOR
DER 7: RANDOMIZE USR 4750
1080 GO SUB 8120
1090 GO TO 50
0000 REM Title
0010 BORDER 5: PAPER 1: INK 4: C
L5: PRINT AT 2,0;
0020 PRINT TAB 4;"███";TAB 20;"█
███"
0030 PRINT TAB 4;"█";TAB 19;"█
███"
0040 PRINT TAB 4;"███ ████ ████
███"
0050 PRINT TAB 4;"███ █ ████
███"
0060 PRINT TAB 4;"███ █ ████
███"
0070 PRINT TAB 16;"█";TAB 14;"███
███"
0080 PRINT TAB 5;"███ ████
███"
0090 PRINT TAB 5;"███ ████
███"
0100 PRINT TAB 5;"███ ████
███"
0110 PRINT TAB 5;"███ ████
███"
0120 PRINT TAB 5;"███ ████
███"
0130 PRINT TAB 22;"█";TAB 20;"███
███"
0135 RESTORE
0140 FOR a=1 TO 10
0150 READ n: IF n=99 THEN PAUSE
5: GO TO 0170
0160 BEEP .15,n
0170 NEXT a
0180 LET x=PEEK 23730+256*PEEK 2
3731
0190 IF x=USR "a"-1 THEN RUN
0200 RETURN
05000 DATA 2,5,3,99,2,5,3,99,5,10
00000 REM set display
0010 BORDER 2: PAPER 5: INK 6: C
L5
0020 GO SUB 70
0040 PRINT PAPER 3; INK 4; AT 17,
0;TAB 31;" ";TAB 31;" "
0050 PRINT PAPER 1;TAB 31;" ";TA
B 31;" "
0053 PAPER 8

```

```

0055 PRINT AT 18,0;"██████████████
██████████████"
0057 PRINT AT 19,0;"██████████████
██████████████"
0060 PRINT AT 21,0; PAPER 2; INK
7; BRIGHT 1;" HIGH SCORE ";hs;"
";n$
0070 LET y=17: LET x=19
0080 LET jp=0
0090 LET ds=20
0100 RETURN
0110 LET hs=0: LET n$=""
0120 DIM i$(480)
0160 LET b=0: LET w=0
0170 LET n=USR "a"-776: RANDOMIZ
E n: LET lw=PEEK 23670: LET hh=P
EEK 23671: RANDOMIZE
0180 LET b=0: LET w=0
0190 LET s=0
0200 LET frog=3
0210 LET en=2000
0220 LET sc=0
0240 DIM g$(32)
0300 RETURN
0000 REM define graphics
0010 RESTORE 9000
0020 FOR c=1 TO 10: READ c$
0030 FOR b=0 TO 7: READ n
0040 POKE USR c$+b,n
0050 NEXT b: NEXT c
0060 LET mc=USR "a"-42
0070 FOR m=mc TO mc+41
0080 READ byte: POKE m,byte
0090 NEXT m
0100 RETURN
0195 DATA "a",0,0,0,0,0,0,0,0
0200 DATA "b",36,126,60,153,126,
126,60,99: REM frog
0210 DATA "c",165,231,66,129,153
,153,126,60: REM 1/2 jumping frog
0220 DATA "d",126,126,60,90,129,
195,36,195: REM 1/2 jumping frog
0230 DATA "e",170,85,170,85,170,
85,170,85
0240 DATA "f",170,85,170,85,0,0,
0,0: REM log (bottom)
0250 DATA "g",17,109,85,130,170,
85,109,17: REM 1/2 butterfly
0260 DATA "h",16,108,212,130,170
,212,108,16: REM 1/2 butterfly
0270 DATA "i",1,2,4,73,253,56,86
,73: REM insect

```

```
9280 DATA "j",224,16,112,128,85,  
0,0,0: REM insect tail  
9300 DATA "l",0,34,83,159,186,17  
1,144,80: REM spider  
9310 DATA "m",0,68,202,242,98,21  
0,10,10: REM spider  
9320 REM machine code scroll left  
9330 DATA 33,32,80,62,3,50,129,9  
0,6,6,197,229,126,245,84,93,35  
9340 DATA 1,31,0,237,176,241,18,  
0,36,193,16,237,167,17  
9350 DATA 224,7,237,82,58,129,93  
,61,32,220,201
```

# LEISURE LINES





```

0300 FOR I=1 TO 10: PRINT AT 0,04;
0310 NEXT I
0320 PRINT AT 0,00; "Turn over";
T 0,00; " "
0340 IF INKEY#<>="" THEN GO TO 034
0350 LET A#=INKEY#: IF A#="1" OR
A#="4" THEN GO TO 0300
0360 PRINT AT 0,04; A#: BEEP .05,
0370 LET A1=VAL A#
0370 IF INKEY#<>="" THEN GO TO 037
0380 LET A#=INKEY#: IF A#="1" OR
A#="5" THEN GO TO 0300
0390 PRINT AT 0,00; A#: BEEP .05,
0400 LET A2=VAL A#
0400 IF A1(A1,A2) <> 0 THEN GO TO 4
0410 PRINT AT 10,00; INVERSE 1; "
T 0,00; " "
0420 BEEP 1, -10: PRINT AT 0,04; "
"; AT 10,00; " "
0430 GO TO 0300
0450 PRINT AT 0,00; " "
T 0,04; " "
0470 LET X1=(A1-1)*5+0: LET X2=(
A2-1)*4+2: LET X=X1: LET Y=X2: L
ET C=A1: LET A=A2: GO SUB 0500
0490 LET A1=C: LET A2=A
0500 PRINT AT 0,00; "Turn over";
T 0,00; " "
0540 IF INKEY#<>="" THEN GO TO 054
0550 LET A#=INKEY#: IF A#="1" OR
A#="4" THEN GO TO 0300
0560 PRINT AT 0,04; A#: BEEP .05,
0570 LET P1=VAL A#
0570 IF INKEY#<>="" THEN GO TO 057
0580 LET A#=INKEY#: IF A#="1" OR
A#="5" THEN GO TO 0300
0590 PRINT AT 0,00; A#: BEEP .05,
0600 LET A2=VAL A#
0600 IF A1(A1,A2) <> 0 THEN GO TO 0
0610 PRINT AT 10,00; INVERSE 1; "
T 0,00; " "

```

```

0620 BEEP 1, -10: PRINT AT 0,04; "
"; AT 10,00; " "
0630 GO TO 0300
0650 PRINT AT 0,00; " "
T 0,04; " "
0670 LET X2=(P1-1)*5+0: LET A2=(
P2-1)*4+2: LET X=X2: LET A=A2: L
ET C=A1: LET A=A2: GO SUB 0600
0690 LET A1=C: LET A2=A
0690 LET A1=C+1
0700 IF A1(A1,A2) <> 0 (P1,P2) THEN
GO TO 0600
0705 FOR I=0 TO 10 STEP 0: BEEP
.1, .1: NEXT I
0710 FOR I=0 TO 0: PRINT AT X1+1
,Y1; " "; AT X2+1,Y2; " "; NEXT
I
0720 LET B(A1,A2)=0: LET B(P1,P2
)=0
0730 LET C=0: FOR I=1 TO 4: FOR
J=1 TO 5: LET C=C+B(I,J): NEXT J
: NEXT I
0740 IF C=0 THEN GO TO 0800
0750 GO TO 0300
0760 BEEP .0,0
0765 FOR I=1 TO 200: NEXT I: FOR
I=0 TO 0: PRINT AT X1+1,Y1; INK
0; " "; AT X2+1,Y2; " "; NEXT
I
0770 GO TO 0300
0800 RESTORE 1000: FOR Z=1 TO B(
C,A): BEEP Z#: NEXT Z
0810 PRINT AT X,Y; Z#(1 TO 0); AT X
+1,Y; Z#(4 TO 0)
0820 PRINT AT X+2,Y; Z#(7 TO 0);
T X+0,Y; Z#(10 TO 10)
0850 RETURN
0860 CLS
0870 PRINT AT 0,0; "You cleared t
he table after"; AT 10,11; 00; " to
0880 FOR I=1 TO 25: BEEP .01, .1:
NEXT I
0890 LET B#="0705 0300 0600 070
0900 BEEP 1: PRINT AT 10,00; "
0910 FOR I=1 TO 100: BEEP .01, .1
X 0,0; B#(1); IF B#(1) <> " " THEN 0
920 BEEP .03,00;
0930 IF B#(1) = " " THEN BEEP 0

```

```

0055 NEXT 4
0060 IF INKEY#="" THEN GO TO 0060
0070 RUN
0080 DATA " "
0090 DATA " "
0100 DATA " "

```

## Sub Search

In this challenging game from Malcolm Young, you have to track down and destroy a submarine in the shortest possible time, by employing your fleet in a strategically sound manner.

You control a fleet of warships made up of a battle cruiser, two patrol vessels and three frigates. Your task is to hunt and destroy an enemy submarine which has been lurking around the shipping lanes near you. Your battle cruiser is equipped with a sonar system which rises in pitch as it gets closer to the sub.

Each vessel can carry only a limited quantity of depth charges. Once they've been fired, your ship must return to port. Full instructions are within the program.

```

40 REM SUB Search
000 REM © 1983 Malcolm Young
000 REM a=# b=# c=# d=# e=# f=#
000 REM g=# h=# i=# j=# k=# l=# m=# n=#
40 DEF FN b()=SQR (FN s(sx,s(1
),1))+FN s(sy,s(1,2)))
50 DEF FN s(a,b)=ABS (a-b)*ABS
(a-b)
60 DEF FN r(x)=INT (RAND*x)+1
70 DEF FN t(j,k,l,m)=(ABS (j-k
)<7)*(l=m)
80 GO SUB 9000: REM define gra
phics
90 GO SUB 8000: REM instructio
ns
100 GO SUB 7000: REM set up scr
een
110 REM vvvvvvvvvvvvvvvvvvvvvvvvvvvv
120 REM move submarine

```

```

130 LET SM=FN r(6)
140 LET SX=SX+(SM=1)-(SM=2)
150 LET SY=SY+(SM=3)-(SM=4)
160 LET SD=SD+(SM=5)-(SM=6)
170 LET SX=SX+(SX<1)-(SX>19)
180 LET SY=SY+(SY<1)-(SY>19)
190 LET SD=SD+(SD<1)-(SD>9)
200 FOR r=1 TO 6
210 IF FN t(SX,S(C,1),SY,S(C,2))
THEN GO TO 5000
220 IF FN t(SY,S(C,2),SX,S(C,1))
THEN GO TO 5000
230 NEXT c
240 REM player's move
250 PRINT #0;" Press any key to
throw dice."
260 IF S(1,1)>=0 THEN PRINT AT
18,21;"Scanner";AT 16,21;"reads:"
; INK 3; FLASH 1; INT FN b(); FLA
SH 0; INK 5;" "
270 IF S(1,1)>=0 THEN BEEP .1,1
280 IF FN b(): PAUSE 10
290 IF INKEY#="" THEN GO TO 260
300 REM dice
310 LET nb=nb+1
320 LET d=FN r(20)+10: LET M=FN
r(6)
330 FOR i=1 TO d
340 LET M=M*(M<6)+1: BEEP .1,M
350 PRINT AT 18,21;"You have ";
AT 19,21;CHR$(151+M);" moves."
360 NEXT i
370 PRINT BRIGHT 1; FLASH 1; IN
K M; PAPER 9;AT 19,21;CHR$(151+
M)
380 BEEP 1,10: LET n=0
390 LET n=n*(n<6)+1
400 LET p=s(n,2): LET r=s(n,1)
410 IF r<0 THEN GO TO 380
420 PRINT FLASH 1; INK n-1; PAP
ER 7;AT p,r;d$(n)
430 PRINT PAPER n-1; INK 9;AT 2
0,3;s$(n);"(");s(n,1);",";s(n,2);
";
440 PRINT AT 21,8;"depth charge
s left:";s(n,3)
450 INPUT AT 0,0;"Enter move. "
; LINE M#: IF M#="" THEN PRINT I
NK n-1; PAPER 5+(n<6);AT p,r;d$(
n): GO TO 380
460 LET l=LEN M#: IF l>M THEN B
EEP 1,-10: GO TO 430
470 FOR r=1 TO l

```

```

480 IF M$(c)<>"n" AND M$(c)<>"s
" AND M$(c)<>"w" AND M$(c)<>"e"
AND M$(c)<>"d" THEN INPUT AT 0,0
;"Invalid entry. Try again. "; LI
NE M#: GO TO 440
490 IF M$(c)="d" THEN GO TO 670
500 NEXT c
510 REM move ship
520 FOR q=1 TO l
530 PRINT AT p,r; PAPER 5; BRIA
HT 1; INK 0;"f"
540 BEEP .1,15
550 LET p=p+(M$(q)="s")-(M$(q)="
n")+ (p<0)-(p>20)
560 IF p=20 AND r<6 THEN LET s(
n,3)=3: BEEP 1,20: PRINT AT 21,0
;"3 Depth Charges loaded.": LET
p=19
570 LET r=r+(M$(q)="e")-(M$(q)="
n")+ (r<0)-(r>20)
580 IF ATTR(p,r)<64 THEN GO TO
590
590 PRINT AT p,r; PAPER 5; INK
n-(n<6); OVER 1;d$(n)
600 BEEP .1,20
610 LET M=M-1
620 PRINT BRIGHT 1; FLASH 1; IN
K M; PAPER 9;AT 19,21;CHR$(151+
M)
630 NEXT q
640 LET s(n,1)=r: LET s(n,2)=p:
IF NOT u THEN GO TO 650
650 INPUT "Do you want to fire
a charge?"; LINE i$
660 IF i$<>"y" AND i$<>"n" THEN
INPUT "Fire a depth charge? ";
LINE i$: GO TO 630
670 IF i$="y" THEN GO TO 670
680 IF M>0 THEN GO TO 380
690 GO TO 120
700 REM fire depth charge
710 IF S(n,3)=1 THEN PRINT FLAS
H 1;AT 21,0;"No depth charges le
ft!": GO TO 650
720 PRINT AT 2,21; PAPER 6; INK
0;"YELLOW"
730 FOR a=1 TO 7: BORDER 7: BEE
P .3,-1: BORDER 6: PAUSE 5: NEXT
a
740 PRINT FLASH 1; INK 7; PAPER
3;AT 21,3;"Depth charges ready!"

```

```

710 INPUT "Enter depth. (1 TO 1
0)";dp
720 IF dp<1 OR dp>10 THEN INPUT
"dp is out of range, try again
";dp: GO TO 720
730 LET dp=INT dp
740 BEEP .5,-10: BEEP .2,20
750 FOR r=10 TO -dp STEP -1
760 BEEP .02,r
770 NEXT r
780 BEEP .8,-10: PAUSE 20
790 IF sx=r AND sy=p THEN BEEP
.5,1: IF sd=dp THEN GO TO 1000
800 LET s(n,3)=s(n,3)-1
810 PRINT FLASH 1; AT 21,0; "Miss
ed!" : IF sx=r AND sy=p THEN PRIN
T AT 21,6; "The sub's right under
you!"
820 PAUSE 100
830 PRINT AT 21,0; "
840 GO TO 110
1000 REM sub destroyed
1010 BORDER 1: PAPER 1: BRIGHT 1
: INK 7: CLS
1020 PRINT AT 2,3; "CONGRATULATIO
NS!"
1030 PRINT AT 4,3; "You manage to
destroy the"
1040 PRINT AT 6,3; "sub in ";nb;"
rounds."
1050 PRINT "That's ";pc/nb*100;"
% efficiency."
1060 INPUT "Would you like to pl
ay again? "; LINE i$
1070 IF i$="y" THEN RUN
1080 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS : STOP
9000 REM collision
9010 BEEP 1,-12
9020 PRINT AT 21,0; "COLLISION!";
9030 IF ATTR (p,r)=40 THEN PRINT
AT 20,0; FLASH 1; INK 0; PAPER
7; s$(n); " has sunk!": LET s(n,1)
=-1
9040 IF ATTR (p,r)<n*8+n-1 THEN
GO TO 3100
9050 PRINT AT 20,0; FLASH 1; INK
0; PAPER 7; s$(n); " has sunk!":
LET s(n,1)=-1
9060 FOR a=1 TO 0 STEP -.1: BEEP
a,a*5-10: NEXT a

```

```

9070 PAUSE 100
9080 PRINT AT 20,0;TAB 31;" ";TA
B 31;" "
9090 GO TO 110
9100 LET s=ATTR (p,r)-40
9110 STOP
9120 REM ++++++
9130 REM submarine attack
9140 IF RND>.7 THEN GO TO 250
9150 BORDER 2: LET u=1: PRINT AT
2,21; PAPER 2; INK 7; " RED "
9160 FOR a=1 TO 7: BORDER 4: BEE
P .5,-5: BORDER 2: PAUSE 5: NEXT
a
9170 IF RND<0.5 THEN GO TO 5500
9180 PRINT AT 21,0; INK 7; BRIGHT
1; s$(c); " under attack!"
9190 PAUSE 50
9200 LET p=5
9210 PRINT AT 21,0; PAPER 6; FLA
SH 1; "Press 'F' and hope for the
best."
9220 FOR t=1 TO 2
9230 LET t1=FN r(10): LET pf=(t1
*8+164)-200
9240 PRINT AT 8,20+p; PAPER 5; I
NK 1; "+"
9250 LET p=p+(p<10)+1
9260 PAUSE 10
9270 PRINT AT 8,20+p; PAPER 5; I
NK c-(c<>5);d$(c)
9280 PAUSE 10
9290 IF INKEY$<>"f" THEN GO TO 5
9300
9310 FOR i=7 TO 1 STEP -1
9320 PLOT 200,62: DRAW INVERSE 1
: INK i;pf,40
9330 BEEP .05,a: NEXT i
9340 PLOT 200,62: DRAW INK 1;pf,
40: PAUSE 50
9350 IF t1=p THEN GO TO 5300
9360 PRINT AT 21,0; PAPER 2; INK
7; " That was good evasive acti
on!"
9370 PAUSE 100: PRINT AT 21,3; "W
atch out! Another torpedo!": FOR
a=1 TO 7: BEEP .05,a: NEXT a
9380 PAUSE 50: PRINT AT 21,0;TAB
31;" "
9390 NEXT t
9400 PRINT AT 20,0; "False Alarm!
Damage ";u-1; " Hits"

```



```

6250 PAUSE 50: PRINT AT 20,0;TAB
631;" ";TAB 31;" "
6350 GO TO 250
6400 REM hit
6410 IF U=1 THEN PRINT AT 21,0;"
a HIT!"; BEEP .8,-5: PAUSE 50: P
RINT AT 20,6;"The ";s$(c);" has
ejected all it's depth charges
": LET s(c,3)=0
6420 IF U=2 THEN PRINT FLASH 1;
INK 7; PAPER 0; BRIGHT 1;AT 21,0
;s$(c);" has been lost!";
6430 LET U=U+1: PAUSE 100
6440 IF U=2 THEN NEXT t
6450 PRINT BRIGHT 1;AT s(c,2),s(
r,1); INK 0; PAPER 5;"|"
6460 LET s(c,1)=-1
6470 FOR a=1 TO 0 STEP -.1
6480 BEEP a,a*5-10
6490 NEXT a
6500 FOR a=1 TO 20: BEEP 0.0025,
a: NEXT a
6510 GO TO 250
6520 REM submarine in close prox
imity
6530 PRINT AT 21,0; FLASH 1; BRI
GHT 1; INK 7;"Alert!"
6540 PAUSE 100
6550 PRINT AT 21,0;s$(c);" has s
potted sub!";
6560 PAUSE 100
6570 GO TO 250
6580 GO TO 9999
7000 REM display
7010 BORDER 4: PAPER 5: INK 1: B
RIGHT 0: CLS
7020 FOR i=0 TO 19: PRINT BRIGHT
1;"|";REM 2
0 graphic
7030 NEXT i
7040 PLOT 0,15: DRAW 160,0: DRAW
0,160
7050 REM mines
7060 LET M=FN r(10)+10
7070 FOR i=1 TO 8: LET mx=FN r(1
9): LET my=FN r(19): IF ATTR (M#
,mx)>64 THEN PRINT AT my,mx;"#"
7075 NEXT i
7080 REM ships
7090 FOR a=1 TO 6: PRINT AT s(a,
2),s(a,1); OVER 1; BRIGHT 1; PAP
ER 7; INK a-1;d$(a): NEXT a

```

```

7100 PRINT AT 0,21; FLASH 1;"SUB
SEARCH"; FLASH 0;AT 1,21;"CONDI
TION";AT 2,21: PAPER 4;"GREEN"
7110 REM periscope
7120 CIRCLE 208,102,40
7130 FOR p=40 TO 0 STEP -1
7140 PLOT 208-p,102: DRAW p*2,0,
PI
7150 NEXT p
7160 FOR i=0 TO 9: PRINT OVER 1;
AT 4+i,25;" + ": LET i=i+(i=3): N
EXT i
7170 PRINT OVER 1;AT 8,21;"++++
++++"
7180 RETURN
8000 REM instructions
8010 BORDER 1: PAPER 5: INK 1: B
RIGHT 1: CLS
8020 DIM s$(6,15): DIM s(6,3): D
IM i$(1): LET U=0: LET i=1: LET
d$="": REM graphic b,c,c,c
,a,a
8030 FOR a=1 TO 5: BEEP .1,-a: P
RINT AT 1,9; INK 2; INVERSE 1;"S
UB SEARCH": BEEP .25,a: LET i=i-
1: NEXT a
8040 INPUT BRIGHT 1;"Do you want
to read instructions(y/n)? ";i
$
8050 IF i$<>"y" THEN GO TO 8210
8060 FOR a=1 TO 6: READ p$: READ
a$
8070 PRINT AT 3,0;p$'a$
8080 DIM b$(LEN p$): DIM t$(LEN
a$)
8090 INPUT "Press "; FLASH 1;"EN
TER"; FLASH 0;" to continue. ";
LINE p$
8100 PRINT AT 3,0;b$'t$: NEXT
a
8110 IF p$="n" THEN GO TO 8210
8120 IF p$<>"y" THEN INPUT "DO y
ou want to name your own file
(? Answer (y/n) ";p$: GO TO 8110
8130 REM rename ships
8140 POKE 23658,8: RESTORE 9300:
BORDER 4: PAPER 5: BRIGHT 0: CL
S
8150 FOR a=1 TO 6: READ c$
8160 PRINT AT 0,0;"What would yo
u like to call";c$;"
: INPUT n$

```

```

0170 IF LEN n$ > 15 OR n$ = "" THEN
INPUT "No you can't have that, try again"; n$: GO TO 0170
0180 LET s$(a) = n$
0190 PRINT PAPER 0-a; INK a-1; AT
a+5,0;n$; " "; d$(a); NEXT a
0200 POKE 23650,0: GO TO 0230
0210 REM name ships
0220 RESTORE 0010: FOR a=1 TO 6:
READ c$: LET s$(a) = c#: NEXT a
0230 FOR i=1 TO 6: READ d: LET s
(i,1) = i-1: LET s(i,2) = 20: LET s
(i,3) = d: NEXT i
0240 LET sx = FN r(20) - 1: LET sy = F
N r(20) - 1: LET sd = FN r(10)
0250 LET pc = FN b(): LET nb = 0: RE
TURN
0300 REM Data
0310 REM Define graphics
0320 FOR c=1 TO 14: READ c#
0330 FOR b=0 TO 7: READ bt: POKE
c#+b, bt: NEXT b
0340 NEXT c: RETURN
0350 DATA "a", 0, 0, 0, 10, 20, 127, 10
0360 REM patrol boat
0370 DATA "b", 15, 15, 50, 100, 255, 2
125, 0: REM battle cruiser
0380 DATA "c", 0, 0, 0, 204, 92, 255, 2
0390 REM frigate
0400 DATA "d", 137, 74, 44, 31, 240, 5
0, 90, 145: REM explosion
0410 DATA "e", 0, 40, 40, 120, 30, 110
0, 20, 0: REM mine
0420 DATA "f", 255, 128, 128, 128, 12
0, 128, 128, 128: REM grid
0430 DATA "g", 0, 0, 1, 1, 7, 1, 1, 0: R
EM 1
0440 DATA "h", 0, 0, 0, 0, 224, 0, 0, 0:
REM 1
0450 DATA "i", 0, 0, 0, 24, 24, 0, 0, 0:
REM dice no. 1
0460 DATA "j", 0, 96, 96, 0, 0, 6, 6, 0:
REM dice no. 2
0470 DATA "k", 0, 0, 0, 24, 24, 0, 102,
0: REM dice no. 3
0480 DATA "l", 0, 102, 102, 0, 0, 102,
102, 0: REM dice no. 4
0490 DATA "m", 195, 195, 0, 24, 24, 0,
195, 195: REM dice no. 5
0500 DATA "n", 102, 102, 0, 102, 102,
0, 102, 102: REM dice no. 6

```

```

0190 DATA "You control a fleet o
f warships consisting of: 1 Battl
e cruiser, 2 Patrol vessels and
0 frigates."
0200 DATA "Your task is to hunt
and destroy an enemy submarine wh
ich has been lurking around t
he shipping lanes near you."
0210 DATA "Your battlecruiser is
equipped with a sonar system
that rises in pitch as it gets c
loser to the sub."
0220 DATA "Each vessel can carry
only a limited amount of dep
th charges and once they are fir
ed the ship must return to port. T
he limits are 4 to the battle c
ruiser 3 to each of the
frigates and 2 to the patrol b
oats."

```

```

0230 DATA "The submarine may tor
pedo any ship that is within a
range of 5 units and is in lin
e either horizontally or verti
cally."
0240 DATA "The submarine moves 0
ne unit each round and may hi
de up to a depth of 10 units. If
you move 1 of your ships directl
y above the submarine an alarm wi
ll sound."

```

```

0250 DATA "The number of moves y
ou get each round depends on a
dice, you may move any number o
f ships as long as it is within
the limits."
0260 DATA "The computer will seq
uence through each ship and
ask you if you want to move it. A
nswer with 'ENTER' if you don't
or use the directions N,S,E,W or
'D' for depth charges."
0270 DATA "If you navigate your
ship into another ship then the
smaller ship will be sunk. If
you crash into mine your ship w
ill sink."

```

```

9280 DATA " ", "The ships under y
our command are -HMS INVINCIBLE (
your flagship) -HMS SHEFFIELD (f
rigate 1) -RNZS OTAGO (frig
ate 2) -RNZS CANTERBURY
(frigate 3) -HMS BRITANNIA (p
atrol boat 1) -USS TITANIC (pat
rol boat 2)"
9290 DATA "Would you like to nam
e your own fleet? Answer (y/n)."
9300 DATA "your Battlecruiser (s
tagship)", "Frigate 1", "Frigate 2
", "Frigate 3", "Patrol boat 1", "P
atrol boat 2"
9310 DATA "HMS INVINCIBLE", "HMS
SHEFFIELD", "RNZS OTAGO", "RNZS CA
NTERBURY", "HMS BRITANNIA", "USS T
ITANIC"
9320 DATA 4,3,3,3,2,2
9999 STOP : BORDER 7: PAPER 7: I
NK 0: BRIGHT 0: INVERSE 0: CLS :
STOP

```

## Hangman

This is a superior version (by Raymond Blake) of the ever-popular hangman game. It has been designed so you can customise it to your own requirements and the word list has been left small deliberately to encourage this.

To alter the words, simple change the 30 in line 70 to the total number of words in the list, and enter the words into DATA statements at the end of the program in the same fashion as lines 1500 to 1520. Witty comments and the scope for alterations can make running this game a long-lasting enjoyable experience. It is even better if you can get a friend to change the word list for you, so you don't know what the words are. As stated in line 20, CAPS LOCK must be engaged before running the program.

```

20 REM ENGAGE CAPS LOCK
BEFORE RUNNING
30 LET F=0: RANDOMIZE
40 BORDER 5: PAPER 7: INK 0: 0
LS : RESTORE
40 PRINT AT 0,7: "SPECTRUM HANG
MAN": INK 0: AT 1,7: "-----"
50 IF F=0 THEN PRINT AT 0,0: "I
SAY: THESE WORDS WILL BE
AT 0,1: "IF YOU CAN'T GUESS THEIR
WORD."
50 IF F=1 THEN PRINT AT 0,0: "O
K. NOW THEY WANT TO BE
"friend! You've got to help his
60 LET U=0: LET W$="": LET C=0

```

```

70 FOR I=1 TO INT (RAND*100+1):
READ B#: NEXT I
75 FOR I=100 TO 99: BEND ,B#,I
: NEXT I
80 LET L=LEN B#: LET B#="-----
-----" ( TO L)
90 PRINT AT 10,0: INK 0; B#
100 PRINT AT 0,0: INK 4; B#
1000 SUBPR:
120 INPUT C#: IF LEN C# < 1 AND
LEN C# > 1 THEN GO TO 100
130 IF C#="D" OR C#="I" THEN GO
TO 100
140 PRINT AT 0,0;"

150 IF LEN C#=1 THEN GO TO 100
160 IF C#="B" THEN LET B#=C#
170 IF C#="A" THEN LET C#=""
175 GO TO 100
180 LET C=0: FOR I=1 TO L: IF B
#(I) = C# THEN LET B#(I) = C#
190 IF C#="B" THEN LET B#(I) = B#(I) + 1
200 NEXT I
210 PRINT AT 10,0: INK 0; B#: IF
C#="" THEN LET C#="A": GO TO 100
220 C#="A" AND LET C#="B": GO TO 100
230 C#="B" AND LET C#="A": GO TO 100
240 IF C#="A" THEN FOR I=100 TO 99
: BEND ,B#,I: NEXT I
250 PRINT AT 10,0: "Word letter
S: PRINT AT 10,0: INK 4; B#
260 IF B#="B" THEN GO TO 100
270 IF C#="A" THEN GO TO 100
280 GO TO 100
300 PRINT AT 0,0: INK 0; "0000!
Next time you see a "000
t come running to me, motor!"
310 FOR I=1000 TO 100: BEND ,I: B#
320 B#: NEXT I: GO TO 100
330 PRINT AT 0,0: INK 0; "0000
S: VERT SUCT, BEND ,B#,I: BEND
340 TO 100: BEND ,B#,I: BEND ,I
: NEXT I
350 PRINT AT 10,0: INK 1; "The #
of #s: AT 10,0; B#
360 BEND ,B#,I: BEND ,I: B#
370 LET T=10: GO TO 30
380 FOR L=10 TO 10: PLOT 100,0:
DRAW 0,0: NEXT L

```

```

810 FOR L=10 TO 10: PLOT 0,10
: DRAW 0,100: NEXT L: RETURN
820 FOR L=1001 TO 200: PLOT 0,10
: DRAW 10,10: NEXT L
830 FOR L=101 TO 20: PLOT 100,0:
DRAW 10,10: NEXT L: RETURN
840 FOR L=111 TO 110: PLOT 111,
0: DRAW 10,10: NEXT L
850 FOR L=1001 TO 200: PLOT 0,11
0: DRAW 10,10: NEXT L: RETURN
860 PLOT 171,110: DRAW 0,100: R
ETURN
880 CIRCLE 171,89,7: PLOT 100,0
0: PLOT 170,00
890 PLOT 100,00: DRAW 4,0,10+10
I: RETURN
900 FOR L=170 TO 172: PLOT 0,70
: DRAW 0,14: NEXT L: RETURN
920 FOR L=70 TO 72: PLOT 100,0:
DRAW 10,4: NEXT L: RETURN
940 FOR L=70 TO 72: PLOT 170,0:
DRAW 10,4: NEXT L: RETURN
960 FOR L=100 TO 80: PLOT 171,0:
DRAW 17,10: NEXT L: RETURN
980 FOR L=100 TO 80: PLOT 171,0:
DRAW 7,10: NEXT L: RETURN
1000 DATA "MOTOPE", "POKER", "PLA
NET", "SPHERE", "FALLOUT", "MOUND",
"KANGAROO", "ATTRIBUTE", "DEADLI
NE", "ROUTE"
1010 DATA "HOTEL", "GALAXY", "UNIV
ERSE", "MIRAGE", "BAINSTONE", "QUIV
ER", "COLOUR", "RAINBOW", "LAUGH", "
VISION"
1020 DATA "SLATE", "USHERETTE", "F
ELLOW", "PRINTER", "COMPUTER", "POL
DER", "DIVISION", "MAGAZINE", "WAGE
T", "JUMPER"

```

# SPACE GAMES



## Stellar Evade

This tiny program, written by Graham Charlton, produces a surprisingly entertaining game. You use the "z" and "m" keys to move left and right to dodge the asterisks coming up at you from below. (Note that the odd thing after the second equals sign in line 90 is an "m".)

```
10 BORDER 0: PAPER 0: CLS
20 LET a=10: LET b=10: LET c=2
30 LET t=0
40 POKE 23692,0
45 LET t=t+1
50 PRINT AT 21,31;" "
60 FOR z=1 TO 5
70 INK 7: PRINT AT c,AND+21;"#
80 NEXT z
90 LET b=b+(INKEY#="z" AND b<3
91)-(INKEY#="m" AND b>1)
100 IF SCREEN# (b,3)="#" THEN a
110 PRINT AT 2,3:"YOU SCORED ";t:
120 FOR x=1 TO 200: NEXT x: GOTO
130 PRINT AT b,3:"@"
140 GO TO 30
```

# Space Trek

Skilled programmer David Perry wrote this program for the book. In this game, you fly through each galaxy, avoiding the stars, aiming to shoot all the enemy alien battle cruisers. Once you've finished, you must leave the galaxy via the Black Hole to penetrate into deeper space.

Full instructions are included within the program. You move around using the cursor keys, with "O" to fire. The pound signs should be entered as hash (#) symbols.

```
1 REM
2 REM   SPACE TREK!
3 REM   #####
4 REM
5 REM   COPYRIGHT
6 REM   DAVID PERRY
7 REM   1983
8 REM
9 POKE 23658,8: LET NUM=10: D
IM N$(NUM+1,8): DIM N(NUM+1): FO
R N=1 TO NUM: LET N(N)=(1000-(10
0*N)): LET N$(N)="SPECTRUM": NEX
T N: GO SUB 2000: LET N$(1)="DAV
E OK!": LET N(1)=5000
10 BORDER 1: PAPER 0: INK 7: B
RIGHT 1: CLS
20 LET GAL=1: LET BON=3: LET S
C=0: LET LEV=3: LET MEN=3
```

```
30 PRINT AT 0,0: INK 5: BRIGHT
0:"
":AT 21,0:"
"
35 FOR N=1 TO 3: PRINT INK 5:
AT N,0:" ";AT N,31:" ": NEXT N:
FOR N=4 TO 20: PRINT INK 5:AT N
,0:" ";AT N,31:" ": NEXT N
40 IF MEN=0 THEN GO TO 1000
42 FOR n=1 TO LEV
45 LET X=INT (RND*14)+6: LET Y
=INT (RND*29)+1: IF SCREEN# (X,Y
)<>" " THEN GO TO 45
50 PRINT AT X,Y: INK 2:"K"
51 NEXT N
54 FOR J=1 TO BON*2
55 LET X=INT (RND*14)+6: LET Y
=INT (RND*30)+1: IF SCREEN# (X,Y
)<>" " THEN GO TO 55
60 PRINT AT X,Y: INK 6:"*"
70 NEXT J
75 LET X=INT (RND*14)+6: LET Y
=INT (RND*29)+2: IF SCREEN# (X,Y
)<>" " THEN GO TO 75
80 PRINT AT X,Y: INK 7:"O"
90 PRINT AT 1,3: INK 4:"SHIPS
LEFT:";: FOR N=1 TO MEN: PRINT
INK 7:"a";: NEXT N: PRINT AT 1,1
9: INK 6:"SCORE:";SC
100 PRINT AT 2,8: INK 3:"HIGHE
T SCORE:";N(1): PRINT AT 4,0: IN
K 5:"
"
105 PRINT AT 3,4: INK 2:"GALAXY
:";GAL: PRINT AT 3,14: INK 6:"KL
INGONS LEFT:";LEV
```



```

110 LET D=2: LET A=5: LET B=1
120 PRINT AT A,B: " "
130 IF D=1 THEN LET A=A-1
140 IF D=2 THEN LET B=B+1
150 IF D=3 THEN LET A=A+1
160 IF D=4 THEN LET B=B-1
163 IF B<1 THEN LET B=30
164 IF B>30 THEN LET B=1
165 IF A<5 THEN LET A=20
166 IF A>20 THEN LET A=5
167 IF LEV=0 THEN BEEP .01,50:
IF SCREEN$(A,B)="0" THEN LET
BON=BON+2: LET GAL=GAL+1: PRINT
AT 3,11;GAL: GO SUB 800: LET LEV
=BON: GO TO 40
170 IF SCREEN$(A,B)<>" " THEN
FOR I=7 TO 0 STEP -1: PRINT AT
A,B: INK I:"X": BEEP .02,I*7: NE
XT I: GO SUB 800: PRINT AT 1,13:
" ": LET MEN=MEN-1: FOR N=1 T
O MEN: PRINT AT 1,13+N:"a": NEXT
N: GO TO 40
180 IF INKEY$="7" THEN LET D=1
190 IF INKEY$="8" THEN LET D=2
200 IF INKEY$="6" THEN LET D=3
210 IF INKEY$="5" THEN LET D=4
220 IF D=1 THEN PRINT AT A,B:
INK 6:"a"
230 IF D=2 THEN PRINT AT A,B:
INK 6:"b"
240 IF D=3 THEN PRINT AT A,B:
INK 6:"c"
250 IF D=4 THEN PRINT AT A,B:
INK 6:"d"
255 IF INKEY$="0" THEN GO SUB
300

```

```

260 BEEP .01,D*10: GO TO 120
300 IF D=1 THEN GO TO 400
310 IF D=2 THEN GO TO 500
320 IF D=3 THEN GO TO 600
330 IF D=4 THEN GO TO 700
340 STOP
400 IF A=5 THEN RETURN
405 FOR N=A-1 TO 5 STEP -1
410 LET A$=SCREEN$(N,B)
420 IF A$="K" THEN LET LEV=LEV
-1: PRINT AT 3,28;LEV:" ": LET
SC=SC+100: PRINT AT 1,25;SC: FOR
I=7 TO 0 STEP -1: PRINT AT N,B:
INK I:"K": BEEP .01,I*7: NEXT I
: PRINT AT N,B:" ": NEXT N
430 IF A$="*" OR A$="0" THEN B
EEP .03,10: LET N=6: RETURN
440 PRINT AT N,B: ("f" AND N>5):
BEEP .01,N
450 PRINT AT N,B: (" " AND N>5):
NEXT N: RETURN
500 IF B=30 THEN RETURN
505 FOR N=B+1 TO 30
510 LET A$=SCREEN$(A,N)
520 IF A$="K" THEN LET LEV=LEV
-1: PRINT AT 3,28;LEV:" ": LET
SC=SC+100: PRINT AT 1,25;SC: FOR
I=7 TO 0 STEP -1: PRINT AT A,N:
INK I:"K": BEEP .01,I*7: NEXT I
: PRINT AT A,N:" ": NEXT N
530 IF A$="*" OR A$="0" THEN B
EEP .03,10: LET N=30: RETURN
540 PRINT AT A,N: ("e" AND N<31)
: BEEP .01,N
550 PRINT AT A,N: (" " AND N<31)
: NEXT N: RETURN

```

```

600 IF A=20 THEN RETURN
605 FOR N=A+1 TO 20
610 LET A$=SCREEN$(N,B)
620 IF A$="K" THEN LET LEV=LEV
-1: PRINT AT 3,28;LEV;" ": LET
SC=SC+100: PRINT AT 1,25;SC: FOR
I=7 TO 0 STEP -1: PRINT AT N,B;
INK I;"K": BEEP .01,I*7: NEXT I
: PRINT AT N,B;" ": NEXT N
630 IF A$="*" OR A$="0" THEN B
EEP .03,10: LET N=20: RETURN
640 PRINT AT N,B;("f" AND N<21)
: BEEP .01,N
650 PRINT AT N,B;(" " AND N<21)
: NEXT N: RETURN
700 IF B=1 THEN RETURN
705 FOR N=B-1 TO 1 STEP -1
710 LET A$=SCREEN$(A,N)
720 IF A$="K" THEN LET LEV=LEV
-1: PRINT AT 3,28;LEV;" ": LET
SC=SC+100: PRINT AT 1,25;SC: FOR
I=7 TO 0 STEP -1: PRINT AT A,N;
INK I;"K": BEEP .01,I*7: NEXT I
: PRINT AT A,N;" ": NEXT N
730 IF A$="*" OR A$="0" THEN B
EEP .03,10: LET N=1: RETURN
740 PRINT AT A,N;("e" AND N>0):
BEEP .01,N
750 PRINT AT A,N;(" " AND N>0):
NEXT N: RETURN
801 LET B=2: FOR N=20 TO 5 STEP
-1
810 BEEP .02,N*2: PRINT AT N,0;
INK 5;"
"
820 NEXT N

```

```

830 PRINT AT 3,31; INK 5;" ": P
RINT AT 21,1; INK 5;"
": RETURN
1000 PRINT AT 10,4; PAPER 7; INK
0; BRIGHT 0;" YOU HAVE SCORED "
;SC;" "
1010 BEEP .1,24: BEEP .1,23: BEE
P .84,24: BEEP .5,21: BEEP .5,20
: BEEP .85,21
1500 IF SC<=N(10) THEN PAUSE 50
: GO TO 1540
1502 LET NUM=11: IF SC>=N(NUM) T
HEN INPUT "ENTER 8 INITIALS! ";P
$: IF LEN P$>8 THEN GO TO 1500
1505 IF SC>=N(NUM) THEN LET N(N
UM)=SC: LET N$(NUM)=P$
1510 FOR A=1 TO (NUM-1): LET B$=
N$(A): LET C$=N$(A+1): LET B=N(A
): LET C=N(A+1): IF B<C THEN LE
T N(A)=C: LET N(A+1)=B: LET N$(A
)=C$: LET N$(A+1)=B$
1520 NEXT A: FOR N=1 TO NUM-1: I
F N(N)<N(N+1) THEN GO TO 1510
1530 NEXT N
1540 CLS
1550 PRINT AT 2,4;"H A L L O F
F A M E !"
1560 PRINT AT 3,4;"=====
=====
"
1580 FOR N=1 TO NUM-1: PRINT AT
N+5,7; INK 6;"( "; INK 2;N; INK
6;")";AT N+5,12; INK 7;N(N): PRI
NT AT N+5,17; INK 5;N$(N): NEXT
N

```

```

1590 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
PRINT AT 17,8; PAPER 2; INK 7;"
PRESS ANY KEY! ": LET I=0
1600 LET I=I+1: IF I>7 THEN LET
I=0
1610 PRINT AT 2,4; INK I;"H A L
L O F F A M E !"
1620 BEEP .01,I*7: PAUSE 2: IF I
NKEY#="" THEN GO TO 1600
1630 GO TO 10
2000 BORDER 1: PAPER 0: INK 7: B
RIGHT 1: CLS
2030 PRINT AT 2,6;"S P A C E T
R E K !"
2040 PRINT AT 3,6;"XXXXXXXXXXXXX
XXXXXXXX"
2050 PRINT AT 6,1;"CONTROL YOUR
SPACESHIP USING THE CURSOR KE
YS AND 'O' TO FIRE. THE OBJ
ECT OF THE GAME IS TO PASS TH
ROUGH ONE GALAXY AFTER ANOTHER
WEAVING BETWEEN THE STARS TO
SHOOT THE ENEMY"
2060 PRINT " KLINGON BATTLE CRUI
SERS.WHEN THEY ARE ALL DESTRO
YED YOU CAN LEAVE VIA THE B
LACK HOLE."
2065 RESTORE 2085: FOR C=0 TO 7:
READ HAT: POKE USR "A"+C,HAT: N
EXT C
2070 PRINT : PRINT " YOU 'a'
BLACK HOLE '"; INK 7;"O'"
2080 PRINT : PRINT " STAR '"
; INK 6;"*"; INK 7;"' KLINGON '
"; INK 2;"K"; INK 7;"'"

```

```

2081 PRINT : PRINT TAB 2; PAPER
2; INK 7;" PRESS A KEY TO COMMEN
CE! "
2082 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
2084 FOR C=0 TO 39: READ HAT: PO
KE USR "B"+C,HAT: NEXT C
2085 DATA 24,36,36,36,60,90,102,
66
2086 DATA 0,224,94,49,49,94,224,
0
2087 DATA 66,102,90,60,36,36,36,
24
2088 DATA 0,7,122,140,140,122,7,
0
2089 DATA 0,0,0,255,255,0,0,0
2090 DATA 24,24,24,24,24,24,24,2
4
2110 LET I=0
2120 LET I=I+1: IF I>7 THEN LET
I=0
2130 PRINT AT 2,6; INK I;"S P A
C E T R E K !"
2140 BEEP .01,50-(I*7): PAUSE 2:
IF INKEY#="" THEN GO TO 2120
2150 RETURN
9998 FOR N=0 TO 7: PRINT INK N:
" ";: NEXT N
9999 GO TO 9998

```

# Stellar Probe

This is a fascinating game in which, once again, you patrol a sector of the galaxy. As you might have guessed, there are aliens at the bottom of the galaxy, and it is your task to find them, and blast hell out of them.

The program shows a map of the current Galactic Sector, along with its important features. You'll quickly learn how to read the map once you run the game a few times.

At any time you can move, scan or fire. Your scanners operate in two ways. The short range scanner, which looks into the eight squares immediately surrounding you, consumes little energy. The long range scanners look two squares in a single direction, and use up more energy.

The game ends if you land on top of an alien ship. The aliens do not move around during a single game. You have limited reserves in your energy bank and must try and kill as many aliens as you can before your energy is totally depleted.

An alien (and the program invents names for the aliens for each game) can only shoot back after you have fired your laser at it (thus revealing your position) and if the alien is within a single square of

you. Damage to your ship from an accurate alien shot is shown in energy terms (that is, energy is drained from your bank). The game continues until you land on an alien ship, or run out of energy. Tim Rogers converted this program from an original program of mine.

```
10 BORDER 0: RANDOMIZE
20 INK 6
30 PAPER 0
40 CLS
50 GO SUB 2000: GO SUB 1000
70 GO SUB 3000
110 GO TO 70
1010 DIM a(10,10)
1020 DIM b(10,10)
1030 FOR a=1 TO 20
1060 LET a(FN a(x),FN a(x))=1
1100 NEXT a
1110 LET a(u,p)=0
1120 LET b(u,p)=0
1130 LET r0=FN a(0)
1135 LET r1=FN a(1)
1135 LET r2=FN a(2)
1140 RETURN
1510 LET b1=FN a(10)
1520 BEEP .1,b1
1530 BEEP .1,b1-10
1540 RETURN
2000 DEF FN a(x)=INT (RND*x)+1
2010 DEF FN a#(i)=x#(r0, TO t(r0,
1)+a#(r0,1, TO t(r0,1,2))
2015 LET x=10
2020 LET u=5: LET p=5
2030 LET a#(u,p)=0
2040 DIM c#(5,5)
2050 FOR a=1 TO 5
2060 READ c#(a)
2070 NEXT a
2080 LET a1=0
2090 DIM t(5): DIM c(5)
2100 FOR a=1 TO 5
2110 READ t(a): READ c(a)
2120 NEXT a
2130 LET e=1000+2000*RND
2140 DIM x#(10,0)
2150 DIM u#(10,0)
2160 DIM t(0,2)
```

```

0000 FOR a=1 TO 9
0005 READ c#: LET t(a,1)=LEN c#
0010 LET x#(a)=c#
0015 READ c#: LET t(a,2)=LEN c#
0020 LET y#(a)=c#
0025 NEXT a
0030 DIM s#(10,11)
0035 FOR a=1 TO 9
0040 LET r0=FN a(8)
0045 LET r01=FN a(8)
0050 LET s#(a)=FN a#()
0055 NEXT a
0060 FOR a=1 TO 3
0065 FOR b=0 TO 7
0070 READ c
0075 POKE USR CHR# (143+a)+b,c
0080 NEXT b: NEXT a
0085 RETURN
0090 CLS
0095 PRINT AT 0,0;"STAR ";
0100 GO SUB 1500
0105 PRINT "MAP ";
0110 GO SUB 1500
0115 PRINT "REPD-OUT"
0120 GO SUB 1500
0125 FOR a=1 TO 32
0130 PRINT " ";
0135 NEXT a
0140 GO SUB 1500
0145 PRINT AT 3,0;" 1234567890"
0150 FOR a=1 TO 10
0155 IF a<10 THEN PRINT a: GO TO
0 0160
0165 PRINT "0";
0170 FOR b=1 TO 10
0175 LET b#=s#(s(a,b)+1)
0180 PRINT INVERSE 1;b#;
0185 IF b#<>" " THEN GO SUB 1500
0190 NEXT b
0195 IF a<10 THEN PRINT a: GO TO
00000
0200 PRINT "0"
0205 NEXT a
0210 PRINT " 1234567890"
0215 PRINT AT 5,14; INK 0; ORDER
7; FLASH 1;"Energy Banks ";INT
a
0220 PRINT AT 6,14;
0225 FOR a=1 TO a/200
0230 PRINT INK 0;"█";
0235 NEXT a
0240 GO SUB 1500

```

```

3570 IF a<1 THEN GO TO 4000
3580 IF at=0 THEN PRINT AT 8,14;
"alien ships";AT 9,14;"destroyed
";at: GO SUB 1500
3590 PRINT AT 10,14;"Ship is loc
ated"
3600 PRINT AT 11,14;"at sub-quad
700t"
3610 PRINT AT 12,14; FLASH 1;"{
";"D";"P";"3"
3620 GO SUB 1500
3630 PRINT AT 13,14;"in ";s#(INT
(U*P/10+.5));AT 14,14;"sector"
3640 PRINT AT 16,0;
3645 GO SUB 1500
3650 GO TO 4500
4010 GO SUB 5000
4020 PRINT FLASH 1;"Energy banks
completely drained"
4030 PRINT "We killed ";at;" ali
ens"
4110 PRINT ",c#(FN a(5));": Do y
ou want to be","captain again, s
ir?
4120 IF INKEY#<>" " THEN GO TO 41
20
4130 IF INKEY#="" THEN GO TO 413
0
4140 IF INKEY#<>"r" AND INKEY#<>
"N" THEN RUN
4160 PRINT ",c#(FN a(5));": You
always were"
4170 PRINT "a bit of a ";
4180 LET r0=FN a(8)
4190 LET r01=FN a(8)
4200 PRINT FN a#()
4210 PRINT END
4500 GO SUB 5000
4510 PRINT AT 16,0;c#(FN a(5));"
: Your order, Sir?"
4515 BEEP .2,RND#50
4520 PRINT AT 17,12;"1 Scan"
4530 GO SUB 1500
4540 PRINT TAB 12;"2 Move"
4550 GO SUB 1500
4560 PRINT TAB 12;"3 Fire"
4570 GO SUB 1500
4580 PRINT AT 21,0;"Press 1 2 or
3 on your console"
4590 LET i#=INKEY#

```

```

4692 IF CODE I#<49 OR CODE I#>51
  THEN GO TO 4690
4695 GO SUB 1500
4698 GO SUB 5000
4610 IF I#="1" THEN GO TO 6000
4620 GO SUB 5000+1000#VAL I#
4630 FOR a=0 TO 20
4640 PRINT AT 21,a;"■■■■■"
4650 BEEP .01,a+10
4660 BEEP .01,a
4670 PRINT AT 21,a;" "
4680 NEXT a
4690 PRINT AT 21,a;" "
4700 RETURN
5005 PRINT AT 16,0;
5010 FOR a=1 TO 5
5020 PRINT "
5030 NEXT a
5040 PRINT AT 16,0;
5050 RETURN
6005 PRINT TAB 11; FLASH 1; INK
7;"■■■■■ SCANNER ■■■■■"
6010 PRINT c$(FN a(5));"- "
6020 PRINT "(long 2) or short 1)
range, sir?"
6030 LET I#=INKEY#
6040 BEEP .01,0
6050 IF CODE I#<49 OR CODE I#>50
  THEN GO TO 6030
6060 LET K=VAL I#
6070 LET e=e-10*K
6080 IF K=2 THEN GO TO 6500
6090 PRINT AT 16,0;" scanning.-
.
6095 LET f=0
6100 FOR a=-(U>1) TO (U<10)
6110 FOR b=-(P>1) TO (P<10)
6115 BEEP .01,0
6120 IF g(U+a,P+b)=1 THEN LET f=
1
6130 NEXT b: NEXT a
6140 IF f=1 THEN PRINT z$;" ship
in vicinity, sir.": GO TO 6160
6150 PRINT "Nothing in vicinity.
.
6155 GO SUB 1500
6160 PRINT AT 21,0;"Scanner off,
sir?"
6165 BEEP .1,RND*50
6170 IF INKEY#="" THEN GO TO 617
0

```

```

6175 GO SUB 1500
6180 IF INKEY#="D" OR INKEY#="N"
  THEN GO SUB 5000: GO TO 6010
6190 GO TO 4600
6500 PRINT AT 16,0;c$(FN a(5));"
: Direction?
6510 PRINT "north:1 east:2 south
:3 west:4"
6515 IF INKEY#<>"" THEN GO TO 65
15
6520 LET I#=INKEY#
6530 BEEP .01,0
6540 IF CODE I#<49 OR CODE I#>52
  THEN GO TO 6520
6550 LET K=(VAL I#-1)*2+1
6560 LET z=(g(U+b(K)*2,P+c(K)*2)
=1)
6570 IF z=1 THEN PRINT FLASH 1;"
Positive": GO TO 6590
6580 PRINT "negative"
6590 GO SUB 1500
6595 IF INKEY#<>"" THEN GO TO 65
95
6600 GO TO 6160
7002 LET f1=0
7005 LET e=e-50
7010 PRINT "Bearings:"
7015 GO SUB 1500
7020 CIRCLE 40,24,16
7030 PLOT 40,24
7040 DRAW 0,16
7050 GO SUB 1500
7060 PRINT AT 16,16;"000"
7065 PLOT 40,24
7070 DRAW 16,0
7100 GO SUB 1500
7110 PRINT AT 16,16;"090"
7130 PLOT 40,24
7140 DRAW 0,-16
7150 GO SUB 1500
7160 PRINT AT 20,16;"180"
7180 PLOT 40,24
7190 DRAW -16,0
7200 GO SUB 1500
7210 PRINT AT 16,14;"270"
7230 INPUT "Degrees? ";r
7240 IF r>315 OR r<0 THEN GO TO
7230
7250 LET r=INT (r/45)+1
7255 IF f1=1 THEN RETURN
7260 LET g(U,P)=0
7270 LET s(U,P)=3

```

```

7280 LET U=U+b(r)
7290 LET P=P+c(r)
7300 IF g(U,P)<>1 THEN GO TO 111
0
7310 PRINT AT RND*21,0; FLASH 1;
INK 2; PAPER 6;"You have landed
on a ";z$;" ship!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!"
7320 BEEP 1,0
7325 IF RND>.95 THEN GO TO 7310
7330 FOR a=1 TO RND*5
7340 PLOT 120,87
7350 DRAW 120-RND*255,87-RND*175
7360 NEXT a
7370 BEEP .3,-23
7380 GO TO 7320
8005 LET fl=1
8010 PRINT AT 14,12; INK 6; PAPE
R 2; FLASH 1;"RED ALERT"
8020 GO SUB 1520; GO SUB 1500
8030 PRINT AT 15,0;c$(FN a(5));"
- Direction of fire:"
8040 GO SUB 7015
8045 LET e=e-100
8050 IF g(U+b(r),P+c(r))<>1 THEN
GO TO 8000
8070 BEEP 1,20
8080 GO SUB 5000
8090 PRINT c$(FN a(5));" - You h
it the ";z$;" sir!"
8095 LET s(U+b(r),P+c(r))=4
8097 LET g(U+b(r),P+c(r))=4
8100 LET at=at+1
8120 PRINT s;c$(FN a(5));" - Sha
ll I turn";" the red alert off, s
ir?"
8130 IF INKEY#<>"" THEN GO TO 81
40
8135 IF INKEY#="" THEN GO TO 813
5
8140 GO TO 1110
8000 GO SUB 5000
8310 PRINT c$(FN a(5));".";
8320 LET s(U+b(r),P+c(r))=3
8330 PRINT " We missed, sir."
8340 FOR a=0 TO 30
8350 BEEP .1,a
8360 NEXT a
8370 BEEP 1,35
8380 FOR a=1 TO 50
8390 BORDER RND*7
8400 NEXT a

```

```

8410 BORDER 0
8420 PRINT s;c$(FN a(5));": The
";z$;" are";" shooting back!!"
8430 GO SUB 1500
8450 IF RND>.6 THEN GO TO 8700
8460 PRINT "They HIT US SIR!!!!"
8470 LET e=e-100*AND
8480 GO TO 8110
8700 PRINT c$(FN a(5));": But th
ank the stars-"
8710 PRINT "The ";z$;" missed us
!"
8720 GO TO 8110
9020 DATA "Dr Sock","Spottie","L
t. Looku","Checkout","Zulu"
9030 DATA -1,0,-1,1,0,1,1,1,1,0,
1,-1,0,-1,-1,-1
9040 DATA "Glob","Ulian","Frax",
"tloid","Mesh","nix","Rom","ulan
","Grup","fish","Krell","on","OU
es","tar","Coch","rane"
9050 DATA 255,165,195,153,153,19
5,165,255
9060 DATA 255,189,219,255,255,21
0,189,255
9070 DATA 248,152,152,255,47,57,
249,253

```



# Lunar Storm

David Perry's LUNAR STORM uses sophisticated string-handling to provide a mass of meteors and spaceships moving sideways on the screen.

Below this is the surface of the planet McClariana (named after the intrepid explorer O. J. McClaren who discovered the planet in 2068). You have to try and weave your way between these obstacles, and land on one of the three landing pads.

To simplify an extremely difficult task, you'll see you've been provided with a thrust button, to allow you to high tail it out of there when the going gets too rough. The game uses lots of sound and color, plus some rather clever UDG's.

**LUNAR  
STORM**

ENTER LEVEL (1 TO 3)  
=====

5-SHIP LEFT 8-SHIP RIGHT  
0-THRUST

```

1 REM
2 REM
3 REM
4 REM
5 REM
6 REM
7 REM
8 BORDER 0: PAPER 0: INK 7: C
LS
9 LET SC=0: LET BC=100: LET I
=7: GO TO 45
10 CLS : RESTORE : GO SUB 29
11 LET X=0: LET Y=INT (RND*31)
+1
12 PRINT AT X,Y," ": PRINT AT
0,0: INK 5: a#: AT 7,0: a#: AT 11,0:
a#: AT 15,0: a#: AT 19,0: a#
13 PRINT AT 5,0: INK 5: b#: AT 9
,0: b#: AT 13,0: b#: AT 17,0: b#
14 IF INT X>18 THEN GO TO 79
15 IF SCREEN$ (X,Y) <> " " THEN
GO TO 36: PRINT AT X,Y,"@"
16 BEEP .01,X*2
17 PRINT AT X,Y," "
18 LET X=X+.5: IF IN 61406=254
THEN LET X=X-.5: LET SC=SC-1
19 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
20 IF Y<0 THEN LET Y=31
21 IF Y>31 THEN LET Y=0
22 IF X>18 THEN GO TO 79
23 IF SCREEN$ (X,Y) <> " " THEN
GO TO 36
24 LET SC=SC+1
25 PRINT AT X,Y: INK 7: BRIGHT
0:"@"
26 LET a#=a#(2 TO )+a#(1)
27 LET b#=b#(31)+b#( TO 30)
28 GO TO 12
29 PRINT AT 20,0: INK 2:"
30 PRINT AT 19,0: INK 3:"
FLASH 1:"": FLASH 0: INK
INK 3: FLASH 1:"": FLASH 0:
FLASH 1:"": FLASH 1:"": INK 7: F
LASH 0:"": INK 3: FLASH 1:"":
FLASH 0:"": FLASH 1:"":
FLASH 0: INK 7: INK 3: FLA
SH 1:"": FLASH 0:"":
31 FOR a=1 TO 3: READ c#: FOR
a=0 TO 7: READ b: POKE USR c#+a,
b: NEXT a: NEXT a

```

```

32 DATA "c",24,36,90,165,219,2
19,189,195
33 DATA "a",0,60,126,171,213,1
25,60,0
34 DATA "b",195,189,90,126,126
,90,189,195
35 RETURN
36 FOR n=7 TO 0 STEP -1
37 PRINT AT x,u; INK n;" "
38 BEEP .01,n*3: BEEP .01,50-n
*3
39 NEXT n
40 LET I=7
41 PRINT AT 2,8; INK 5;"S C D
R E :";SC
42 PRINT INK I; BRIGHT 8;AT 0,
1;"P R E S S A N Y K E Y ! ! !
43 LET I=I-1: IF I<0 THEN LET
I=7
44 IF INKEY#="" THEN GO TO 42
45 LET SC=0: CLS : LET d=0: BR
IGHT 1: INK 5
46 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
47 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
48 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
49 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
50 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
51 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
52 PRINT : INK 5: BRIGHT 0
53 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
54 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
55 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
56 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
57 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
58 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
59 PRINT " ■ ■ ■ ■ ■
■ ■ ■ ■ ■
60 PRINT AT 15,6; INK 1;"ENTER
LEVEL (1 TO 3)"

```

```

61 PRINT AT 10,8; INK 1/2;"===
=====
62 PRINT AT 10,8; INK 4; INK 7
;"S-SHIP LEFT S-SHIP RIGHT"
63 PRINT AT 10,11; INK 4;"O-TH
RUST"
64 PRINT AT 20,0; INK 6;"TRY T
O LAND SHIP ON LANDING PADS"
65 PRINT #1; BRIGHT 8; INK 7;"
LUNAR STORM © DAVID PERRY 1983"
66 PRINT AT 15,6; INK 1;"ENTER
LEVEL (1 TO 3)"

```

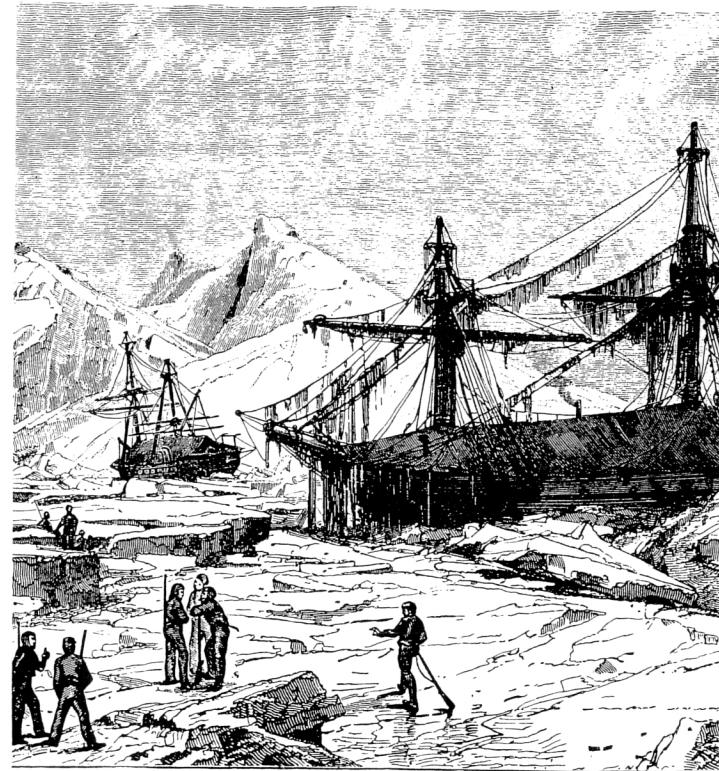
```

67 LET a#=INKEY#: IF a#<"1" OR
a#>"3" THEN GO TO 75
68 IF a#="" THEN GO TO 75
69 IF a#="1" THEN LET a#=""
L
ET d=1
70 IF d=1 THEN LET b#=""
L
71 IF a#="2" THEN LET a#=""
L
ET d=2
72 IF d=2 THEN LET b#=""
L
73 IF a#="3" THEN LET a#=""
L
ET d=3
74 IF d=3 THEN LET b#=""
L
75 GO TO 10
76 LET i=i-1: IF i<0 THEN LET
i=7
77 GO TO 65
78 STOP
79 PRINT AT x,u; INK 7; BRIGHT
8;" " : IF Y=4 THEN LET BO=150:
GO TO 83
80 IF Y=16 THEN LET BO=100: GO
TO 83
81 IF Y=28 THEN LET BO=150: GO
TO 83
82 GO TO 36
83 LET SC=SC+BO: FOR N=1 TO 3:
FOR I=0 TO 7
84 PRINT AT 1,0; INK I;"C O N
G R A T U L A T I O N S !"
85 PRINT AT 3,7;"BONUS POINTS
:";BO;"."

```

```
86 PRINT AT 5,11;"SCORE :";SC
87 NEXT I: NEXT N
88 FOR N=-20 TO 50 STEP 5: BEE
P .005,N: BEEP .001,50-ABS (N):
BEEP .005,N/2: NEXT N
89 CLS : GO TO 10
```

# THE LURE OF THE MAZE

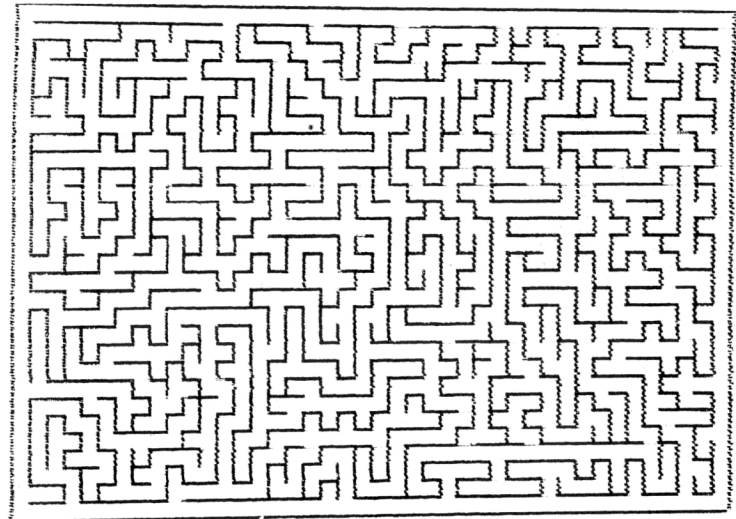


# ABSOLUTELY AMAZING

We now look at a number of different, outstanding maze programs, most of which were written by Graham Charlton.

## Maze-maker

We start with a program which draws a maze of any size, such as this one:



When prompted to do so, enter any number between two and 23 to see the program in action.

```
10 DIM a$(9999)
20 DIM b$(9999)
30 INPUT s
40 LET t=INT (255/s) #s
```

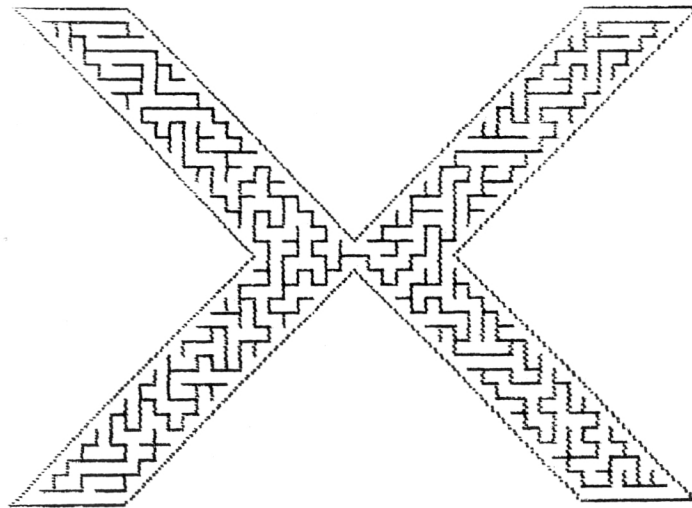
```

45 LET W=INT (175/5) #5
50 PLOT 0,0
60 DRAW t,0
70 DRAW 0,5
80 DRAW -t,0
90 DRAW 0,-5
100 LET X=5
110 LET Y=5
120 LET Z=1
130 PLOT X,Y
140 IF (POINT (X+S,Y)+POINT (X,
-S)+POINT (X,Y+S)+POINT (X-S,Y)
<>4 THEN GO TO 200
150 LET Z=Z-1
160 IF Z=0 THEN STOP
170 LET X=CODEM B#(Z)
180 LET Y=CODEM B#(Z)
190 GO TO 130
200 LET B#(Z)=CHR# X
210 LET B#(Z)=CHR# Y
220 LET Z=Z+1
230 LET T=INT (RND #4)
240 LET C=S*((T=0)-(T=2))
250 LET D=S*((T=0)-(T=3))
260 IF POINT (X+C,Y+D) THEN GO
TO 130
270 DRAW C,D
280 LET X=X+C
290 LET Y=Y+D
300 GO TO 130

```

## Mangled Mazes

There is no reason why the walls of the maze should be at right angles. Here are four MANGLED MAZES programs, which use walls at odd angles:



```

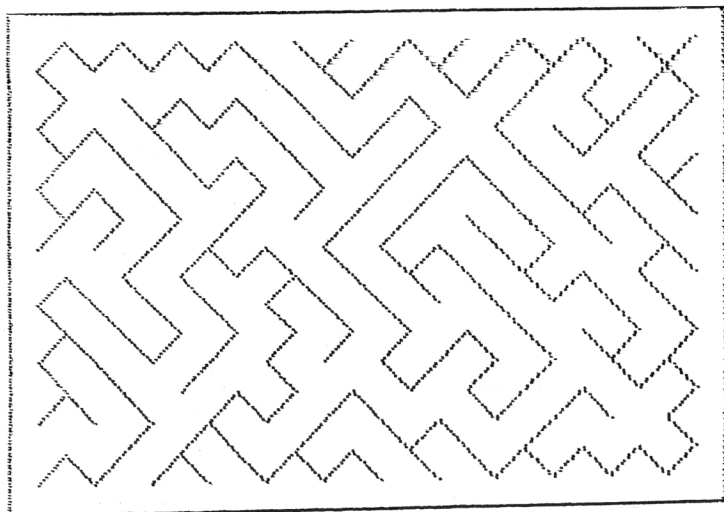
10 DIM B#(10000)
20 DIM S#(10000)
30 INPUT S
40 LET T=INT (254/5) #5
50 LET W=INT (174/5) #5
60 PLOT 0,0
70 DRAW S/W,S/W
72 DRAW -S/W,S/W
74 DRAW (T-1)/W,W
76 DRAW S/W-S,W-S/W
78 DRAW S/W-S,S/W-S
80 DRAW (T-1)/W,W
82 DRAW -S/W,-S/W
84 DRAW -(T-1)/W,W
86 DRAW -S/W+S,S/W-S
88 DRAW -S/W+S,-S/W+S

```

```

000000 DRAW (t-12)/10,0
000000 LET X=100
000000 LET Y=100
000000 LET N=1
000000 PRINT (X,Y)+POINT (X+1,Y)+POINT (X-1,Y)

```



```

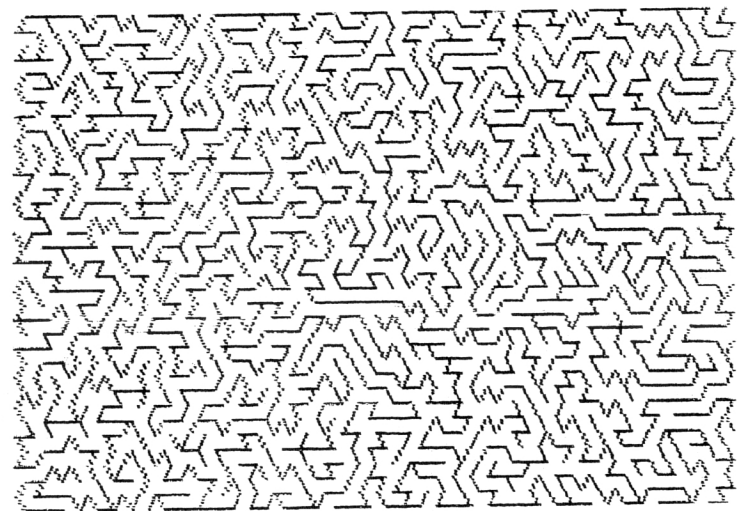
000000 INPUT B#(100000)
000000 INPUT B#(100000)
000000 LET t=1
000000 LET X=100
000000 LET Y=100
000000 PRINT (X,Y)+POINT (X+1,Y)+POINT (X-1,Y)
000000 IF X=100 THEN GO TO 000000
000000 IF Y=100 THEN GO TO 000000
000000 LET N=N+1
000000 LET C=INT (RND*4)
000000 LET D=C*(C<2)-(C>2)
000000 LET D=D*(C=0)+(C=2)-(C=1)-1
000000 IF POINT (X+C,Y+D) THEN GO TO 000000
000000 DRAW C,D
000000 LET X=X+C
000000 LET Y=Y+D
000000 GO TO 100000

```

```

000000 IF N=0 THEN STOP
000000 LET X=100000
000000 LET Y=100000
000000 GO TO 100000
000000 LET B#(N)=CH# X
000000 LET B#(N)=CH# Y
000000 LET N=N+1
000000 LET C=INT (RND*4)
000000 LET D=C*(C<2)-(C>2)
000000 LET D=D*(C=0)+(C=2)-(C=1)-1
000000 IF POINT (X+C,Y+D) THEN GO TO 000000
000000 DRAW C,D
000000 LET X=X+C
000000 LET Y=Y+D
000000 GO TO 100000

```



```

000000 INPUT B#(100000)
000000 INPUT B#(100000)
000000 LET X=100000
000000 LET Y=100000
000000 PRINT (X,Y)+POINT (X+1,Y)+POINT (X-1,Y)
000000 IF X=100000 THEN GO TO 000000
000000 IF Y=100000 THEN GO TO 000000
000000 LET N=N+1
000000 LET C=INT (RND*4)
000000 LET D=C*(C<2)-(C>2)
000000 LET D=D*(C=0)+(C=2)-(C=1)-1
000000 IF POINT (X+C,Y+D) THEN GO TO 000000
000000 DRAW C,D
000000 LET X=X+C
000000 LET Y=Y+D
000000 GO TO 100000

```

```

00 LET q=INT (SOR (t*t-5#5))
100 PLOT X,Y
000 FOR c=-t TO t STEP t*2
010 IF x+c>255 OR x+c<0 THEN GO
TO 230
020 IF POINT (x+c,y)=0 THEN GO
TO 000
030 NEXT c
050 FOR c=-s TO s STEP t
060 IF x+c>255 OR x+c<0 THEN GO
TO 010
070 FOR d=-q TO q STEP q*2
080 IF y+d>175 OR y+d<0 THEN GO
TO 000
090 IF POINT (x+c,y+d)=0 THEN GO
TO 000
0100 NEXT d
0110 NEXT c
0120 LET Z=Z-1
0130 IF Z=0 THEN STOP
0140 LET X=CODE a#(Z)
0150 LET Y=CODE b#(Z)
0160 GO TO 100
0170 LET a#(Z)=CHR# X
0180 LET b#(Z)=CHR# Y
0190 LET Z=Z+1
0200 LET r=INT (RND#6)
0210 LET c=(r=1)+(r=5)-(r=0)-
(r=4)+t*((r=3)-(r=2))
0220 LET d=q*((r<0)-(r>0))
0230 IF x+c>255 OR x+c<0 THEN GO
TO 000
0240 IF y+d>175 OR y+d<0 THEN GO
TO 000
0250 IF POINT (x+c,y+d) THEN GO
TO 000
0260 DRAW c,d
0270 LET X=X+c
0280 LET Y=Y+d
0290 GO TO 100

```



## Walls of Suspense

In this program, you have to race two computer opponents (both shown as "O") in trying to solve a maze. You are the "\*". You start at the top left hand corner and are trying to get to the bottom right hand one.

The first one to solve the maze is the winner. Delete line 3140 to make it a real time game. 3120

```

10 DIM a#(20,20)
110 DIM b#(1000)
120 DIM c#(1000)
000 LET X=10
010 LET Y=17
020 LET N=1
030 FOR M=0 TO 20
040 LET c#(M)="#####"
050 NEXT M
060 LET c#(X,Y)=" "
070 IF c#(X+2,Y)="#" OR c#(X-2,
Y)="#" OR c#(X,Y+2)="#" OR c#(X,
Y-2)="#" THEN GO TO 1000
080 LET Z=Z-1
090 IF Z=0 THEN GO TO 2000
100 LET X=CODE a#(Z)
110 LET Y=CODE b#(Z)
120 GO TO 100
1300 LET a#(Z)=CHR# X
1400 LET b#(Z)=CHR# Y
1500 LET Z=Z+1
1600 LET r=INT (RND#4)
1700 LET c=(r=0)-(r=1)
1800 LET d=(r=2)-(r=3)
1900 IF c#(X+2#C,Y+2#d)="#" THEN
GO TO 1000
2000 LET c#(X+c,Y+d)=" "
2100 LET c#(X+2#C,Y+2#d)="#"
2200 LET X=X+2#C
2300 LET Y=Y+2#d
2400 GO TO 1000
2500 FOR M=0 TO 20

```



```

000010 PRINT INK 7;C#(W,R TO 1
000020 NEXT W
000030 PRINT AT 10,20;"X"
000040 LET X=1
000050 LET Y=1
000060 LET C=1
000070 LET U=1
000080 LET V=1
000090 LET S=0
000100 LET T=0
000110 PRINT AT X,Y;"#"
000120 LET N#="INKEY#"
000130 IF N#="" THEN GO TO 0110
000140 LET C=(N#="5")-(N#="7")
000150 LET U=(N#="0")-(N#="5")
000160 IF SCREEN# (X+C,Y+U)="# THEN
000170 PRINT AT X,Y;"█"
000180 LET X=X+C
000190 LET Y=Y+U
000200 PRINT AT X,Y;"#"
000210 IF X=10 AND Y=10 THEN STOP
000220 LET S=S-1
000230 IF S=-1 THEN LET S=0
000240 PRINT AT P,Q;"█"
000250 IF S=4 THEN LET S=0
000260 LET C=(S=1)-(S=0)
000270 LET U=(S=0)-(S=0)
000280 IF SCREEN# (P+C,Q+U)="# THEN
000290 LET S=S+U: GO TO 00025
000300 LET P=P+C
000310 LET Q=Q+U
000320 PRINT AT P,Q;"█"
000330 IF P=10 AND Q=10 THEN STOP
000340 LET T=T-1
000350 IF T=-4 THEN LET T=0
000360 PRINT AT U,V;"█"
000370 IF T=4 THEN LET T=0
000380 LET C=(T=0)-(T=0)
000390 LET U=(T=1)-(T=0)
000400 IF SCREEN# (U+C,V+U)="# THEN
000410 LET T=T+U: GO TO 00036
000420 LET V=V+U
000430 LET W=W+U
000440 PRINT AT U,V;"█"
000450 IF U=10 AND V=10 THEN STOP
000460 GO TO 0100

```

## Wall of Suspense

In this version of the game, you are only racing a single computer opponent. However, it is far more intelligent than the double opponents and can race as fast as you can.

```

10 DIM C#(20,20)
10 DIM A#(1000)
14 DIM B#(1000)
0000 LET X=10
0000 LET Y=17
0000 LET N=1
0000 FOR W=0 TO 20
0000 LET C#(W)= "#####"
#####
70 NEXT W
100 LET C#(X,Y)=""
110 IF C#(X+10,Y)="#" OR C#(X-10,
Y)="#" OR C#(X,Y+10)="#" OR C#(X,
Y-10)="#" THEN GO TO 1000
1200 LET N=N-1
1300 IF N=0 THEN GO TO 2000
1400 LET X=CODE# A#(N)
1500 LET Y=CODE# B#(N)
1600 GO TO 100
1700 LET A#(N)=CHR# X
1800 LET B#(N)=CHR# Y
1900 LET N=N+1
2000 LET T=INT (RND*4)
2100 LET C=(T=0)-(T=1)
2200 LET U=(T=0)-(T=0)
2300 IF C#(X+10*C,Y+10*U)="" THEN
GO TO 1000
2400 LET C#(X+C,Y+U)=""
2500 LET C#(X+10*C,Y+10*U)=""
2600 LET X=X+10*C
2700 LET Y=Y+10*U
2800 GO TO 100
2900 FOR W=0 TO 20
3000 PRINT INK 7;C#(W,R TO 1)
3100 NEXT W
3200 PRINT AT 10,20;"X"
3300 LET X=1
3400 LET Y=1
3500 LET C=1
3600 LET U=1

```





```

500 LET C=100
510 LET N=1
520 FOR W=0 TO 44
530 LET C#(W)="#####"
#####
#####
540 NEXT W
100 LET C#(X,Y)=" "
110 IF C#(X+2,Y)="#" OR C#(X-2,
Y)="#" OR C#(X,Y+2)="#" OR C#(X,
Y-2)="#" THEN GO TO 1000
120 LET Z=N-1
130 IF Z=0 THEN GO TO 2000
140 LET X=CODE A#(Z)
150 LET Y=CODE B#(Z)
160 GO TO 100
110000 LET A#(Z)=CHR# X
110100 LET B#(Z)=CHR# Y
110200 LET Z=N+1
110300 LET R=INT (RND*4)
110400 LET C=(R=0)-(R=1)
110500 LET D=(R=2)-(R=3)
110600 IF C#(X+2*C,Y+2*D)="#" THEN
GO TO 1000
110700 LET C#(X+C,Y+D)="#"
110800 LET C#(X+2*C,Y+2*D)="#"
110900 LET X=X+2*C
111000 LET Y=Y+2*D
111100 GO TO 100
000000 FOR W=0 TO 99
000100 PRINT C#(W,0 TO 99)
000200 NEXT W
000300 LET X=1
000400 LET Y=1
000500 LET P=0
000600 LET Q=0
000700 PRINT FLASH 1;AT X,Y;"*"
0010 IF P=0 AND Q=0 THEN PRINT
AT 0,0;"X"
0020 LET N#=INKEY#
0030 IF N#="" THEN GO TO 0000
0040 LET C=(N#="6")-(N#="7")
0050 LET D=(N#="8")-(N#="9")
0060 IF SCREEN#(X+C,Y+D)="#" TH
EN GO TO 0000
0070 PRINT AT X,Y;"■"
0080 LET X=X+C
0090 LET Y=Y+D
0100 IF Y=0 THEN GO SUB 5000
0110 IF Y=31 THEN GO SUB 5100
0120 IF X=0 THEN GO SUB 5200

```

```

02150 IF X=21 THEN GO SUB 5000
02160 PRINT FLASH 1;AT X,Y;"*"
02170 IF X=0 AND Y=30 AND P=20 A
ND Q=30 THEN STOP
02180 GO TO 0010
000000 RANDOMIZE USR 50021
000100 LET Y=Y+1
000200 LET Q=Q-1
000300 FOR W=0 TO 21
000400 PRINT AT W,0;C#(P+W,Q)
000500 NEXT W
000600 RETURN
001000 RANDOMIZE USR 50000
001100 LET Y=Y-1
001200 LET Q=Q+1
001300 FOR W=0 TO 21
001400 PRINT AT W,01;C#(P+W,Q+01)
001500 NEXT W
001600 RETURN
002000 RANDOMIZE USR 50111
002100 LET X=X+1
002200 LET P=P-1
002300 PRINT AT 0,0;C#(P,Q TO Q+31)
002400 RETURN
003000 RANDOMIZE USR 50040
003100 LET X=X-1
003200 LET P=P+1
003300 PRINT AT 21,0;C#(P+21,Q TO
Q+31)
003400 RETURN
004000 DATA 00,0,64,05,02,20,0,01
004100 DATA 00,04,40,11,00,05,10,040
004200 DATA 114,05,02,00,040,001
004300 DATA 00,05,07,00,0,100
004400 DATA 0,01,40,04,05,120,40
004500 DATA 16,240,114,40,01,00
004600 DATA 040,001
004700 DATA 00,0,64,17,00,04,00
004800 DATA 010,14,00,0,0,0,110
004900 DATA 101,00,7,054,0,0,0
005000 DATA 101,10,05,10,10,041,10
005100 DATA 40,10,101,00,7,054,0
005200 DATA 40,00,054,7,00,0,10
005300 DATA 17,0,7,05,00,04,00,7
005400 DATA 000,000,00,00,004,000,0
005500 DATA 7,00,000,00,001,004,000,0
005600 DATA 00,005,007,004,100,00
005700 DATA 00,005,07,17,000,07,00
00610 DATA 010,14,20,6,00,05,110

```

```

00000 DATA 101,200,7,00054,1,0,0
00000 DATA 101,100,4,00004,1,0,0
00004 DATA 40,0,1,1001,0,0,4,0
00000 DATA 40,0,4,0004,0,0,0,0
00000 DATA 17,0,7,0007,0,0,0,0
00000 DATA 24,0,15,000,0,0,0,0
00000 DATA 104,0,54,000,0,0,0,0
00000 DATA 000,0,0,0,0,0,0,0
00000 DATA 00,0,0,0,0,0,0,0

```

This is the BASIC version:

```

10 DIM c$(45,65)
20 DIM a$(1000): DIM b$(1000)
40 LET x=43: LET y=63: LET z=1
70 FOR w=2 TO 44
80 LET c$(w)="" #####
#####
##### ": NEXT w
100 LET c$(x,y)=""
110 IF c$(x+2,y)="#" OR c$(x-2,
c)="#" OR c$(x,y+2)="#" OR c$(x,
c-2)="#" THEN GO TO 1000
130 LET z=z-1: IF z=0 THEN GO T
O 2000
140 LET x=CODE a$(z)
150 LET y=CODE b$(z): GO TO 100
1000 LET a$(z)=CHR# x
1010 LET b$(z)=CHR# y: LET z=z+1
1030 LET r=INT (RND*4)
1040 LET c=(r=0)-(r=2)
1050 LET d=(r=0)-(r=3)
1060 IF c$(x+2*c,y+d#2)="" THEN
GO TO 1030
1070 LET c$(x+c,y+d)=""
1080 LET c$(x+2*c,y+d#2)=""
1090 LET x=x+2*c: LET y=y+d#2
1110 GO TO 100
2000 LET x=1: LET y=1: LET p=0
2010 LET q=0: LET c$(40,00)="X"
2020 PRINT AT 0,0:
2100 FOR w=p TO p+21
2110 PRINT c$(w,q TO q+31)
2120 NEXT w
2200 PRINT FLASH 1: AT x,y: "*"
3010 IF x=20 AND y=30 AND p=23 A
ND q=33 THEN STOP
3020 LET z#=INKEY#
3030 IF z#="" THEN GO TO 3020

```

```

00040 LET c=(z#="0")-(z#="7")
00050 LET d=(z#="0")-(z#="0")
00060 IF SOMEFN# (x+c,y+d)="#" TH
EN GO TO 30000
00070 PRINT AT x,y: "*"
00075 LET c$(x+p,y+q)=""
00080 LET x=x+c: LET y=y+d
00100 IF y=0 THEN LET y=y+1: LET
y=y-1: GO TO 30000
00110 IF y=31 THEN LET y=y-1: LET
y=y+1: GO TO 30000
00140 IF x=0 THEN LET x=x+1: LET
x=x-1: GO TO 30000
00150 IF x=21 THEN LET x=x-1: LET
x=x+1: GO TO 30000
00180 GO TO 30000

```

# Rollermaze

This maze program was written by Neil Pellinacci. It was written for the 16K computer. It will run on the 48K machine but may be difficult to modify because the program's machine code will have to be altered.

The aim of the game is to travel across a randomly-generated maze, first from top to bottom and then back to the top again in the shortest possible time. To enliven the game, bits of the maze roll left and the rest rolls right. You, however, don't roll anywhere.

You have three lives and you lose one every time you collide with a chunk of maze to your left or to your right. A siren noise sounds when this happens. The number of lives, together with the time taken and the best time so far, is displayed at the top of the screen.

The title page sequence begins when you start the program. This tells you which keys to press and also the top ten times and the names of the players which achieved them.

Presssing "K" at any time in the sequence will allow you to change the control keys. Just follow the instructions. "S" will leave the title pages and start a game.

ROLLERMAZE has five levels of difficulty. I suggest you start with level one, then look at level five in action to see the difference. The clock starts as soon as the maze appears. If your final time is in the top ten, you'll be asked for your name, which you will then see in the best scores list.

To put the game into your machine, carry out the following, and remembering to VERIFY after SAVE:

1/ Type in program one, the machine-code loader program, and run it. You should get the message '9 STOP statement' if the program is correct. Now save this program on tape as you may need it later.

## PROGRAM 1...Machine Code

```
05 0000 0000 0000 0000 0000 0000 0000
10 0000 0000 0000 0000 0000 0000 0000
20 0000 0000 0000 0000 0000 0000 0000
30 0000 0000 0000 0000 0000 0000 0000
40 0000 0000 0000 0000 0000 0000 0000
50 0000 0000 0000 0000 0000 0000 0000
60 0000 0000 0000 0000 0000 0000 0000
70 0000 0000 0000 0000 0000 0000 0000
80 0000 0000 0000 0000 0000 0000 0000
90 0000 0000 0000 0000 0000 0000 0000
100 0000 0000 0000 0000 0000 0000 0000
110 0000 0000 0000 0000 0000 0000 0000
120 0000 0000 0000 0000 0000 0000 0000
130 0000 0000 0000 0000 0000 0000 0000
140 0000 0000 0000 0000 0000 0000 0000
150 0000 0000 0000 0000 0000 0000 0000
160 0000 0000 0000 0000 0000 0000 0000
170 0000 0000 0000 0000 0000 0000 0000
180 0000 0000 0000 0000 0000 0000 0000
190 0000 0000 0000 0000 0000 0000 0000
200 0000 0000 0000 0000 0000 0000 0000
210 0000 0000 0000 0000 0000 0000 0000
220 0000 0000 0000 0000 0000 0000 0000
230 0000 0000 0000 0000 0000 0000 0000
240 0000 0000 0000 0000 0000 0000 0000
250 0000 0000 0000 0000 0000 0000 0000
260 0000 0000 0000 0000 0000 0000 0000
270 0000 0000 0000 0000 0000 0000 0000
280 0000 0000 0000 0000 0000 0000 0000
290 0000 0000 0000 0000 0000 0000 0000
300 0000 0000 0000 0000 0000 0000 0000
310 0000 0000 0000 0000 0000 0000 0000
320 0000 0000 0000 0000 0000 0000 0000
330 0000 0000 0000 0000 0000 0000 0000
340 0000 0000 0000 0000 0000 0000 0000
350 0000 0000 0000 0000 0000 0000 0000
360 0000 0000 0000 0000 0000 0000 0000
370 0000 0000 0000 0000 0000 0000 0000
380 0000 0000 0000 0000 0000 0000 0000
390 0000 0000 0000 0000 0000 0000 0000
400 0000 0000 0000 0000 0000 0000 0000
410 0000 0000 0000 0000 0000 0000 0000
420 0000 0000 0000 0000 0000 0000 0000
430 0000 0000 0000 0000 0000 0000 0000
440 0000 0000 0000 0000 0000 0000 0000
450 0000 0000 0000 0000 0000 0000 0000
460 0000 0000 0000 0000 0000 0000 0000
470 0000 0000 0000 0000 0000 0000 0000
480 0000 0000 0000 0000 0000 0000 0000
490 0000 0000 0000 0000 0000 0000 0000
500 0000 0000 0000 0000 0000 0000 0000
510 0000 0000 0000 0000 0000 0000 0000
520 0000 0000 0000 0000 0000 0000 0000
530 0000 0000 0000 0000 0000 0000 0000
540 0000 0000 0000 0000 0000 0000 0000
550 0000 0000 0000 0000 0000 0000 0000
560 0000 0000 0000 0000 0000 0000 0000
570 0000 0000 0000 0000 0000 0000 0000
580 0000 0000 0000 0000 0000 0000 0000
590 0000 0000 0000 0000 0000 0000 0000
600 0000 0000 0000 0000 0000 0000 0000
610 0000 0000 0000 0000 0000 0000 0000
620 0000 0000 0000 0000 0000 0000 0000
630 0000 0000 0000 0000 0000 0000 0000
640 0000 0000 0000 0000 0000 0000 0000
650 0000 0000 0000 0000 0000 0000 0000
660 0000 0000 0000 0000 0000 0000 0000
670 0000 0000 0000 0000 0000 0000 0000
680 0000 0000 0000 0000 0000 0000 0000
690 0000 0000 0000 0000 0000 0000 0000
700 0000 0000 0000 0000 0000 0000 0000
710 0000 0000 0000 0000 0000 0000 0000
720 0000 0000 0000 0000 0000 0000 0000
730 0000 0000 0000 0000 0000 0000 0000
740 0000 0000 0000 0000 0000 0000 0000
750 0000 0000 0000 0000 0000 0000 0000
760 0000 0000 0000 0000 0000 0000 0000
770 0000 0000 0000 0000 0000 0000 0000
780 0000 0000 0000 0000 0000 0000 0000
790 0000 0000 0000 0000 0000 0000 0000
800 0000 0000 0000 0000 0000 0000 0000
810 0000 0000 0000 0000 0000 0000 0000
820 0000 0000 0000 0000 0000 0000 0000
830 0000 0000 0000 0000 0000 0000 0000
840 0000 0000 0000 0000 0000 0000 0000
850 0000 0000 0000 0000 0000 0000 0000
860 0000 0000 0000 0000 0000 0000 0000
870 0000 0000 0000 0000 0000 0000 0000
880 0000 0000 0000 0000 0000 0000 0000
890 0000 0000 0000 0000 0000 0000 0000
900 0000 0000 0000 0000 0000 0000 0000
910 0000 0000 0000 0000 0000 0000 0000
920 0000 0000 0000 0000 0000 0000 0000
930 0000 0000 0000 0000 0000 0000 0000
940 0000 0000 0000 0000 0000 0000 0000
950 0000 0000 0000 0000 0000 0000 0000
960 0000 0000 0000 0000 0000 0000 0000
970 0000 0000 0000 0000 0000 0000 0000
980 0000 0000 0000 0000 0000 0000 0000
990 0000 0000 0000 0000 0000 0000 0000
1000 0000 0000 0000 0000 0000 0000 0000
```

```

1000 PRINT AT 1,0; INK 0; PAPER 0
1010 INPUT "TIME: "; T
1020 INPUT "LIVES: "; L
1030 INPUT "BEST: "; B
1040 FOR A=4 TO 20 STEP 2: PRINT
1050 AT A,0; "*****"
1060 FOR A=1 TO LEV: FOR B=3 TO
20 STEP 2
1070 LET B1=INT (RAND#30)+1: PRIN
T AT B,B1; " "
1080 NEXT B: NEXT A
1090 FOR A=1 TO LEV: FOR B=4 TO
20 STEP 2
1100 LET B1=INT (RAND#30)+1: PRIN
T AT B,B1; INK 0; " "
1110 NEXT B: NEXT A
1120 FOR A=3 TO 20: PRINT AT A,0
; OVER 1; INK 0; S$: NEXT A
1130 INK 0: RETURN
1140 GO SUB 600
1150 LET LIVES=L: LET T=T+0
1160 LET FIN=21: LET L=L-2: LET C=
25
2010 PRINT AT 1,6; PAPER 0; INK
6;T;AT L,C; INK 0; PAPER 7;" "

```

2/Type RUN again. When the program stops, type NEW to remove the program, and nothing else, as you still need the machine code.

3/ Type in the main program, program two which follows. Then save it directly after program one. To save it, use SAVE "ROLLERMAZE" - do not use line 9600 in the program.

```

PROGRAM 2...Main Game

1 REM ** ROLLING MAZE **
  Neil Pelligracci

800 CLS : LET Q$="ENTER LEVEL 0
7 DIFFICULTY (1-5)"
810 INPUT "Q$"; LEV
820 IF LEV<1 OR LEV>5 OR LEV<>I
NT LEV THEN LET Q$="ENTER A WHOL
E NUMBER, 1 TO 5": GO TO 810
830 POKE 32551,0: POKE 32555,1:
POKE 32557,1: POKE 32569,1
840 BORDER 2: PAPER 7: CLS
810 PRINT AT 0,0; PAPER 0; INK
7;" "
  THE ROLLING MAZE

820 PRINT AT 1,0; INK 6; PAPER
0;"TIME: "; T
830 PRINT AT 1,0; INK 6; PAPER 0;
" LIVES: "; L
840 PRINT AT 1,0; INK 6; PAPER 0;
" BEST: "; B
850 PRINT AT 1,4; "<"; AT 1,17; "<"; AT 1,27
;"<"
860 PRINT AT 1,29; PAPER 0; INK
6;H(1)
870 PRINT AT 2,0; INK 1; S$; AT 2
1,0; S$
880 GO SUB 6000
890 INK 7
900 FOR A=4 TO 20 STEP 2: PRINT
AT A,0; "*****"
910 NEXT A
920 FOR A=1 TO LEV: FOR B=3 TO
20 STEP 2
930 LET B1=INT (RAND#30)+1: PRIN
T AT B,B1; " "
940 NEXT B: NEXT A
950 FOR A=1 TO LEV: FOR B=4 TO
20 STEP 2
960 LET B1=INT (RAND#30)+1: PRIN
T AT B,B1; INK 0; " "
970 NEXT B: NEXT A
980 FOR A=3 TO 20: PRINT AT A,0
; OVER 1; INK 0; S$: NEXT A
990 INK 0: RETURN
1000 GO SUB 600
1010 LET LIVES=L: LET T=T+0
1020 LET FIN=21: LET L=L-2: LET C=
25
2010 PRINT AT 1,6; PAPER 0; INK
6;T;AT L,C; INK 0; PAPER 7;" "

```



```

2020 LET A=USR 32000: IF A=5 THE
N GO TO 2100
2025 IF A=1 THEN IF L>2 THEN IF
SCREEN# (L-1,C)<>" THEN LET L=L
-1
2030 IF A=2 THEN IF L<21 THEN IF
SCREEN# (L+1,C)<>" THEN LET L=
L+1
2035 LET C=C+(A=4 AND C<31)-(A=3
AND C>0)
2040 IF SCREEN# (L,C)="" THEN LE
T A=USR 32550: GO TO 2100
2050 IF L<>FIN THEN PRINT AT L,C
: INK 1;"*": LET T=T+1: GO TO 20
20
2060 IF FIN=21 THEN LET FIN=2: G
O TO 2010
2070 FOR B=1 TO 2: FOR A=40 TO -
20 STEP -2: BEEP .004,A: BEEP .0
04,A-15: BEEP .004,30-B: BEEP .0
04,30-A-15: NEXT A: NEXT B
2080 GO TO 2000
2100 LET LIVES=LIVES-1: PRINT AT
1,10: PAPER 0: INK 6:LIVES
2110 IF LIVES<>0 THEN GO TO 2000
2120 PRINT AT 6,0: INK 1;S#:AT 1
5,0;S#
2130 PRINT AT 6,11: INK 8:"GAME
OVER":AT 15,11:"GAME OVER"
2140 FOR A=1 TO 256: LET B=USR 3
2300: NEXT A
2150 FOR A=1 TO 150: NEXT A: GO
TO 4000
2200 FOR A=1 TO 10: IF T<H(A) TH
EN GO TO 2230
2210 NEXT A: GO TO 4000
2230 FOR B=0 TO A STEP -1: LET H
(B+1)=H(B): LET H#(B+1)=H#(B): N
EXT B
2240 LET H(A)=T: INPUT "PLEASE T
YPE YOUR NAME:" : LINE N#: LET H#
(A)=0#: LET H#(A, TO LEN N#+1)=N
#
2250 GO SUB 5000: GO TO 4070
4000 BORDER 0: PAPER 0: CLS : RE
M TITLE PAGES
4010 GO SUB 5100: LET INK=6: GO
SUB 4500: REM TITLE
4020 FOR C=1 TO 20: GO SUB 4500:
IF CHR# PEEK 23556="S" OR CHR#
PEEK 23556="K" THEN GO TO 4100
4030 NEXT C

```

```

4040 GO SUB 5000: REM OPTIONS
4050 FOR C=1 TO 20: GO SUB 4500:
IF CHR# PEEK 23556="S" OR CHR#
PEEK 23556="K" THEN GO TO 4100
4060 NEXT C
4070 GO SUB 5500: REM HI-SCORES
4080 FOR A=1 TO 500: IF CHR# PEE
K 23556="S" OR CHR# PEEK 23556="
K" THEN GO TO 4100
4090 NEXT A: GO TO 4000
4100 LET A=USR 32500: IF CHR# PE
EK 23556="S" THEN GO SUB 4200: G
O TO 1500
4110 GO SUB 4200: GO TO 4300
4200 FOR A=1 TO 34: LET B=USR 32
250+USR 32400: NEXT A: RETURN
4300 REM CHANGE KEYS
4310 INK 0: BORDER 6: PAPER 6: C
LS : BORDER 3
4315 POKE 32557,1: POKE 32551,10
: POKE 32555,1: POKE 32565,10
4320 PRINT AT 1,10: INK 6:"REDEF
INE KEYS"
4330 PRINT " INK 1:" PRESS YOU
R NEW KEY WHEN EACH TAB 9:"PROM
PT APPEARS"
4340 PRINT " INK 2;TAB 11: PAP
ER 7:"UP";TAB 21;
4350 LET K=PEEK 23556: IF K=255
THEN GO TO 4350
4360 POKE 32311,K: PRINT PAPER 7
:CHR# K
4365 LET A=USR 32550: FOR A=1 TO
150: NEXT A
4370 PRINT " INK 2;TAB 11: PAPER
7:"DOWN";TAB 21;
4380 LET K=PEEK 23556: IF K=255
THEN GO TO 4380
4390 POKE 32319,K: PRINT PAPER 7
:CHR# K
4395 LET A=USR 32550: FOR A=1 TO
150: NEXT A
4400 PRINT " INK 2;TAB 11: PAPER
7:"LEFT";TAB 21;
4410 LET K=PEEK 23556: IF K=255
THEN GO TO 4410
4420 POKE 32327,K: PRINT PAPER 7
:CHR# K
4425 LET A=USR 32550: FOR A=1 TO
150: NEXT A

```

```

4430 PRINT " INK 2;TAB 11; PAPER
7;"RIGHT";TAB 21;
4440 LET K=PEEK 32556: IF K=255
THEN GO TO 4440
4450 POKE 32335,K: PRINT PAPER 7
:CHR# K
4455 LET A=USR 32550: FOR A=1 TO
150: NEXT A
4460 PRINT " INK 2;TAB 11; PAPER
7;"ABORT";TAB 21;
4470 LET K=PEEK 32556: IF K=255
THEN GO TO 4470
4480 POKE 32343,K: PRINT PAPER 7
:CHR# K
4490 LET A=USR 32550: GO TO 4000
4500 INK INK: RESTORE 4600: FOR
A=0 TO 2: READ A#: PRINT AT A,1;
A#: NEXT A
4510 FOR A=3 TO 5: READ A#: PRIN
T AT A,5;A#: NEXT A
4520 FOR A=7 TO 11: READ A#: PRI
NT AT A,14;A#: NEXT A
4530 PRINT INK 7;AT 5,1;"Written
to";AT 10,4;"New!";AT 12,1;"Pe
l
linacci";
4540 LET INK=INK+1: IF INK=0 THE
N LET INK=1
4550 INK 0: RETURN
4600 DATA " THE "
4610 DATA " "
4620 DATA " "
4630 DATA " "
4640 DATA " "
4650 DATA " "
4660 DATA " "
4670 DATA " MAZE "
4680 DATA " "
4690 DATA " "
4700 DATA " "
4710 DATA " "
5000 REM OPTIONS
5010 FOR A=20 TO 21: PRINT AT A,
0;0#: NEXT A
5020 INK 5: PRINT AT 15,10;"PRES
S";AT 17,10; BRIGHT 1;"S"; BRIG
HT 0;" TO START";AT 19,7; BRIGH
T 1;"K"; BRIGHT 0;" TO CHANGE K
EYS"

```

```

5030 RETURN
5100 INK 6: PRINT AT 15,9;"GAME
CONTROLS";AT 17,6;" - LEFT";AT 17
,22;" - UP";AT 19,6;" - RIGHT";AT
19,21;" - DOWN"
5110 PRINT BRIGHT 1;AT 17,4;CHR#
PEEK 32327;AT 17,20;CHR# PEEK 3
2311;AT 19,4;CHR# PEEK 32335;AT
19,20;CHR# PEEK 32319
5120 PRINT AT 21,11; BRIGHT 1;CH
R# PEEK 32343; BRIGHT 0;" - ABOR
T"
5130 INK 0: RETURN
5500 BORDER 0: PAPER 0: CLS : IN
K 7: PRINT AT 0,6; INK 6; BRIGHT
1;"TODAY'S BEST SCORES"
5510 FOR A=1 TO 10: PRINT " " A
ND A<>10;A;" ";H#(A);". .";H(A);
NEXT A
5515 PRINT #0; INK 5;" P
RESS "; BRIGHT 1;"S"; BRIGHT 0;"
OR "; BRIGHT 1;"K"
5520 RETURN
6000 REM JINGLE
6010 RESTORE 6400
6020 READ A,B: IF A=99 THEN RETU
RN
6025 IF A=88 THEN PAUSE B+5: GO
TO 6020
6030 BEEP A/3,B-3+LEV: GO TO 602
0
6400 DATA .3,4,.3,4,.6,10,.3,4,.
3,4,.6,10,.3,4,.3,4,.9,10.1
6410 DATA 88,19
6420 DATA .4,10,.4,12,.4,5,.4,0,
.3,4.5
6430 DATA 85,20,.5,0,.5,4
6500 DATA 99,0
6600 REM FORM UDG'S
6610 RESTORE 9000: FOR A=USR "A"
TO USR "B"+7: READ B: POKE A,B:
NEXT A
9000 DATA 16,56,16,124,16,40,56,
56,24,126,126,255,255,126,126,24
9050 LET S#=""
9060 DIM H(10): DIM H$(10,20)
9070 FOR A=1 TO 10: LET H$(A)="N
ELL PELLINACCI . . . .": NEXT A
9080 FOR A=1 TO 10: READ B: LET
H(A)=B: NEXT A

```

```

9000 DATA 350,390,430,470,510,55
0,590,630,720,1000
9100 LET D#=".....
*."
9200 GO TO 4000
9500 FOR Z=32550 TO 32560: PRINT
Z;" "PEEK Z: NEXT Z
9510 STOP
9600 SAVE "ROLLERMAZE" LINE 9700
: SAVE "ROLLERCODE"CODE 32000,60
@
9610 STOP
9700 CLEAR 01000: LOAD ""CODE :
GO TO 9000

```

```

User Defined Graphics
A....A          #....B

```

4/ Type RUN 9000. When the title page appears, press "S" to play. If the program crashes, you've made a mistake somewhere. If the program runs properly, try redefining the keys. Test everything.

5/ When you're happy with the program, save your final copy, using RUN 9600. Then check it by using VERIFY "ROLLERMAZE" and then VERIFY "ROLLERCODE"CODE 32000,600

6/ Play the game, either loading it, or by typing RUN 9000

The program, as is pretty obvious, is written in BASIC and machine code. The machine code reads the keyboard, rolls the maze and handles sound and other special effects. Various POKES in the program control these routines. Line 9500 can be used for checking machine code by changing

the addresses. There is enough space in the machine code memory map for your own special effects routines, if you want to add them.

If you own a 48K machine and you want to move the machine code higher up in memory, you must remember to:

i/ Change the USR addresses in the main programs

ii/ Change individual bytes within the machine code, as some routines call others. For those who know about such things, the two address bytes following CALL (205 decimal) will have to be changed

iii/ Make the necessary adjustments to the machine code loader program

# Three-D Maze

This great program was written by Scott Vincent. You need a 48K computer to run it.

A ten by ten maze, with a single entrance/exit is stored in the computer's memory. The object of the game is to wander through the maze, find the treasure and then get back to where you started.

If you lose your sense of direction, you can press the "R" key for a repeat. This takes you through the maze from the beginning, showing you quickly all of your previous moves. When the repeat has finished, you continue on with your trek.

Use the following keys to move around the maze:

- 1 - to move left
- 2 - to move right
- 0 - to go forwards
- L - to turn around

Note that you need to press L twice in order to turn through 180 degrees; you cannot press 1 or 2 twice.

If you manage to get out of the maze with the treasure, you'll be told how many moves you took. The maze is not generated at random, so it is the same from game to

game. However, the treasure is placed in a random location each time you run the program.

```
5 BORDER 5: PAPER 5: INK 1: C
L8
10 RANDOMIZE : DIM M(9): DIM N
(9): DIM M$(201,9): DIM N$(200,1
2)
30 FOR X=0 TO 31: READ Y: POKE
USR "9"+X,Y: NEXT X
30 FOR X=1 TO 9: READ M(X): NE
XT X
40 FOR X=1 TO 9: READ N(X): NE
XT X
50 FOR X=1 TO 200: READ M$(X):
NEXT X
50 FOR X=1 TO 200: READ N$(X):
NEXT X
70 LET Q$="001": LET P$="": LE
T L=1: LET T=0: LET X=1+INT (RN
D*7)
80 LET L3=201*(X=1 OR X=4 OR X
=6 OR X=7)+81*(X=2)+127*(X=3)+16
9*(X=5)
90 LET L6=201*(X=4 OR X=5)+100
*(X=1)+84*(X=2)+122*(X=3)+172*(X
=5)+196*(X=7)
100 LET L7=94*(X=1)+96*(X=2)+12
4*(X=3)+131*(X=4)+117*(X=5)+191*(
X=6)+199*(X=7)
110 CLS : FOR X=1 TO 9: IF N$(L
,X)="0" THEN GO SUB M(X)
120 NEXT X
130 FOR X=1 TO 9: IF M$(L,X)="1
" THEN GO SUB N(X)
140 NEXT X
150 IF L=2 OR L=L3 OR L=L6 OR L
=L7 THEN GO SUB 200: IF L=2 AND
T=1 THEN GO TO 70
160 LET A$=INKEY$: IF A$="" THE
N GO TO 160
170 LET P=VAL N$(L, TO 3)*(A$="
1")+VAL N$(L,4 TO 6)*(A$="0")+VA
L N$(L,7 TO 9)*(A$="2")+VAL N$(L
,10 TO )*(A$="1")
180 IF A$="r" THEN GO TO 300
190 IF P=0 THEN GO TO 160
```

```

200 BEEP .008,0: LET L=P: LET Q
#=#0$+("0" AND L<100)+("0" AND L<
10)+STR$ L
210 LET P$=P$+A$: GO TO 110
220 IF L=2 AND Tr=0 THEN CLS :
PRINT AT 10,12; FLASH 1;"ENTRANC
E": RETURN
230 IF L=2 THEN CLS : PRINT AT
8,4;"Well done, that took you";A
T 10,11;LEN P$;" moves.";AT 14,0
;"press any key for another game
": PAUSE 464: RETURN
240 IF L=L2 THEN PLOT 116,99: D
RAW OVER 1;22,0: PLOT 116,76: DR
AW OVER 1;22,0: RETURN
250 IF L=Lb THEN PLOT 100,115:
DRAW OVER 1;54,0: PLOT 100,60: D
RAW OVER 1;54,0: RETURN
260 IF L=Lc THEN PLOT 72,143: D
RAW OVER 1;110,0: PLOT 72,32: DR
AW OVER 1;110,0
270 IF Tr=1 THEN PRINT AT 9,11;
"you already";AT 10,12;"have the
";AT 11,12;"treasure.": RETURN
280 LET Tr=1: PRINT AT 9,11;"Yo
u have";AT 10,11;"found the";AT
11,11;"treasure."
290 RETURN
300 LET Y=1
310 FOR J=1 TO LEN Q$ STEP 3
320 IF J=1 THEN GO TO 390
330 IF P$(Y)="1" THEN PRINT #1;
AT 0,0;"D"
340 IF P$(Y)="0" THEN PRINT #1;
AT 0,15;"B"
350 IF P$(Y)="2" THEN PRINT #1;
AT 0,25;"C"
360 IF P$(Y)="1" THEN PRINT #1;
AT 1,15;"D"
370 LET Y=Y+1
380 FOR X=1 TO 25 STEP 5: FOR S
=X TO X+4 STEP 1.5: BEEP .004,X:
BEEP .004,X+.5: NEXT S: NEXT X
390 LET L=VAL Q$(J TO J+2)
400 CLS : FOR X=1 TO 9: IF M$(L
,X)="0" THEN GO SUB N(X)
410 NEXT X
420 FOR X=1 TO 9: IF M$(L,X)="1
" THEN GO SUB N(X)
430 NEXT X

```

```

440 IF L=2 OR L=L2 OR L=Lb OR L
=Lc THEN GO SUB 220
450 NEXT J: GO TO 150
5000 PLOT 48,175: DRAW 32,-32: P
LOT 48,0: DRAW 32,32: RETURN
5005 PLOT 24,175: DRAW 0,-175: D
RAW 0,32: DRAW 48,0: DRAW 0,111
DRAW -48,0: RETURN
5010 PLOT 163,143: DRAW 32,32: P
LOT 163,32: DRAW 32,-32: RETURN
5015 PLOT 231,175: DRAW 0,-175:
DRAW 0,32: DRAW -48,0: DRAW 0,11
1: DRAW 48,0: RETURN
5020 PLOT 73,142: DRAW 27,-27: P
LOT 73,33: DRAW 27,27: RETURN
5025 PLOT 73,142: DRAW 11,-11: D
RAW 0,-87: DRAW -11,-11: PLOT 85
,88: DRAW 15,0: DRAW 0,55: DRAW
-15,0: RETURN
5030 PLOT 155,115: DRAW 27,27: P
LOT 152,33: DRAW -27,27: RETURN
5035 PLOT 182,33: DRAW -11,11: D
RAW 0,87: DRAW 11,11: PLOT 170,5
9: DRAW -15,0: DRAW 0,55: DRAW 1
5,0: RETURN
5040 PLOT 101,114: DRAW 19,-19:
PLOT 101,61: DRAW 19,19: RETURN
5045 PLOT 101,114: DRAW 8,-8: DR
AW 0,-37: DRAW -8,-8: PLOT 110,9
4: DRAW 6,0: DRAW 4,-4: DRAW -4,
4: DRAW 0,-23: DRAW -6,0: DRAW 6
,0: DRAW 4,4: RETURN
5050 PLOT 154,114: DRAW -19,-19:
PLOT 154,61: DRAW -19,19: RETUR
N
5055 PLOT 154,114: DRAW -8,-8: D
RAW 0,-37: DRAW 8,-8: PLOT 145,9
4: DRAW -6,0: DRAW -4,-4: DRAW 4
,0: DRAW 0,-23: DRAW 6,0: DRAW -
6,0: DRAW -4,4: RETURN
5060 PLOT 73,143: DRAW 110,0: DR
AW OVER 1;0,-110: DRAW 0,-1: DR
AW -111,0: DRAW OVER 1;0,110: RET
URN
5065 PLOT 101,115: DRAW 54,0: DR
AW OVER 1;0,-54: DRAW 0,-1: DRAW
-55,0: DRAW OVER 1;0,54: RETURN

```

5070 OVER 1: PLOT 117,77: DRAW 3  
3: PLOT 138,98: DRAW -3,-3: PLO  
T 138,77: DRAW -3,3: PLOT 120,95  
: DRAW -4,4: OVER 0: DRAW 23,0:  
DRAW OVER 1;0,-22: DRAW 0,-1: DR  
AW -23,0: DRAW OVER 1;0,23: RETU  
RN

5080 DATA 16,48,96,255,255,96,48  
.16,24,60,126,219,224,24,24  
5090 DATA 8,12,6,255,255,6,12,8,  
24,24,24,24,219,126,60,24  
5000 DATA 5000,5010,290,5020,503  
5010 DATA 5005,5015,5060,5025,50  
05,5055,5045,5055,5070  
5020 DATA "111" "0000000010" "  
101" "000010000" "010101" "01000  
0011" "000010101" "000010111" "  
00011" "000000010" "011" "100000  
000" "111" "000100000" "101" "10  
000" "001" "010101" "101" "100001"  
1" "011" "001" "011" "101"  
5030 DATA "10000000" "011" "010  
011" "000110011" "011" "011" "10  
1" "100011" "101" "000111" "111"  
"011" "111" "000111" "000011"  
100000111" "010000101" "101" "11  
1" "000101" "010101" "101" "1000  
10101" "001" "000100010"  
5040 DATA "011" "010111" "111" "  
011" "100010000" "011" "001" "10  
1" "110011" "111" "011" "000111"  
"001" "001" "110001" "110000111  
"001" "101" "110001" "111" "00  
0010100" "101" "010100101" "0101  
01" "111"  
5050 DATA "100101" "000010101" "  
010000000" "101" "100000010" "10  
0000001" "010100000" "011" "0000  
01" "111" "000000111" "111  
"100111" "000111" "000100111"  
"111" "000000100" "001" "001" "1  
00001" "010011" "001" "000010011  
"010001"  
5060 DATA "010110000" "100000010  
"111" "000101" "000111" "101" "  
10101" "000000111" "101" "011" "  
000000110" "111" "000101" "001"  
"11000000" "100000101" "111" "0  
11" "000001" "100011" "001" "000  
100011"

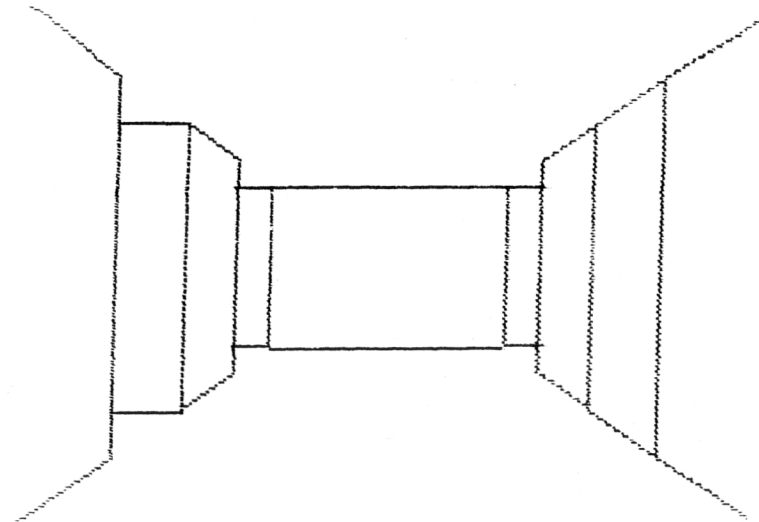
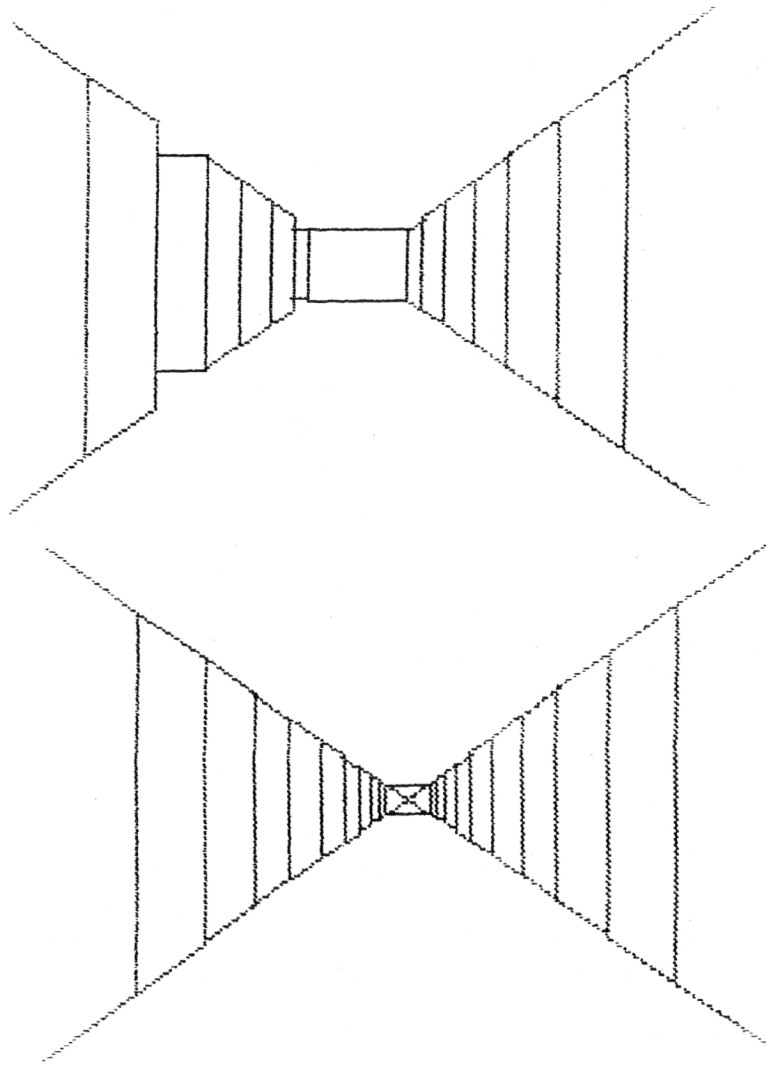
5070 DATA "100011" "0100000001" "  
011" "101" "101" "001" "111" "01  
1" "010101" "011" "101" "101" "1  
01" "0000000110" "011" "000110000  
" "000011" "110000000" "00000011  
14" "100011" "0000000011" "0000001  
10" "1100000000" "000110111" "000  
110000"  
5080 DATA "110111" "110011" "000  
011" "011" "011" "011" "000011"  
"01" "101" "101" "111" "101" "011"  
00100000" "011" "101" "100000100  
" "101" "011" "000010001" "011"  
"101" "010001" "000111" "1101000  
11" "111"  
5090 DATA "000110100" "000000110  
" "000000100" "000010011" "10000  
0000" "010011" "000100000" "0100  
00010" "100000101" "000010000" "  
101" "000101" "000000010" "111"  
"011" "001" "011" "101" "101" "0  
100000000" "000001" "000010000" "  
011" "001" "000011"  
5100 DATA "000000004000" "0000000  
0000001" "00000050000006" "025000000  
00000" "00000070000000" "000000400000  
00000" "00000100000011" "000000000000005  
" "00000500000050" "00000120000013"  
" "0000000000007" "0000000014015"  
5110 DATA "009011000010" "0100000  
0017015" "00000100000012" "015000000  
0014" "0000000000000021" "000000000000  
10" "0000017016010" "0230000000000000  
" "01500100000017" "000000000210000"  
" "00000000000024" "00000000200003"  
5120 DATA "02000000000027" "0000000  
0000004" "00000000000025" "000000000  
90001" "00000500000060" "00000000300  
00" "00000000000070000" "005000000000004  
" "02900010000030" "03300000000000000  
" "00000000000037" "03000000000040"  
5130 DATA "00000000340035" "0420000  
0430041" "00000530000052" "000000700  
00006" "04000390000030" "000004500440  
40" "05400000000055" "0710000072070  
" "00000470000040" "0000043041042"  
" "04000000000050" "0440460000045"



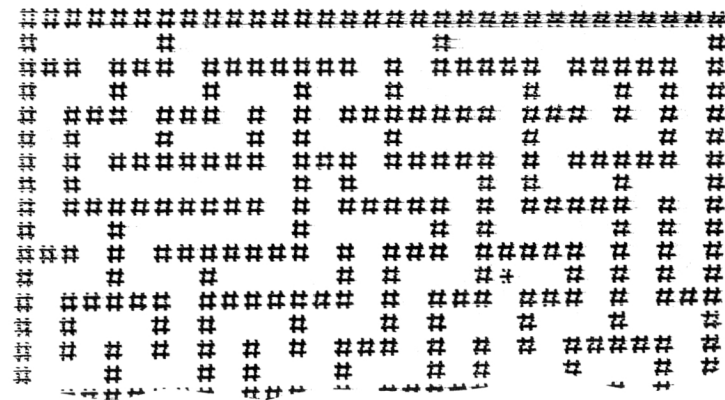




create a full maze. Here are some shots  
of it in action:

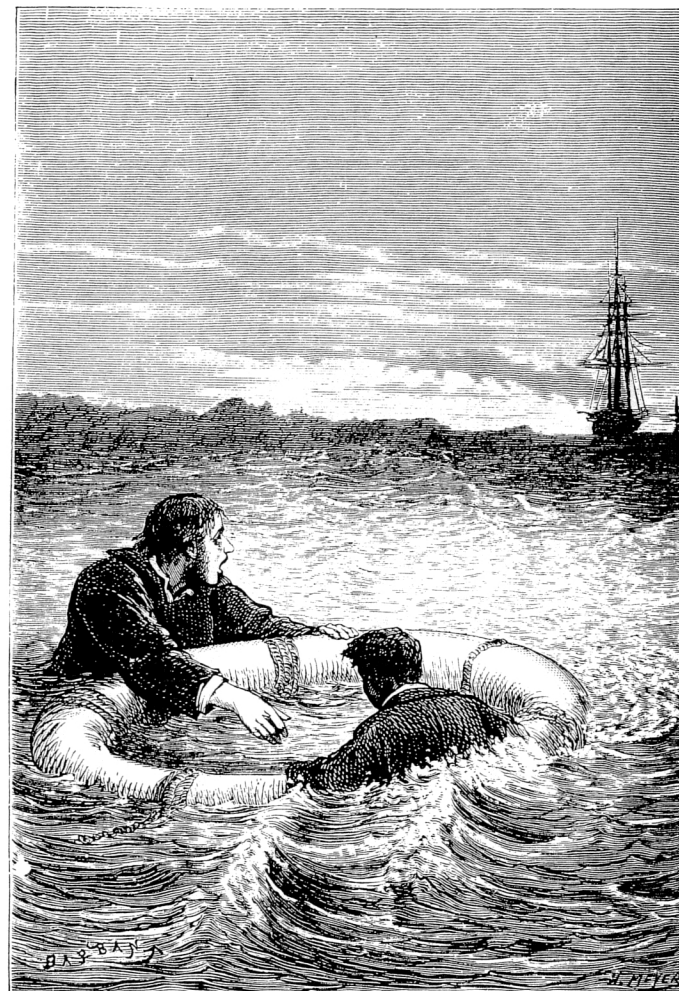


Your object is to move from the top left  
hand corner of the screen to the bottom  
right hand one. You press the "R" to turn  
right, "L" to go left, and "F" to go  
forward. Pressing "H" (for 'help') will  
show you the maze from above, and will  
show you where you are on it. You start  
facing south.





# TWO PLAYER GAMES



## Squares

This program, from Neil Pellinacci, includes versions for either one or two players. In both versions, the game is played on a ten by ten board.

You have to place counters on the board, following the rule that you cannot place a piece on, or adjacent to, one of the computer's pieces.

The one-player game is called 'Area' and you play against the computer. The first player who cannot legally place a piece on the board is the loser. In this game, your move will be rejected if it involves placing a piece on or adjacent to one of the computer's moves.

The computer moves randomly unless it has problems finding a move, in which case it will search methodically. If it still cannot move, it will admit defeat.

The two-player game is called 'Lines' and the winner is the first person to get five squares of their color in a row, horizontally or vertically. A move is rejected if the square is already filled. However, if you move adjacent to (above, below, or to the side of) one of your opponent's squares, the piece becomes one of the opponent's. There are a number of winning techniques which can be used (as

there are in the one-player game). One slight mistake can cost you the game.

It is essential that CAPS LOCK is switched ON during this game. When the program starts, you will see the title page, and four options A to D. If you cannot select an option, check CAPS LOCK before you check your program.

The first two options, one and two player games, are fairly obvious. The third option, to change colors, allows you to redefine both players' colors, by pressing the appropriate key, 1 to 7, when the prompt appears. The two colors cannot be the same.

Asking for information will give you a brief rundown on the aims of the game.

When you play a game, you'll be presented with:

PLAYER ONE NAME ('ENTER' TO SKIP)

Type your name, or if the player's name is the same as before, simply press ENTER. The same goes for player two, when necessary.

The display shows the board when the game is in progress, together with a score display, which shows which player is which color and how many squares of that color

there are on the board. A prompt appears at the bottom right of the screen, indicating whose move it is. Below this is the last move entered, displayed in white.

Enter your move as a number and a letter, eg 7A, 1D in that order. Entering AA will terminate the game, and return you to the options page.

The program is written entirely in BASIC and is made up from several modules which are linked together by the main routine, lines 540 to 740.

```
500 REM LINES
510 REM *****
520 GO SUB 6000: GO SUB 6000
530 GO SUB 6000: GO SUB 6000
540 GO SUB 6000: GO SUB 6000
550 GO SUB 6000: GO SUB 6000
560 GO SUB 6000: GO SUB 6000
570 GO SUB 6000: GO SUB 6000
580 GO SUB 6000: GO SUB 6000
590 GO SUB 6000: GO SUB 6000
600 GO TO 700
610 STOP
620 PRINT "PLAYER ONE"
630 IF AF THEN GO TO 6000
640 GO SUB 6000
650 IF NP=1 THEN GO SUB 6000
660 IF NP=2 THEN GO SUB 6000
670 IF NP=3 THEN GO TO 6000
680 GO TO 700
690 PRINT "PLAYER ONE"
700 PRINT "PLAYER ONE" AT 1,20:PRINT
710 PRINT AT 10,20:INK 0:TO H
720 GO SUB 6000: PRINT AT 10,20
730 IF AF THEN LET AF=1: RETURN
740 IF B(X,Y) THEN GO TO 600
750 IF NP=1 THEN GO TO 6100
760 LET LL=(Y-D)*D+1: LET CC=(X
-D)*D+1
```

```

0040 LET P20=P20+1: LET B(X,Y)=1
0041 PRINT AT LL,CO: OVER 1: PAPER
0042 IF B(X,Y)=0 OR B(X,Y)=1 THEN
0043 IF B(X,Y+1)=0 OR B(X,Y-1)=0 THEN
0044 LET B(X+1,Y)=0 OR B(X-1,Y)=0 THEN
0045 LET B(X,Y)=0: PRINT AT LL,CO:
OVER 1: PAPER 01: AT LL+1,CO
0046 LET P20=P20+1: LET P40=P2
0047
0050 REM P , 0, 40
0051 IF P=1 THEN LET W=0: RETU
0052
0053 LET W=0: GO SUB 7000
0054 RETURN
0055 IF B(X,Y+1)=0 OR B(X,Y-1)=0
OR B(X+1,Y)=0 OR B(X-1,Y)=0 THEN
0056 GO TO 0055
0057 GO TO 0055
0058 REM MOVE
0059 INPUT LINE Q#: IF LEN Q#>0
THEN GO TO 0060
0060 IF Q#="RR" THEN LET W=1: R
RETURN
0061 IF CODE Q#(1) 40 OR CODE Q#
(1) 57 THEN GO TO 0062
0062 IF CODE Q#(2) 40 OR CODE Q#
(2) 74 THEN GO TO 0063
0063 LET X=CODE Q#(1)-40
0064 LET Y=CODE Q#(2)-40
0065 RETURN
0066 REM SPECTRUM
0067 PRINT AT 10,00: INK 0: "PLAY
ER TWO": AT 17,00: Q#: AT 10,00: "TO
MOVE:"
0068 FOR Q=1 TO 10
0069 LET X=INT (RND*10)+0: LET Y
=INT (RND*10)+2
0070 IF B(X,Y) <> 0 THEN GO TO 002
0071
0072 IF B(X+1,Y)=1 OR B(X-1,Y)=1
OR B(X,Y+1)=1 OR B(X,Y-1)=1 THEN
0073 NEXT Q: GO SUB 0000: IF W=1
THEN RETURN
0074 LET LL=(Y-0) # 2 + 1: LET CO=(X
-0) # 2 + 1: PRINT AT LL,CO: OVER 1:
PAPER 01: AT LL+1,CO:
0075 LET B(X,Y)=0: LET P20=P20+1
0076 RETURN
0077 REM SEARCH

```

```

0010 FOR X=0 TO 11: FOR Y=0 TO 1
0011 IF B(X,Y) <> 0 THEN GO TO 0040
0012 IF B(X+1,Y)=1 OR B(X-1,Y)=1
OR B(X,Y+1)=1 OR B(X,Y-1)=1 THEN
0013 GO TO 0040
0014 RETURN
0015 NEXT Y: NEXT X: LET W=1: R
RETURN
0016 REM PLAYER TWO
0017 BEEP 3,-10: PRINT INK 0: AT
10,00: "PLAYER TWO": AT 17,00: Q#
0018 PRINT AT 10,00: INK 0: "TO M
OVE:"
0019 GO SUB 0000: PRINT AT 10,00
0020 IF Q# THEN LET W=0: RETURN
0021 IF B(X,Y) <> 0 THEN GO TO 002
0022
0023 LET LL=(Y-0) # 2 + 1: LET CO=(X
-0) # 2 + 1
0024 LET P20=P20+1: LET B(X,Y)=0
0025 PRINT AT LL,CO: OVER 1: PAPER
0026 IF B(X,Y+1)=1 OR B(X,Y-1)=1
OR B(X+1,Y)=1 OR B(X-1,Y)=1 THEN
0027 LET B(X,Y)=1: PRINT AT LL,CO:
OVER 1: PAPER 01: AT LL+1,CO
0028 LET P20=P20+1: LET P40=P2
0029
0030 BEEP 0,00
0031 LET W=0: GO SUB 7000
0032 RETURN
0033 REM GAME OVER
0034 PRINT 1: PAPER 1: FOR Q=10
TO 21: PRINT AT Q,00:
0035 NEXT Q
0036 PRINT AT 10,00: "GAME OVER"
0037 IF W THEN PRINT AT 10,00: "
BY PLAYER": AT 10,00: "BY PLAYER": AT
17,00: Q#
0038 IF W THEN PRINT AT 10,00: "
PLAYER": W: AT 10,00: "W":
0039 PRINT AT 10,00: "W": INK 0: "PAP
ER ONLY": AT 11,00: "RND:"
0040 BRIGHT 0: PAPER 0: GO SUB 0
0041 FOR K=-20 TO 40 STEP 2: BEEP
0,05: X: BEEP 0,1,00-X: NEXT K
0042 P20=0: GO TO 0040
0043 REM PRINT SCORES

```

```

06910 PRINT AT 0,00; INK 7; "PLAYE
R ONE"; AT 0,00; INK 7; "
PAPER 01;"; "PAPER 0; INK 0; "
06920 PRINT AT 0,00; INK 7; "PLAYE
R TWO"; AT 10,00; INK 7; "
PAPER 02;"; "PAPER 0; INK 0; "
06930 RETURN
06940 RETURN
06950 REM CHECK FOR WIN
06960 FOR X=2 TO 11: FOR Y=2 TO 1
06970 IF B(X,Y) <> 0 THEN LET W=X
06980 IF W=1 THEN RETURN
06990 NEXT Y: NEXT X: RETURN
07000 LET X1=X: LET Y1=Y
07010 FOR Z=1 TO 4: LET X1=X1+1:
07020 IF B(X1,Y1) <> 0 THEN GO TO 7030
07030 NEXT Z: LET W=X: RETURN
07040 LET X1=X: LET Y1=Y
07050 FOR Z=1 TO 4: LET Y1=Y1+1:
07060 IF B(X1,Y1) <> 0 THEN GO TO 7030
07070 NEXT Z: LET W=Y: RETURN
07080 RETURN
07090 RETURN
07100 REM VARIABLES
07110 DIM P#(10): DIM Q#(10): LET
07120 Q1=4: LET Q2=6
07130 DIM B(10,10)
07140 POKE 20000,400
07150 RETURN
07160 REM INFO
07170 BORDER 1: PAPER 1: INK 0: 0
07180 PRINT AT 1,10; "AREA"; AT 2,1
07190 "-----"
07200 PRINT "TAB 8; "A ONE-PLAYER
GAME"; YOU MUST BEAT THE COM
PUTER. YOU MUST PLACE SQUARES ON T
HE BOARD SO THAT NONE ARE
ADJACENT TO THOSE OF THE COM
PUTER."
07210 PRINT "THE COMPUTER WILL
PLAY LIKEWISE" AND WILL TELL Y
OU WHO HAS WON."
07220 PRINT "INK 7;"; "PRESS AN
Y KEY TO CONTINUE"
07230 PAUSE 0: BORDER 0: PAPER 0:
CLS : INK 0

```

```

06970 PRINT AT 1,10; "LINES"; AT 2,
10; "-----"
06980 PRINT "TAB 7; "A TWO-PLAYER
GAME"; YOU MUST BEAT YOUR OP
PONENT BY GETTING THREE SQUARE
S IN A ROW" BEFORE THE COM.
06990 PRINT "THE LINES CAN BE H
ORIZONTAL OR VERTICAL."
07000 PRINT "INK 0;"; "PRESS AN
Y KEY TO CONTINUE"
07010 PAUSE 0: BORDER 4: PAPER 4:
CLS : INK 0
07020 PRINT "TAB 7; "SQUARES CAN
BE PLACED"; "TAB 7; "-----"
07030 PRINT "IF YOU PLACE A SQ
UARE ON THE BOARD ADJACENT T
O ONE OF YOURS OPPONENT'S, IT
WILL IMMEDIATELY BECOME ONE O
F HIS."
07040 PRINT "INK 7;"; "PRESS A
NY KEY TO SELECT GAME"
07050 PAUSE 0: GO TO 06900
07060 REM COO'S
07070 FOR A=0 TO 6: POKE USR "A"+
07080 A, POKE USR "A"+7,0; NEXT A
07090 FOR B=0 TO 6: POKE USR "B"+
07100 B, POKE USR "B"+7,0; NEXT B
07110 FOR C=0 TO 6: POKE USR "C"+
07120 C, POKE USR "C"+7,0; NEXT C
07130 FOR D=0 TO 6: POKE USR "D"+
07140 D, POKE USR "D"+7,0; NEXT D
07150 FOR E=0 TO 6: POKE USR "E"+
07160 E, POKE USR "E"+7,0; NEXT E
07170 FOR F=0 TO 6: POKE USR "F"+
07180 F, POKE USR "F"+7,0; NEXT F
07190 FOR G=0 TO 6: POKE USR "G"+
07200 G, POKE USR "G"+7,0; NEXT G
07210 FOR H=0 TO 6: POKE USR "H"+
07220 H, POKE USR "H"+7,0; NEXT H
07230 FOR I=0 TO 6: POKE USR "I"+
07240 I, POKE USR "I"+7,0; NEXT I
07250 FOR J=0 TO 6: POKE USR "J"+
07260 J, POKE USR "J"+7,0; NEXT J
07270 FOR K=0 TO 6: POKE USR "K"+
07280 K, POKE USR "K"+7,0; NEXT K
07290 FOR L=0 TO 6: POKE USR "L"+
07300 L, POKE USR "L"+7,0; NEXT L
07310 FOR M=0 TO 6: POKE USR "M"+
07320 M, POKE USR "M"+7,0; NEXT M
07330 FOR N=0 TO 6: POKE USR "N"+
07340 N, POKE USR "N"+7,0; NEXT N
07350 FOR O=0 TO 6: POKE USR "O"+
07360 O, POKE USR "O"+7,0; NEXT O
07370 FOR P=0 TO 6: POKE USR "P"+
07380 P, POKE USR "P"+7,0; NEXT P
07390 FOR Q=0 TO 6: POKE USR "Q"+
07400 Q, POKE USR "Q"+7,0; NEXT Q
07410 FOR R=0 TO 6: POKE USR "R"+
07420 R, POKE USR "R"+7,0; NEXT R
07430 FOR S=0 TO 6: POKE USR "S"+
07440 S, POKE USR "S"+7,0; NEXT S
07450 FOR T=0 TO 6: POKE USR "T"+
07460 T, POKE USR "T"+7,0; NEXT T
07470 FOR U=0 TO 6: POKE USR "U"+
07480 U, POKE USR "U"+7,0; NEXT U
07490 FOR V=0 TO 6: POKE USR "V"+
07500 V, POKE USR "V"+7,0; NEXT V
07510 FOR W=0 TO 6: POKE USR "W"+
07520 W, POKE USR "W"+7,0; NEXT W
07530 FOR X=0 TO 6: POKE USR "X"+
07540 X, POKE USR "X"+7,0; NEXT X
07550 FOR Y=0 TO 6: POKE USR "Y"+
07560 Y, POKE USR "Y"+7,0; NEXT Y
07570 FOR Z=0 TO 6: POKE USR "Z"+
07580 Z, POKE USR "Z"+7,0; NEXT Z
07590 FOR AA=0 TO 6: POKE USR "AA"+
07600 AA, POKE USR "AA"+7,0; NEXT AA
07610 FOR AB=0 TO 6: POKE USR "AB"+
07620 AB, POKE USR "AB"+7,0; NEXT AB
07630 FOR AC=0 TO 6: POKE USR "AC"+
07640 AC, POKE USR "AC"+7,0; NEXT AC
07650 FOR AD=0 TO 6: POKE USR "AD"+
07660 AD, POKE USR "AD"+7,0; NEXT AD
07670 FOR AE=0 TO 6: POKE USR "AE"+
07680 AE, POKE USR "AE"+7,0; NEXT AE
07690 FOR AF=0 TO 6: POKE USR "AF"+
07700 AF, POKE USR "AF"+7,0; NEXT AF
07710 FOR AG=0 TO 6: POKE USR "AG"+
07720 AG, POKE USR "AG"+7,0; NEXT AG
07730 FOR AH=0 TO 6: POKE USR "AH"+
07740 AH, POKE USR "AH"+7,0; NEXT AH
07750 FOR AI=0 TO 6: POKE USR "AI"+
07760 AI, POKE USR "AI"+7,0; NEXT AI
07770 FOR AJ=0 TO 6: POKE USR "AJ"+
07780 AJ, POKE USR "AJ"+7,0; NEXT AJ
07790 FOR AK=0 TO 6: POKE USR "AK"+
07800 AK, POKE USR "AK"+7,0; NEXT AK
07810 FOR AL=0 TO 6: POKE USR "AL"+
07820 AL, POKE USR "AL"+7,0; NEXT AL
07830 FOR AM=0 TO 6: POKE USR "AM"+
07840 AM, POKE USR "AM"+7,0; NEXT AM
07850 FOR AN=0 TO 6: POKE USR "AN"+
07860 AN, POKE USR "AN"+7,0; NEXT AN
07870 FOR AO=0 TO 6: POKE USR "AO"+
07880 AO, POKE USR "AO"+7,0; NEXT AO
07890 FOR AP=0 TO 6: POKE USR "AP"+
07900 AP, POKE USR "AP"+7,0; NEXT AP
07910 FOR AQ=0 TO 6: POKE USR "AQ"+
07920 AQ, POKE USR "AQ"+7,0; NEXT AQ
07930 FOR AR=0 TO 6: POKE USR "AR"+
07940 AR, POKE USR "AR"+7,0; NEXT AR
07950 FOR AS=0 TO 6: POKE USR "AS"+
07960 AS, POKE USR "AS"+7,0; NEXT AS
07970 FOR AT=0 TO 6: POKE USR "AT"+
07980 AT, POKE USR "AT"+7,0; NEXT AT
07990 FOR AU=0 TO 6: POKE USR "AU"+
08000 AU, POKE USR "AU"+7,0; NEXT AU
08010 FOR AV=0 TO 6: POKE USR "AV"+
08020 AV, POKE USR "AV"+7,0; NEXT AV
08030 FOR AW=0 TO 6: POKE USR "AW"+
08040 AW, POKE USR "AW"+7,0; NEXT AW
08050 FOR AX=0 TO 6: POKE USR "AX"+
08060 AX, POKE USR "AX"+7,0; NEXT AX
08070 FOR AY=0 TO 6: POKE USR "AY"+
08080 AY, POKE USR "AY"+7,0; NEXT AY
08090 FOR AZ=0 TO 6: POKE USR "AZ"+
08100 AZ, POKE USR "AZ"+7,0; NEXT AZ
08110 FOR BA=0 TO 6: POKE USR "BA"+
08120 BA, POKE USR "BA"+7,0; NEXT BA
08130 FOR BB=0 TO 6: POKE USR "BB"+
08140 BB, POKE USR "BB"+7,0; NEXT BB
08150 FOR BC=0 TO 6: POKE USR "BC"+
08160 BC, POKE USR "BC"+7,0; NEXT BC
08170 FOR BD=0 TO 6: POKE USR "BD"+
08180 BD, POKE USR "BD"+7,0; NEXT BD
08190 FOR BE=0 TO 6: POKE USR "BE"+
08200 BE, POKE USR "BE"+7,0; NEXT BE
08210 FOR BF=0 TO 6: POKE USR "BF"+
08220 BF, POKE USR "BF"+7,0; NEXT BF
08230 FOR BG=0 TO 6: POKE USR "BG"+
08240 BG, POKE USR "BG"+7,0; NEXT BG
08250 FOR BH=0 TO 6: POKE USR "BH"+
08260 BH, POKE USR "BH"+7,0; NEXT BH
08270 FOR BI=0 TO 6: POKE USR "BI"+
08280 BI, POKE USR "BI"+7,0; NEXT BI
08290 FOR BJ=0 TO 6: POKE USR "BJ"+
08300 BJ, POKE USR "BJ"+7,0; NEXT BJ
08310 FOR BK=0 TO 6: POKE USR "BK"+
08320 BK, POKE USR "BK"+7,0; NEXT BK
08330 FOR BL=0 TO 6: POKE USR "BL"+
08340 BL, POKE USR "BL"+7,0; NEXT BL
08350 FOR BM=0 TO 6: POKE USR "BM"+
08360 BM, POKE USR "BM"+7,0; NEXT BM
08370 FOR BN=0 TO 6: POKE USR "BN"+
08380 BN, POKE USR "BN"+7,0; NEXT BN
08390 FOR BO=0 TO 6: POKE USR "BO"+
08400 BO, POKE USR "BO"+7,0; NEXT BO
08410 FOR BP=0 TO 6: POKE USR "BP"+
08420 BP, POKE USR "BP"+7,0; NEXT BP
08430 FOR BQ=0 TO 6: POKE USR "BQ"+
08440 BQ, POKE USR "BQ"+7,0; NEXT BQ
08450 FOR BR=0 TO 6: POKE USR "BR"+
08460 BR, POKE USR "BR"+7,0; NEXT BR
08470 FOR BS=0 TO 6: POKE USR "BS"+
08480 BS, POKE USR "BS"+7,0; NEXT BS
08490 FOR BT=0 TO 6: POKE USR "BT"+
08500 BT, POKE USR "BT"+7,0; NEXT BT
08510 FOR BU=0 TO 6: POKE USR "BU"+
08520 BU, POKE USR "BU"+7,0; NEXT BU
08530 FOR BV=0 TO 6: POKE USR "BV"+
08540 BV, POKE USR "BV"+7,0; NEXT BV
08550 FOR BU=0 TO 6: POKE USR "BU"+
08560 BU, POKE USR "BU"+7,0; NEXT BU
08570 FOR BV=0 TO 6: POKE USR "BV"+
08580 BV, POKE USR "BV"+7,0; NEXT BV
08590 FOR BW=0 TO 6: POKE USR "BW"+
08600 BW, POKE USR "BW"+7,0; NEXT BW
08610 FOR BX=0 TO 6: POKE USR "BX"+
08620 BX, POKE USR "BX"+7,0; NEXT BX
08630 FOR BY=0 TO 6: POKE USR "BY"+
08640 BY, POKE USR "BY"+7,0; NEXT BY
08650 FOR BZ=0 TO 6: POKE USR "BZ"+
08660 BZ, POKE USR "BZ"+7,0; NEXT BZ
08670 FOR CA=0 TO 6: POKE USR "CA"+
08680 CA, POKE USR "CA"+7,0; NEXT CA
08690 FOR CB=0 TO 6: POKE USR "CB"+
08700 CB, POKE USR "CB"+7,0; NEXT CB
08710 FOR CC=0 TO 6: POKE USR "CC"+
08720 CC, POKE USR "CC"+7,0; NEXT CC
08730 FOR CD=0 TO 6: POKE USR "CD"+
08740 CD, POKE USR "CD"+7,0; NEXT CD
08750 FOR CE=0 TO 6: POKE USR "CE"+
08760 CE, POKE USR "CE"+7,0; NEXT CE
08770 FOR CF=0 TO 6: POKE USR "CF"+
08780 CF, POKE USR "CF"+7,0; NEXT CF
08790 FOR CG=0 TO 6: POKE USR "CG"+
08800 CG, POKE USR "CG"+7,0; NEXT CG
08810 FOR CH=0 TO 6: POKE USR "CH"+
08820 CH, POKE USR "CH"+7,0; NEXT CH
08830 FOR CI=0 TO 6: POKE USR "CI"+
08840 CI, POKE USR "CI"+7,0; NEXT CI
08850 FOR CJ=0 TO 6: POKE USR "CJ"+
08860 CJ, POKE USR "CJ"+7,0; NEXT CJ
08870 FOR CK=0 TO 6: POKE USR "CK"+
08880 CK, POKE USR "CK"+7,0; NEXT CK
08890 FOR CL=0 TO 6: POKE USR "CL"+
08900 CL, POKE USR "CL"+7,0; NEXT CL
08910 FOR CM=0 TO 6: POKE USR "CM"+
08920 CM, POKE USR "CM"+7,0; NEXT CM
08930 FOR CN=0 TO 6: POKE USR "CN"+
08940 CN, POKE USR "CN"+7,0; NEXT CN
08950 FOR CO=0 TO 6: POKE USR "CO"+
08960 CO, POKE USR "CO"+7,0; NEXT CO
08970 FOR CP=0 TO 6: POKE USR "CP"+
08980 CP, POKE USR "CP"+7,0; NEXT CP
08990 FOR CQ=0 TO 6: POKE USR "CQ"+
09000 CQ, POKE USR "CQ"+7,0; NEXT CQ
09010 FOR CR=0 TO 6: POKE USR "CR"+
09020 CR, POKE USR "CR"+7,0; NEXT CR
09030 FOR CS=0 TO 6: POKE USR "CS"+
09040 CS, POKE USR "CS"+7,0; NEXT CS
09050 FOR CT=0 TO 6: POKE USR "CT"+
09060 CT, POKE USR "CT"+7,0; NEXT CT
09070 FOR CU=0 TO 6: POKE USR "CU"+
09080 CU, POKE USR "CU"+7,0; NEXT CU
09090 FOR CV=0 TO 6: POKE USR "CV"+
09100 CV, POKE USR "CV"+7,0; NEXT CV
09110 FOR CW=0 TO 6: POKE USR "CW"+
09120 CW, POKE USR "CW"+7,0; NEXT CW
09130 FOR CX=0 TO 6: POKE USR "CX"+
09140 CX, POKE USR "CX"+7,0; NEXT CX
09150 FOR CY=0 TO 6: POKE USR "CY"+
09160 CY, POKE USR "CY"+7,0; NEXT CY
09170 FOR CZ=0 TO 6: POKE USR "CZ"+
09180 CZ, POKE USR "CZ"+7,0; NEXT CZ
09190 FOR DA=0 TO 6: POKE USR "DA"+
09200 DA, POKE USR "DA"+7,0; NEXT DA
09210 FOR DB=0 TO 6: POKE USR "DB"+
09220 DB, POKE USR "DB"+7,0; NEXT DB
09230 FOR DC=0 TO 6: POKE USR "DC"+
09240 DC, POKE USR "DC"+7,0; NEXT DC
09250 FOR DD=0 TO 6: POKE USR "DD"+
09260 DD, POKE USR "DD"+7,0; NEXT DD
09270 FOR DE=0 TO 6: POKE USR "DE"+
09280 DE, POKE USR "DE"+7,0; NEXT DE
09290 FOR DF=0 TO 6: POKE USR "DF"+
09300 DF, POKE USR "DF"+7,0; NEXT DF
09310 FOR DG=0 TO 6: POKE USR "DG"+
09320 DG, POKE USR "DG"+7,0; NEXT DG
09330 FOR DH=0 TO 6: POKE USR "DH"+
09340 DH, POKE USR "DH"+7,0; NEXT DH
09350 FOR DI=0 TO 6: POKE USR "DI"+
09360 DI, POKE USR "DI"+7,0; NEXT DI
09370 FOR DJ=0 TO 6: POKE USR "DJ"+
09380 DJ, POKE USR "DJ"+7,0; NEXT DJ
09390 FOR DK=0 TO 6: POKE USR "DK"+
09400 DK, POKE USR "DK"+7,0; NEXT DK
09410 FOR DL=0 TO 6: POKE USR "DL"+
09420 DL, POKE USR "DL"+7,0; NEXT DL
09430 FOR DM=0 TO 6: POKE USR "DM"+
09440 DM, POKE USR "DM"+7,0; NEXT DM
09450 FOR DN=0 TO 6: POKE USR "DN"+
09460 DN, POKE USR "DN"+7,0; NEXT DN
09470 FOR DO=0 TO 6: POKE USR "DO"+
09480 DO, POKE USR "DO"+7,0; NEXT DO
09490 FOR DP=0 TO 6: POKE USR "DP"+
09500 DP, POKE USR "DP"+7,0; NEXT DP
09510 FOR DQ=0 TO 6: POKE USR "DQ"+
09520 DQ, POKE USR "DQ"+7,0; NEXT DQ
09530 FOR DR=0 TO 6: POKE USR "DR"+
09540 DR, POKE USR "DR"+7,0; NEXT DR
09550 FOR DS=0 TO 6: POKE USR "DS"+
09560 DS, POKE USR "DS"+7,0; NEXT DS
09570 FOR DT=0 TO 6: POKE USR "DT"+
09580 DT, POKE USR "DT"+7,0; NEXT DT
09590 FOR DU=0 TO 6: POKE USR "DU"+
09600 DU, POKE USR "DU"+7,0; NEXT DU
09610 FOR DV=0 TO 6: POKE USR "DV"+
09620 DV, POKE USR "DV"+7,0; NEXT DV
09630 FOR DW=0 TO 6: POKE USR "DW"+
09640 DW, POKE USR "DW"+7,0; NEXT DW
09650 FOR DX=0 TO 6: POKE USR "DX"+
09660 DX, POKE USR "DX"+7,0; NEXT DX
09670 FOR DY=0 TO 6: POKE USR "DY"+
09680 DY, POKE USR "DY"+7,0; NEXT DY
09690 FOR DZ=0 TO 6: POKE USR "DZ"+
09700 DZ, POKE USR "DZ"+7,0; NEXT DZ
09710 FOR EA=0 TO 6: POKE USR "EA"+
09720 EA, POKE USR "EA"+7,0; NEXT EA
09730 FOR EB=0 TO 6: POKE USR "EB"+
09740 EB, POKE USR "EB"+7,0; NEXT EB
09750 FOR EC=0 TO 6: POKE USR "EC"+
09760 EC, POKE USR "EC"+7,0; NEXT EC
09770 FOR ED=0 TO 6: POKE USR "ED"+
09780 ED, POKE USR "ED"+7,0; NEXT ED
09790 FOR EE=0 TO 6: POKE USR "EE"+
09800 EE, POKE USR "EE"+7,0; NEXT EE
09810 FOR EF=0 TO 6: POKE USR "EF"+
09820 EF, POKE USR "EF"+7,0; NEXT EF
09830 FOR EG=0 TO 6: POKE USR "EG"+
09840 EG, POKE USR "EG"+7,0; NEXT EG
09850 FOR EH=0 TO 6: POKE USR "EH"+
09860 EH, POKE USR "EH"+7,0; NEXT EH
09870 FOR EI=0 TO 6: POKE USR "EI"+
09880 EI, POKE USR "EI"+7,0; NEXT EI
09890 FOR EJ=0 TO 6: POKE USR "EJ"+
09900 EJ, POKE USR "EJ"+7,0; NEXT EJ
09910 FOR EK=0 TO 6: POKE USR "EK"+
09920 EK, POKE USR "EK"+7,0; NEXT EK
09930 FOR EL=0 TO 6: POKE USR "EL"+
09940 EL, POKE USR "EL"+7,0; NEXT EL
09950 FOR EM=0 TO 6: POKE USR "EM"+
09960 EM, POKE USR "EM"+7,0; NEXT EM
09970 FOR EN=0 TO 6: POKE USR "EN"+
09980 EN, POKE USR "EN"+7,0; NEXT EN
09990 FOR EO=0 TO 6: POKE USR "EO"+
10000 EO, POKE USR "EO"+7,0; NEXT EO

```



```

0000 PRINT AT 0,0;" "AT 1,0;" "
0001 AT 1,01;" "AT 0,01;" "
0002 RETURN
0003 REM OPTIONS
0004 LET DT=0: LET WF=0: LET P10
0005 LET P05=0: DIM B(10,10)
0006 BORDER 0: PAPER 2: INK 7: C
0007
0008 PRINT AT 1,0;"###SPECTRUM###"
0009 AT 0,0;"WRITTEN BY KEIL PELLIN"
0010 GOTO 0000
0011 PRINT "TAB 10;"OPTIONS"
0012 PRINT "TAB 0;"(0) ONE PLAY
0013 GOTO 0010
0014 PRINT "TAB 0;"(0) TWO PLAYE
0015 GOTO 0010
0016 PRINT "TAB 0;"(0) CHANGE CO
0017 GOTO 0010
0018 PRINT "TAB 0;"(0) INFORMATI
0019 GOTO 0010
0020 PRINT " " CHOOSE YOUR OP
0021 THON YOU BY " " PRESSING THE AP
0022 PROPRATE KEY"
0023 IF INKEY#="0" THEN LET NP=1
0024 GOTO 0700
0025 IF INKEY#="0" THEN LET NP=2
0026 GOTO 0000
0027 IF INKEY#="0" THEN GO TO 00
0028
0029 IF INKEY#<>"0" THEN GO TO 0
0030
0031 BORDER 7: PAPER 7: CLS . IN
0032 PRINT " "PLAYER ONE COLOUR
0033 " "PRESS THE KEY FOR THE COLOU
0034 " "YOU WANT NOW"
0035 LET A=CODE INKEY#
0036 IF A<40 OR A>50 THEN GO TO
0037
0038 LET C1=A-40: BEEP .5,C1
0039 PRINT " "PLAYER TWO COLOURS
0040 " "PRESS THE KEY FOR THE COLO
0041 " "YOU WANT NOW"
0042 LET A=CODE INKEY#: IF A<40
0043 OR A>50 THEN GO TO 0000
0044 LET C2=A-40: IF C2=C1 THEN
0045 GO TO 0000
0046 BEEP .5,00: GO TO 0000
0047 INPUT "TYPE YOUR NAME I'ENT
0048 TO SKIP" : LINE B#
0049 IF B#="" THEN GO TO 0700
0050 LET P#=#

```

```

0700 LET B#="SPECTRUM": RETURN
0000 INPUT "PLAYER ONE NAME I'ENT
0001 TO SKIP" : LINE B#
0002 IF B#="" THEN GO TO 0000
0003 LET P#=#
0004 INPUT "PLAYER TWO NAME I'ENT
0005 TO SKIP" : LINE B#
0006 IF B#="" THEN GO TO 0000
0007 LET B#=#
0008 RETURN

```

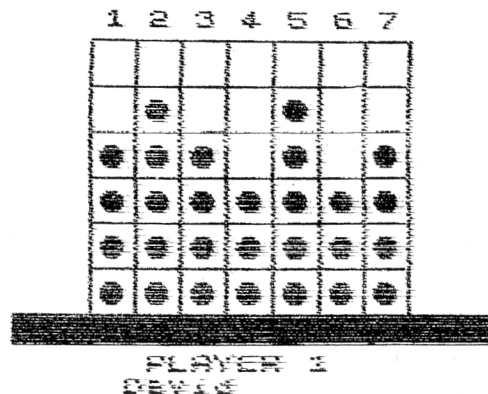
User Defined Graphics



# Four in a Row

From the fertile mind of David Perry comes this version of FOUR IN A ROW. The program contains full instructions and has a hires load.

This is the program in action:



And here's the Listing for it:

```

2 FOR n=0 TO 7: READ a: POKE
USR "a"+n,a: NEXT n: GO SUB 5000
: DIM r(7,7)
: DATA 00,100,200,200,200,200
,120,00
10 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS
20 PRINT AT 3,7;"1 2 3 4 5 6 7"

40 FOR n=52 TO 166 STEP 16
50 PLOT n,40
60 DRAW n,100
80 NEXT n

```

```

90 FOR n=44 TO 142 STEP 16
100 PLOT n,n: DRAW 110,n
110 NEXT n
120 FOR n=04 TO 44: PLOT 25,n:
DRAW 100,n: NEXT n
130 LET play=1
140 BRIGHT 0
200 LET play=1: PRINT AT 10,0:
INK 7: PAPER 0: FLASH 1:" PLAYER
1 ": PRINT AT 10,0;B#;"
210 INPUT "MOVE PLAYER 1 ? ";m0
220 IF m0>7 OR m0<1 THEN GO TO
210
230 GO SUB 1000
250 LET play=2: PRINT AT 10,0:
INK 7: PAPER 0: FLASH 1:" PLAYER
2 ": PRINT AT 10,0;B#;"
260 INPUT "MOVE PLAYER 2 ? ";m0
270 IF m0>7 OR m0<1 THEN GO TO
210
280 GO SUB 1000
290 GO TO 200
1000 GO TO 1000
1000 FOR n=0 TO 1 STEP -1: IF r(
n,m0)=1 THEN NEXT n: RETURN
1070 LET r(n,m0)=1
1080 IF play=1 THEN PRINT AT 3+(
n*2),(m0*2)+5: INK 2: BRIGHT 1;"
●"
1081 IF play=2 THEN PRINT AT 3+(
n*2),(m0*2)+5: INK 5: BRIGHT 1;"
●"
1090 RETURN
5000 BORDER 1: CLS : PLOT 0,0: D
RAW 0,175: DRAW 255,0: DRAW 0,-1
75: DRAW -255,0
5010 PRINT AT 2,3: BRIGHT 1;"FO
UR IN A ROW!"
5020 PRINT AT 3,3:"
5030 PRINT AT 4,1;"In the game o
f "; FLASH 1: PAPER 2: INK 7;"FO
UR IN A ROW!"
5040 PRINT TAB 2;"you must try t
o defeat your "; PRINT TAB 1;"op
ponent by connecting four of"; P
RINT TAB 1;"your counters in a r
ow either"; PRINT TAB 1;"VERTICA
LLY,HORIZONTALLY or"; PRINT TAB
1;"DIAGONALLY... eg."

```

```

5050 PLOT 0,0: DRAW 0,175
5060 PRINT AT 10,4; "●●●●"; PRINT
  AT 10,11; "●"; AT 11,11; "●"; AT 12
  ,11; "●"; AT 13,11; "●"
5070 PRINT AT 10,16; "●"; AT 11,16
  ; "●"; AT 12,17; "●"; AT 10,18; "●"
5080 PRINT AT 10,26; "●"; AT 11,25
  ; "●"; AT 12,24; "●"; AT 13,23; "●"
5090 INPUT "Player 1's name?"; A
$: IF LEN A$ > 10 THEN GO TO 5090
5091 PRINT AT 15,1; INK 7; PAPER
  2; FLASH 1; "PLAYER 1:"; A$
5100 INPUT "Player 2's name?"; B
$: IF LEN B$ > 10 THEN GO TO 5100
5101 PRINT AT 16,1; INK 7; PAPER
  2; FLASH 1; "PLAYER 2:"; B$
5110 PRINT AT 18,1; "PLAYER 1=";
  INK 2; "●"
5120 PRINT AT 19,1; "PLAYER 2=";
  INK 5; "●"
5130 PRINT AT 20,1; FLASH 1; INK
  2; "PRESS ENTER TO COMMENCE!!!"
": PAUSE 0: RETURN

```

## Tanx

This is a two-person game, in which you have to try and shoot the turret of your opponent's tank.

Here are the controls for player one:

Left - Z  
 Right - X  
 Up - 2nd row/left  
 Down - 3rd row/left  
 Fire - Top row/left

And these are the ones which player two should use:

Left - M  
 Right - Symbol Shift  
 Up - 2nd row/right  
 Down - 3rd row/right  
 Fire - Top row/right

You'll notice that the tanks disappear for a moment when a shell is fired, but they can still be hit, and can still move out of the way. You must not run over any of the mines dropped by the shells. You have three lives each and the first to die is, naturally enough, the loser. TANX was written by Neal Cavalier-Smith.

```

10 REM TANX
15 REM Neal Cavalier-Smith
20 LET X1=INT (RND*20): LET Y1
=5: LET X2=INT (RND*20): LET Y2=
20

```

```

30 LET Y1B=1: LET X1B=0: LET Y
2B=-1: LET X2B=0
40 LET CC=0: LET C1=0: LET C1F
60=0: LET C1F=0
50 GO SUB 1000
60 GO SUB 100
70 GO TO 60
100 REM XXXXXXXXXX
101 PRINT AT X1,Y1:" "
102 IF X1B=0 THEN PRINT AT X1,Y
1+Y1B:" "
103 IF Y1B=0 THEN PRINT AT X1+X
1B,Y1:" "
110 IF IN 65022<255 AND X1<20 T
HEN LET X1=X1+1: LET X1B=1: LET
Y1B=0
120 IF IN 64510<255 AND X1>0 TH
EN LET X1=X1-1: LET X1B=-1: LET
Y1B=0
130 IF IN 65278=253 AND Y1>0 TH
EN LET Y1=Y1-1: LET Y1B=-1: LET
X1B=0
140 IF IN 65278=251 AND Y1<30 T
HEN LET Y1=Y1+1: LET Y1B=1: LET
X1B=0
150 PRINT AT X2,Y2:" "
152 IF X2B=0 THEN PRINT AT X2,Y
2+Y2B:" "
153 IF Y2B=0 THEN PRINT AT X2+X
2B,Y2:" "
160 IF IN 49150<255 AND X2<20 T
HEN LET X2=X2+1: LET X2B=1: LET
Y2B=0
170 IF IN 57342<255 AND X2>0 TH
EN LET X2=X2-1: LET X2B=-1: LET
Y2B=0
180 IF IN 32766=251 AND Y2>0 TH
EN LET Y2=Y2-1: LET Y2B=-1: LET
X2B=0
190 IF IN 32766=253 AND Y2<30 T
HEN LET Y2=Y2+1: LET Y2B=1: LET
X2B=0
200 REM XXXXXXXXXX
201 IF IN 61438<255 THEN LET CC
=0
202 IF CC>0 THEN GO SUB 300
210 IF SCREEN# (X2+X2B,Y2+Y2B)=
"." THEN GO TO 2200: PRINT AT X1
,Y1:CHR# 144
211 PRINT AT X1,Y1:CHR# 144
212 IF IN 63486<255 THEN LET C1
=0

```

```

213 IF C1>0 THEN GO SUB 400
214 IF SCREEN# (X1+X1B,Y1+Y1B)=
"." THEN GO TO 2200: PRINT AT X2
,Y2:CHR# 144
220 IF X1B=0 THEN PRINT AT X1,Y
1+Y1B:CHR# 146
230 IF Y1B=0 THEN PRINT AT X1+X
1B,Y1:CHR# 145
240 IF SCREEN# (X1+X1B,Y1+Y1B)=
"." THEN GO TO 2200: PRINT AT X1
,Y1:CHR# 144
240 PRINT AT X2,Y2:CHR# 144
250 IF X2B=0 THEN PRINT AT X2,Y
2+Y2B:CHR# 146
260 IF Y2B=0 THEN PRINT AT X2+X
2B,Y2:CHR# 145
270 RETURN
300 REM XXXXXXXXXX
310 IF CC=0 THEN LET X2S=X2+X2B
: LET Y2S=Y2+Y2B
311 FOR C=0 TO 9
312 IF C=0 THEN GO TO 335
313 IF Y2S<1 THEN LET Y2B=1: LE
T X2B=INT (RND*3)-1
314 IF Y2S>29 THEN LET Y2B=-1:
LET X2B=INT (RND*3)-1
317 IF X2S>19 THEN LET X2B=-1:
LET Y2B=INT (RND*3)-1
318 IF X2S>18 THEN LET X2B=-1:
LET Y2B=INT (RND*3)-1
319 IF X2S<1 THEN LET X2B=1: LE
T Y2B=INT (RND*3)-1
320 IF X=X1 AND Y=Y1 OR X=X1+X1
B AND Y=Y1+Y1B THEN GO SUB 2200
330 PRINT AT X,Y:" "
335 PRINT AT X2S,Y2S:" "
340 LET X=X2S: LET Y=Y2S
350 LET X2S=X2S+X2B
360 LET Y2S=Y2S+Y2B
370 NEXT C
375 LET CC=CC-1
380 RETURN
400 REM XXXXXXXXXX
410 IF C1=0 THEN LET X1S=X1+X1B
: LET Y1S=Y1+Y1B
411 FOR D=0 TO 9
412 IF D=0 THEN GO TO 435
415 IF Y1S<1 THEN LET Y1B=1: LE
T X1B=INT (RND*3)-1
416 IF Y1S>29 THEN LET Y1B=-1:
LET X1B=INT (RND*3)-1

```

```

417 IF X18>18 THEN LET X18=-1:
LET Y18=INT (RND*3) -1
418 IF X18>18 THEN LET X18=-1:
LET Y18=INT (RND*3) -1
419 IF X18<1 THEN LET X18=1: LE
T Y18=INT (RND*3) -1
420 IF X=X2 AND Y=Y2 OR X=X2+X2
& AND Y=Y2+Y2 THEN GO SUB 2000
430 PRINT AT X,Y: " "
435 PRINT AT X18,Y18: "."
436 LET X=X18: LET Y=Y18
440 LET X18=X18+X18
450 LET Y18=Y18+Y18
460 NEXT Q
470 LET C1=C1-1
480 RETURN
2000 STOP
10000 REM DATA 00,125,255,255,255
10010 FOR Z=144 TO 146
10020 FOR X=0 TO 7
10030 READ Q: POKE USR (CHR# Z)+X
,Q
1040 NEXT X
1050 NEXT M
1060 DATA 00,125,255,255,255,
,125,60,24,24,24,24,24,24,24,
0,0,0,255,255,0,0,0
1080 RETURN
10900 PRINT AT X2,Y2: FLASH 1;"S
"
10910 FOR Z=5 TO 100: NEXT Z
10920 PRINT AT X2,Y2: FLASH 0;"
"
10910 LET Life2=Life2-1
10920 IF Life2=0 THEN PRINT INVER
SE 1: FLASH 1;"G A M E O V E R
Player 1 wins
": STOP
10930 GO TO 100
10940 PRINT AT X1,Y1: FLASH 1;"S
"
10950 FOR X=5 TO 100: NEXT X
10960 PRINT AT X1,Y1: FLASH 0;"
"
10910 LET Life1=Life1-1
10920 IF Life1=0 THEN PRINT INVER
SE 1: FLASH 1;"G A M E O V E R
Player 2 wins
": STOP
10930 GO TO 100

```

# MACHINE CODE UTILITIES



# Memory Monitor

This program by David Perry allows you to examine the contents of ROM and RAM. It has good hex/dec and dec/hex routines, and allows you to move, save and load memory.

THIS PROGRAMME WILL REVEAL THE CONTENTS OF YOUR ROM & RAM IN THE FORM OF ADDRESS, ADDRESS IN HEXADECIMAL, DECIMAL VALUE AT THAT ADDRESS, HEXADECIMAL VALUE AT THAT ADDRESS & THE CHARACTER CORRESPONDING TO THE VALUE AT THE ADDRESS.  
THE PROGRAMME WILL ALSO ALLOW YOU TO FREELY MOVE MEMORY AROUND, CONVERT HEXADECIMAL TO DECIMAL AND VISA VERSA AND ALSO TO RUN A ROUTINE FROM ANY ADDRESS

ADDR	HEX	DEC	HEXPEEK	CHR#
0	0000	242	00FD	NEXT
1	0001	175	00FF	CODE
2	0002	17	0011	?
3	0003	255	00FF	COPY
4	0004	255	00FF	COPY
5	0005	195	00C3	NOT
6	0006	203	00CB	THEN
7	0007	17	0011	?
8	0008	42	0022	*
9	0009	93	005D	L
10	000A	92	005C	/
11	000B	34	0022	:
12	000C	95	005F	
13	000D	92	005C	/
14	000E	24	0018	3
15	000F	67	0043	C
16	0010	195	00C3	NOT
17	0011	242	00FD	PAUSE

A: NEW ADDRESS H: HEX-DEC  
E: EXECUTE CODE D: DEC-HEX  
S-SAVE / L-LOAD BYTES.  
M-MOVE MEMORY C-CONTINUE

```

1 REM
2 REM
3 REM MEMORY MONITOR
4 REM © DAVID PERRY!
5 REM 1983
6 REM
7 REM
20 BRIGHT 0: GO SUB 720: INPUT
"START ADDRESS ? ";ADD
30 GO TO 330
40 REM HEX-DEC
50 PRINT AT 20,0;"

60 PRINT AT 20,0;
70 LET A=0: LET E=0
80 INPUT "Hex. Characters ? ";
A$: LET O$="0"
90 LET Q=0
100 IF LEN A$ > 3 THEN GO TO 120
110 LET A$=O$+A$: GO TO 100
120 LET U=4096: LET A=E
130 FOR R=1 TO 4
140 IF FN P(A$(R)) < E OR FN P(A$
(R)) > 15 THEN GO TO 100
150 LET A=A+U*FN P(A$(R)): LET
U=U/16
160 NEXT R
170 PRINT " Hexadecimal: ";A
180 PRINT " Decimal: ";A;"
190 PAUSE 0: RETURN
200 REM DEC-HEX
210 PRINT AT 20,0;"

220 PRINT AT 20,0;
230 INPUT "Decimal Number ? ";A
240 IF A > 65535 OR A < 0 THEN GO T
O 230
250 PRINT AT 20,6;" Decimal: ";A
260 PRINT AT 21,5;" Hexadecim
al: ";
270 LET U=4096: LET Y=A
280 LET T=INT(Y/U)
290 PRINT PAPER 7; BRIGHT 1; CHR
$(148+T+7*(T>9));
300 LET Y=Y-U*T
310 LET U=U/16
320 IF U > 1 THEN GO TO 270
310 RETURN

```

```

320 REM MAIN LAYOUT
330 PRINT AT 0,0; PAPER 5; INK
0;" ADDR+HEX: DEC: HEXPEEK: CHR$
:
340 LET GO=ADD: LET TO=ADD+17:
FOR L=GO TO TO
350 LET DEC=PEEK ADD: LET C$=CH
R$(DEC AND DEC>32)
360 PRINT TAB (1 AND ADD<10)+(1
AND ADD<100)+(1 AND ADD<1000)+(
1 AND ADD<10000); INK 7; PAPER 4
;ADD; PAPER 7;"
370 LET A=ADD: GO SUB 260
380 PRINT " "; PAPER 5;DEC; PA
PER 7;TAB 17; LET A=DEC: GO SUB
260; PRINT PAPER 7;TAB 24; PAPE
R 6;C$
390 LET ADD=ADD+1: IF ADD > 65535
THEN LET ADD=0
400 NEXT L
410 PRINT AT 19,2; PAPER 2; INK
7;" A:NEW ADDRESS H:HEX-DEC "
420 PRINT AT 20,2; PAPER 2; INK
6;" E:EXECUTE CODE D:DEC-HEX "
430 PRINT AT 21,0; INK 7;" ";
PAPER 2;" S-SAVE / L-LOAD BYTES
"; PAPER 7;"
440 PRINT #1;" M-MOVE MEMORY
C-CONTINUE "
450 LET I$=INKEY$: IF I$="" THE
N GO TO 450
460 IF I$="A" OR I$="a" THEN RU
N
470 IF I$="h" OR I$="H" THEN GO
SUB 40: GO TO 410
480 IF I$="s" OR I$="S" THEN GO
SUB 920
490 IF I$="d" OR I$="D" THEN GO
SUB 200: PAUSE 0 : GO TO 410
500 IF I$="e" OR I$="E" THEN GO
TO 570
510 IF I$="l" OR I$="L" THEN GO
SUB 880: GO TO 410
520 IF I$="c" OR I$="C" THEN CL
S : GO TO 330
530 IF I$="M" OR I$="m" THEN GO
TO 600
540 IF I$ <> "S" OR I$ <> "s" OR I$
<> "L" OR I$ <> "l" OR I$ <> "h" OR I
$ <> "H" OR I$ <> "d" OR I$ <> "D" OR
I$ <> "e" OR I$ <> "E" OR I$ <> "c" OR
I$ <> "C" OR I$ <> "M" OR I$ <> "m" TH
EN GO TO 450

```



```

550 GO TO 410
560 REM RUN ROUTINE
570 INPUT "Execute address ? ";
EX
580 RANDOMIZE USA EX: PAUSE 0:
RUN
590 REM MOVE MEMORY
600 INPUT "Move From Address ?
";MF
610 IF MF>65535 OR MF<0 THEN GO
TO 600
620 INPUT "Move to Address ? ";
MT
630 IF MT>65535 OR MT<0 THEN GO
TO 620
640 INPUT "Number of bytes to m
ove ";N
650 IF MT+N>65535 THEN PRINT #1
; "
700 MANY BYTES TRY AGAIN ";
PAUSE 0: GO TO 600
660 LET ADD=MT
670 FOR P=1 TO N
680 POKE MT,PEEK MF
690 LET MT=MT+1: LET MF=MF+1: N
EXT P
700 CLS
710 GO TO 330
715 REM INSTRUCTIONS
720 DEF FN P(A$)=(CODE A$-48-7*
(A$>"9")-32*(A$>"Z"))
730 BORDER 7: PAPER 7: INK 0: C
LS : PRINT AT 1,6;"
740 PRINT AT 2,6;" MEMORY MONI
TOR
750 PRINT AT 3,6;" @ DAVID PER
RY!
760 PRINT AT 4,6;" 1983
770 PRINT AT 5,6;"
780 PRINT AT 8,0;"THIS PROGRAMM
E WILL REVEAL THE": PRINT "COTEN
TS OF YOUR ROM & RAM IN"
790 PRINT "THE FORM OF ADDRESS,
ADDRESS IN": PRINT "HEXADECIMAL
, DECIMAL VALUE AT"
800 PRINT "THAT ADDRESS,HEXADEC
IMAL VALUE": PRINT "AT THAT ADDR
ESS & THE CHARACTER": PRINT "COR
RESPONDING TO THE VALUE AT"
810 PRINT "THE ADDRESS."

```

```

820 PRINT "THE PROGRAMME WILL A
LSD ALLOW": PRINT "YOU TO FREELY
MOVE MEMORY AROUND"
830 PRINT " , CONVERT HEXADECIMAL
TO DECIMAL": PRINT "AND VISA VE
RSA AND ALSO TO RUN": PRINT "A R
OUTINE FROM ANY ADDRESS"
840 REM TAPE LOADING ROUTINE
850 INPUT "DO YOU WANT TO LOAD
A PROGRAMME FROM TAPE (Y OR N) ?
";T$
860 IF T$="Y" OR T$="y" THEN GO
TO 880
870 CLS : RETURN
880 INPUT "Address to load byte
s to?";t
890 IF t>65535 OR t<0 THEN GO T
O 880
900 PRINT FLASH 1; PAPER 2; INK
7;AT 21,0;" PRESS PLAY ON TAPE-
RECORDER!
910 LOAD ""CODE t: CLS : RETURN
920 REM SAVE MEMORY
930 INPUT "SAVE FROM ADDRESS ?
";SF
940 IF SF<0 OR SF>65535 THEN GO
TO 930
950 INPUT "NUMBER OF BYTES ? ";
N
960 IF N<1 OR N>65535 THEN GO T
O 950
970 INPUT "SAVE NAME ? ";S$
980 IF S$="" OR LEN S$>10 THEN
GO TO 970
990 SAVE S$CODE SF,N
1000 RETURN

```

## MC Screensave

This routine, from David Perry, is written for the 48K machine. It allows you to store and print up screens instantly. It can be used in any program.

```
10 BORDER 0: PAPER 0: INK 7: C
LS
15 RESTORE
16 FOR n=40000 TO 40028
20 READ a: POKE n,a
30 NEXT n
40 DATA 33,0,64,17,48,117,1,0,
27,237,176,201,33,48,117,17,0,64
,1,0,27,237,176,201,195,144,234,
0,0
50 PRINT AT 0,0;"LOAD SCREEN F
ROM TAPE AND I WILL": PRINT "SAV
E IT INTO MEMORY FOR INSTANT": P
RINT "RECALL AT ANY TIME!!!"
60 PRINT : PRINT "TO SAVE SCRE
EN TYPE"
70 PRINT "RANDOMIZE USR 40000"
80 PRINT : PRINT "TO PRINT UP
A SCREEN TYPE"
90 PRINT "RANDOMIZE USR 40012"
100 PRINT : PRINT "HERE
IS AN EXAMPLE."
120 PRINT FLASH 1;"PRESS PLAY O
N TAPE"
130 LOAD ""SCREEN$: RANDOMIZE
USR 40000
140 PAUSE 0: CLS : PRINT AT 0,0
;" PRESS A KEY TO PRINT UP SCREE
N": PAUSE 0: RANDOMIZE USR 40012
: GO TO 140
```

## BASIC Screensave

This is a BASIC version of the preceding program, also intended for the 48K computer. It was also written by David Perry.

```
1 CLS : PRINT "THIS PROGRAMME
WILL STORE THE": PRINT "SCREEN
IN MEMORY AND THEN PRINT IT BACK
. AS AN EXAMPLE I WILL": PRINT "
LIST THIS PROGRAMME,SAVE IT AND
THEN PRINT IT BACK ONTO THE": PR
INT "SCREEN."
2 PRINT : PRINT "THE PROGRAMM
E IS TOTALLY BASIC": PRINT "AND
CAN BE USED FOR MANY": PRINT "DI
FFERENT PURPOSES."
3 PRINT : PRINT "PRESS ANY KE
Y TO START": PAUSE 0: CLS
10 REM
20 REM BASIC SCREENSAVE
30 REM (c) DAVID PERRY
40 REM
45 CLS : LIST 50
50 LET A=30000: REM ADDRESS TO
SAVE TO
60 FOR X=0 TO 21
70 FOR Y=0 TO 31
80 LET A#=SCREEN$(X,Y)
85 PRINT AT X,Y:" "
90 POKE A,CODE A#
100 LET A=A+1
110 NEXT Y
120 NEXT X
130 REM
140 REM ****PRINT UP****
150 REM
160 LET A=30000: REM ADDRESS TO
LOAD FROM
170 FOR X=0 TO 21
180 FOR Y=0 TO 31
190 LET B=PEEK A
200 PRINT AT X,Y;CHR$ B
210 LET A=A+1
220 NEXT Y
230 NEXT X
```

# Ireland

To show your computer's graphics off at their best, run this program by David Perry. The awesome task of entering all that data will be rewarded by the quality of the final picture. You can incorporate this program with the SCREENSAVE one (either BASIC or machine code) so you can get Ireland up on your screen instantly.

```
10 BORDER 1: PAPER 1: INK 7: B
RIGHT 8: CLS
11 CLS
21 PLOT 123,135
31 FOR f=1 TO 166
41 READ a,b,c
51 DRAW a,b,c
61 NEXT f
67 PLOT 109,129
71 FOR f=1 TO 7
81 READ a,b,c
91 DRAW a,b,c
101 NEXT f
105 PLOT 55,50
110 FOR f=1 TO 7
120 READ a,b,c
130 DRAW a,b,c
140 NEXT f
145 PLOT 56,52
150 FOR f=1 TO 6
160 READ a,b,c
170 DRAW a,b,c
180 NEXT f
185 PLOT 68,57
190 FOR f=1 TO 5
200 READ a,b,c
210 DRAW a,b,c
220 NEXT f
225 PLOT 92,90
230 FOR f=1 TO 6
240 READ a,b,c
250 DRAW a,b,c
260 NEXT f
```

```
265 PLOT 93,93: DRAW 0,2,0: DRA
W -1,2,0: DRAW -2,-2,0: DRAW -2,
-3,0: DRAW 4,1,0
270 PLOT 80,125: DRAW 0,3,-PI/2
: DRAW -2,2,0: DRAW -3,1,0: DRAW
-3,-2,PI/3: DRAW 6,-3,-PI/3
290 PLOT 85,115: DRAW 3,-1,0: D
RAW -3,4,PI/2: DRAW -1,3,PI/2: D
RAW -3,0,-PI: DRAW 0,-2,0: DRAW
2,-4,-PI/2
1000 DATA 3,5,0,-2,3,-PI/2,-4,6,
0,-1,5,0,-7,7,PI/2,-8,0,PI/2,-3,
-2,-PI
1010 DATA -5,0,0,-4,-5,0,-3,-1,0
,1,4,0,7,5,0,-5,0,PI/4,-3,5,0,-3
,-2,-PI,-2,-5,(PI/1.5)
1020 DATA -1,-4,0,-2,0,0,0,1,0,2
,3,0,-6,7,0,-3,-2,0,-4,-2,PI/1.5
1030 DATA -5,-2,0,-1,-5,-1,0,-2,
-7,PI/2,-3,-5,PI/1.5,-4,-3,PI/2,
0,-6,0,11,-1,0,-3,0,-10,-5,0,
0,-2,0,3,-2,0,-1,-2,0,-4,PI/4
1040 DATA -3,2,0,-3,-5,PI/4,-1,5
,0,-5,0,PI/2,-5,0,0,-1,-4,PI,-3,
-10,PI/4
1050 DATA -2,-5,-PI
1060 DATA 3,-10,PI,4,-1,0,0,-10,PI
,-7,-1,0,-1,-5,0,-5,2,0,-4,PI
,2,-4,PI,4,0,PI/1.5,5,-10,PI*1.2
1070 DATA 2,-1,PI/2,5,5,-5,PI/1.5,
9,1,0,-1,-3,0,-5,-1,0,-4,-7,PI/6
1080 DATA 0,-3,PI,-1,5,0,-1,5,
PI/6,-4,-5,0,1,-2,PI,7,0,0,0,7
I/4,1,-1,0,-5,0,0,-5,-5,-PI/3
1090 DATA -1,1,0,-1,-3,0,1,-2,0,
2,-4,PI/4,-4,-1,0,-5,0,PI,-4,0,7
I/4,-4,-2,0,0,-5,PI/2,7,0,-PI/4,
7,1,0,1,-1,0,-3,-2,PI/10,-9,-2,0
,-2,-5,PI/4,3,-2,PI
1100 DATA 2,0,0,3,-2,PI/1.5,4,1,
PI/4,6,2,0,-3,-4,0,-5,-5,0,1,-3,
PI,9,3,0,2,1,0,3,0,0,-7,-2,0,2,-
3,PI,5,2,0,2,-3,0,4,-2,-PI/6,4,0
,0,2,2,0,4,-1,0,0,0,5,1,PI/1.5
1110 DATA 4,0,PI/3,5,3,0,0,3,PI,
0,3,PI/1.5,4,0,0,1,-5,PI/4,2,0,0,
,1,2,0,4,2,0,0,1,1,0,4,0,0,3
,2,0,-1,3,PI/2,4,2,0,5,1,PI/4,4,
1,PI/1.5,1,1,0,4,0,PI,4,-1,0,2,0
,0,7,0,PI/4
```

```

1120 DATA -1,3,PI/5,-2,3,PI/4,-3
,1,0,4,0,0,3,5,0,2,7,PI/4,2,6,PI
/6,3,6,0,-1,3,PI/2,0,11,PI/5
1130 DATA -3,3,PI/4,-2,0,0,3,1,0
,2,-1,0,0,3,PI,-1,0,0,5,PI/1.7
,-2,5,0,0,3,PI/3,0,3,PI/3,-1,0,0
,0,3,-PI,3,3,PI/4,4,2,PI/6,3,5,0
,0,1,0
1140 DATA 4,0,0,2,2,PI/3,-1,4,PI
,0,7,PI/6,0,1,PI,2,-9,0,2,0,PI,-
1,10,PI/6,-2,3,PI/2,-2,1,0
1150 DATA -5,-3,PI/4,0,2,0
1160 DATA 1,3,PI/6,2,3,PI/4,-3,0
,PI/1.5,-3,1,-PI/1.5,-3,-8,-PI/3
,1,-1,0,4,-1,PI/5
1170 DATA 3,4,0,5,6,-PI/4,-1,1,0
,-2,-1,0,-2,-3,-PI/3,-3,-4,-PI/3
,0,-3,-PI/2,0
1180 DATA 3,2,0,2,-1,0,5,0,0,-3,
1,PI/6,-4,3,-PI/1.75,-3,-4,PI/3
1190 DATA 2,3,0,-2,5,PI/2,0,-2,0
,0,-3,PI/2,0,-3,PI/2,0
1200 DATA 4,-3,PI/6,4,-4,PI/6,1,
0,0,0,3,0,-4,5,PI/6,-5,-1,0

```

## Master Copier

This handy program from David Perry will copy machine code programs automatically.

If there is a BASIC loader, then you must copy it first, by typing MERGE "" then SAVE"name" LINE the line which loads the machine code part. This program reads the header, loads the program and then tells you how to save it. Just follow the prompts and you'll have no trouble using this program.

```

5 GO SUB 31
6 CLEAR 32747
7 FOR I=32748 TO 32761
8 READ X: POKE I,X
9 DATA 55,62,0,221,33,0,125,1
7,20,0,205,86,5,201
10 NEXT I
11 PRINT AT 5,2; FLASH 1;"I AM
NOW LOADING THE HEADER!"; FLASH
0; AT 7,10;"PLEASE WAIT"
12 RANDOMIZE USR 32748
13 LET U=1: GO SUB 19: LET U=0
14 CLS : PRINT AT 0,10;"Header
Data"
15 IF PEEK 32000=0 THEN GO TO
64
16 IF PEEK 32000=1 THEN GO TO
69
17 IF PEEK 32000=2 THEN GO TO
73
18 IF PEEK 32000<>3 THEN RUN
19 LET Z$="": FOR B=32001 TO 3
2010
20 LET Z$=Z$+CHR$(PEEK B)
21 NEXT B
22 LET C$=Z$
23 LET A=PEEK 32011: LET B=PEE
K 32012
24 LET N=A+(B*256)
25 LET A=PEEK 32013: LET B=PEE
K 32014

```

```

26 LET M=A+(B*255)
27 LET A=PEEK 32015: LET B=PEE
K 32016
28 LET O=A+(B*255)
29 IF U=1 THEN RETURN
30 GO TO 47
31 BRIGHT 1: BORDER 0: PAPER 0
: INK 7: CLS : PRINT "      MA
STER COPIER"
32 PRINT "      =====
="
33 PRINT
34 PRINT "THIS PROGRAMME WILL
COPY MACHINE"
35 PRINT "CODE PROGRAMMES AUTO
MATICALLY."
36 PRINT
37 PRINT "IF THERE IS A BASIC
LOADER THEN YOU MUST COPY IT FIR
ST"
38 PRINT "BY TYPING MERGE""""
THEN"
39 PRINT "SAVE""name"" LINE th
e line which loads the machie c
ode part"

40 PRINT : PRINT "THIS PROGRAM
ME READS THE HEADER"
41 PRINT "LOADS THE PROGRAMME
AND THEN"
42 PRINT "TELLS YOU HOW TO SAV
E IT."
43 PRINT : PRINT "1....INSERT
CASSETTE AT START OFMACHINE CODE
SECTION"
44 PRINT "2....CHECK LEADS"
45 PRINT "3....PRESS ANY KEY U
HEN READY"
46 PAUSE 0: CLS : RETURN
47 CLS : PRINT AT 1,0;"MASTER
COPIER"
48 PRINT AT 2,0;"=====

49 PRINT AT 3,0;"To save the p
rogramme now on": PRINT "cassett
e you must first of all"
50 PRINT "have saved the basic
loader and": PRINT "you must re
wind the cassette to": PRINT "th
e start of the MACHINE CODE"

```

```

51 PRINT "section then 'NEW' t
his": PRINT "programme and type"
52 PRINT : PRINT "LOAD """:z$;
""CODE "";#;"";n
53 PRINT "CHECK LEADS,PRESS A
KEY,PRESS": PRINT "PLAY ON THE T
APE RECORDER""
54 PRINT : PRINT "When it has
loaded type:"
55 PRINT : PRINT "SAVE "";z$;"
CODE "";#;"";n
56 PRINT "INSERT BLANK CASSETT
E,CHECK": PRINT "LEADS,PRESS REC
ORD & PRESS A KEY"
57 INPUT "Copy to the printer
(y/n)";a$
58 IF a$="y" THEN COPY
59 IF a$="n" THEN PRINT AT 21,
0; BRIGHT 0;"YOU SHOULD WRITE TH
IS DOWN THEN!"
60 INPUT "Are there any more m
achine code sections to copy (y/
n)";a$
61 IF a$="y" THEN RUN
62 PRINT AT 21,2; FLASH 1; PAP
ER 2;"PRESS ANY KEY TO 'NEW'!!!!

63 PAUSE 0: PAUSE 0: NEW
64 CLS : PRINT "You have just
loaded a BASIC": PRINT "header"
65 PRINT "IT IS CALLED"
66 PRINT z$;" LINE ";#
67 PRINT "ITS BASIC LENGTH IS
";0
68 PRINT : PRINT "Press any ke
y to run programme": PRINT "agai
n...": PAUSE 0: RUN
69 CLS : PRINT "You have just
loaded a NUMERIC": PRINT " ARRAY
header"
70 PRINT "IT IS CALLED"
71 PRINT z$
72 PRINT : PRINT "Press any ke
y to run programme": PRINT "agai
n...": PAUSE 0: RUN
73 CLS : PRINT "You have just
loaded a CHARACTER": PRINT " ARR
AY header"
74 PRINT "IT IS CALLED"
75 PRINT z$

```

```
76 PRINT : PRINT "Press any ke  
u to run programme": PRINT "agai  
n...": PAUSE 0: RUN  
MASTER COPIER  
=====
```

THIS PROGRAMME WILL COPY MACHINE  
CODE PROGRAMMES AUTOMATICALLY.

IF THERE IS A BASIC LOADER THEN  
YOU MUST COPY IT FIRST  
BY TYPING MERGE"" THEN  
SAVE"name" LINE the line which  
loads the machine code part

THIS PROGRAMME READS THE HEADER  
LOADS THE PROGRAMME AND THEN  
TELLS YOU HOW TO SAVE IT.

- 1....INSERT CASSETTE AT START OF  
MACHINE CODE SECTION
- 2....CHECK LEADS
- 3....PRESS ANY KEY WHEN READY

# UTILITIES AND DEMONSTRATIONS

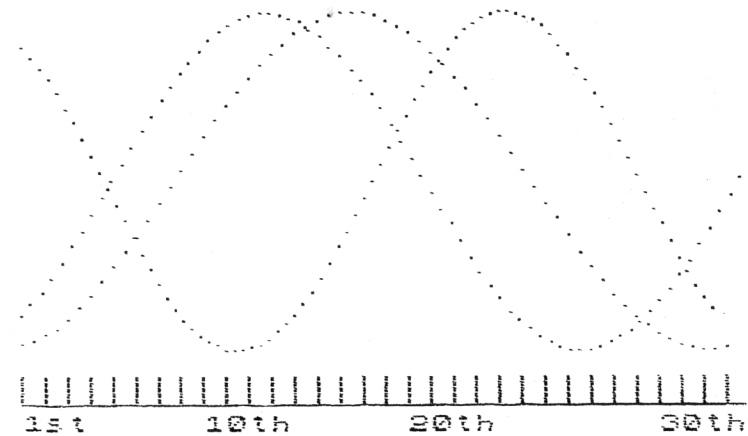


# Biorhythms

Now you can chart your ups and downs in this graphical BIORHYTHMS program.

There are three cycles, which begin at birth and continue right through our lives, which are believed by many to have

physical mental emotional



an effect on our lives. The three cycles are as follows:

PHYSICAL - this is a 23-day cycle which controls such things as endurance and strength

EMOTIONAL - the 28-day emotional cycle governs feelings of optimism and pessimism



MENTAL - Logic and reasoning come under the sway of this 33-day cycle

The program is self-prompting and produces a very effective display.

```

1 RESTORE
2 PRINT AT 0,0;" " BIO
-RHYTHM
10 INPUT "Enter Date Of Birth"
,"Day ";a;" Month ";b;" Year
";c
20 INPUT "Enter Date Now ";;
Month ";d;" Year ";e
25 CLS
30 LET t=INT (((e-c)*365.25)+(
(d-b)*30.35)-a)
300 FOR r=0 TO 255
310 PLOT r,10
315 IF r=INT (r/b)*b THEN FOR u
=10 TO 20: PLOT r,u: NEXT u
320 NEXT r
330 PRINT AT 21,0;"1st 10th
20th 30th"
340 PRINT AT 0,0; INK 1;"physic
al"; INK 2;"mental "; INK 4;"
rational"
350 FOR r=1 TO 3
360 READ u
315 LET l=2*PI*(t-(INT (t/u)*u)
)/u
1000 LET k=2*PI*(33-u)*.03
1000 FOR a=l TO k+(l+(2*PI) STEP
.1
1010 PLOT INK ((1 AND u=23)+(2 AND
u=28)+(4 AND u=33));(a-l)*(35
-20+u),90+5IN a#80
1020 NEXT a
1030 NEXT r
1040 DATA 23,28,33
1050 INPUT "Another Go ? ";a#: I
F a#(1)="y" THEN GO TO 1

```

## Paint Pot

This clever program from Graham Charlton fills a randomly generated shape with the current INK color. Lines 10 to 60 draw the shape, and the subroutine from 9000 onwards fills it. Make the strings (A\$ and B\$) as long as possible on your computer. Note that they take up a lot of memory.

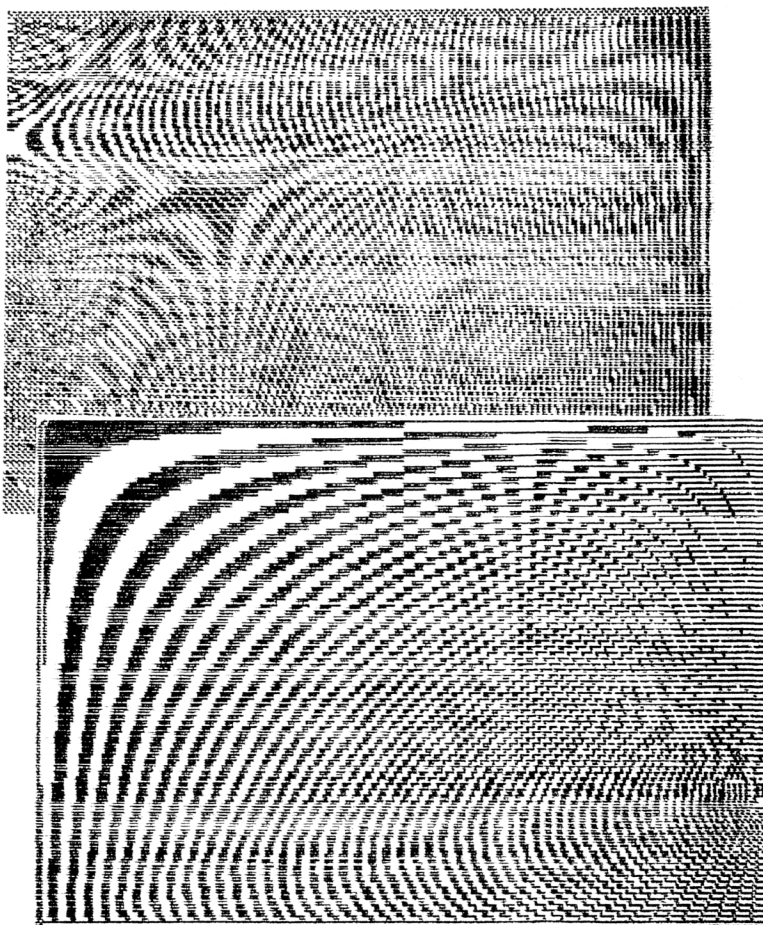
```

10 CIRCLE 120,80,15
20 PLOT 100,80
30 DRAW 40,0
40 DRAW 0,40
50 DRAW -40,0
60 DRAW 0,-40
70 GO SUB 9000
100 STOP
9000 DIM a$(9999)
9010 DIM b$(9999)
9020 LET z=1
9030 INPUT x
9040 INPUT y
9050 PLOT x,y
9060 IF POINT (x+1,y)+POINT (x-1
,y)+POINT (x,y+1)+POINT (x,y-1) <
>4 THEN GO TO 9120
9070 LET z=z-1
9080 IF z=0 THEN RETURN
9090 LET x=CODE a$(z)
9100 LET y=CODE b$(z)
9110 GO TO 9050
9120 LET a$(z)=CHR# x
9130 LET b$(z)=CHR# y
9140 LET z=z+1
9150 FOR c=-1 TO 1
9160 FOR d=- (c=0) TO (c=0) STEP
.1
9170 IF POINT (x+c,y+d)=0 THEN G
O TO 9200
9180 NEXT d
9190 NEXT c
9200 LET x=x+c
9210 LET y=y+d
9220 GO TO 9050

```

# Timothy Leary

This program by Damian Steele creates a number of dramatic graphic displays to demonstrate convincingly (if you still needed convincing) how effective your computer's graphics can be.



```
10 PAPER 5: BORDER 1: INK 1: L
ET D=1: LET E=1: LET F=0: LET X=
0: LET Y=175: LET S=D: LET B=0:
LET C=0: OVER 1
20 CLS : PRINT #0;AT 0,0;"
ZX TRIP P
PRESS ANY KEY IF BORED..."
100 FOR A=X TO Y STEP S
110 IF A>=128 THEN GO TO 400
120 IF A>=87 THEN GO SUB 200
130 PLOT 0,A: DRAW (Y#1.457)-B,
A-C
135 IF INKEY$("<>") THEN GO TO 10
00
140 NEXT A
200 LET B=B+6.1: LET C=C+2
300 RETURN
400 LET S=2
450 GO SUB 700
500 OVER 1
510 FOR A=1 TO 255 STEP S
515 IF INKEY$("<>") THEN GO TO 10
00
520 PLOT A,0: DRAW 0,175
530 NEXT A
535 IF B=1 THEN GO TO 600
537 GO SUB 700
540 FOR A=1 TO 175 STEP S
545 IF INKEY$("<>") THEN GO TO 10
00
550 PLOT 0,A: DRAW 255,0
560 NEXT A
565 GO SUB 700
570 IF B("<>")1 THEN LET B=1: GO TO
450
600 IF E=1 THEN LET E=0: GO TO
537
620 LET F=F+1: IF F<=1 THEN LET
S=1.4: LET B=0: GO TO 410
650 GO SUB 700: GO TO 1000
700 FOR A=1 TO 10: BEEP -.007,(A
ND*30)+35: NEXT A
800 RETURN
1000 PAUSE 0: CLS
1001 LOAD ""
9500 STOP
9990 PRINT FLASH 1;AT 10,10;"STD
D THE TAPE": PRINT AT 21,2;"PRES
S ANY KEY TO CONTINUE...": PAUSE
0: GO TO 1
```

## Color Test

This simple program by David Perry puts all of the computer's colors up on the screen so you can tune in your television for the best possible picture.

```
10 BORDER 2: PAPER 0: BRIGHT 1
   INVERSE 0: OVER 0: FLASH 0: CL
S
20 LET I=-1: FOR N=0 TO 31 STE
P 4
30 LET I=I+1
40 FOR Y=0 TO 3
50 FOR X=0 TO 21
60 PRINT AT X,N+Y: INK I: "■"
70 NEXT X: NEXT Y: NEXT N
80 INK 7
```

## Logic Gate Emulator

This program, by Neville Predebon, shows you the result of having AND, NAND, OR or NOR gates in a circuit. Just follow the prompts.

```
1 REM "gates"
5 REM © Predebon 1983
7 BORDER 6
10 PRINT INK 1; AT 2,13; "GATES"
   INK 0; AT 10,6; "A logic gate ut
   ility"
12 INK 0
15 PRINT AT 10,3; "press any ke
   y to continue": PAUSE 0: CLS
20 PRINT AT 5,1; "Key the gate
   -": PRINT
25 PRINT TAB 5; "AND = 1"
30 PRINT TAB 5; "NAND = 2"
35 PRINT TAB 5; "OR = 3"
40 PRINT TAB 5; "NOR = 4"
45 PAUSE 0: CLS
47 GO SUB 115
50 PRINT AT 0,8; "PUSH 0=OFF":
   PRINT : PRINT
55 PRINT TAB 2; "Switch A - 001
   1"
60 PRINT TAB 2; "Switch B - 010
   1": PRINT
65 IF INKEY$="2" THEN GO TO 85
70 IF INKEY$="3" THEN GO TO 90
75 IF INKEY$="4" THEN GO TO 95
80 PRINT TAB 4; "Output - 0001"
   : GO TO 100
85 PRINT TAB 4; "Output - 1110"
   : GO TO 100
90 PRINT TAB 4; "Output - 0111"
   : GO TO 100
95 PRINT TAB 4; "Output - 1000"
100 INPUT INK 1; "Do you need an
   other run? "; a$
105 IF a$="yes" OR a$="YES" THE
   N GO TO 15
110 PRINT : PRINT INK 2; TAB 9; "
   Very well": STOP
```

```

115 IF INKEY#="1" THEN PRINT AT
120,4,"AND";
125 IF INKEY#="2" THEN PRINT AT
130,4,"NAND";
135 IF INKEY#="3" THEN PRINT AT
140,4,"OR";
145 IF INKEY#="4" THEN PRINT AT
150,4,"NOR";
155 PRINT " GATE": RETURN

```

# Aural Assault

Each RANDOMIZE call in this brief program by Graham White produces a different sound. There are five such sounds which you can incorporate into your own programs.

```

1000 RANDOMIZE
1010 PRINT "RANDOMIZE TO 000000"
1020 PRINT "RANDOMIZE TO 000000"
1030 PRINT "RANDOMIZE TO 000000"
1040 PRINT "RANDOMIZE TO 000000"
1050 PRINT "RANDOMIZE TO 000000"
1060 PRINT "RANDOMIZE TO 000000"
1070 PRINT "RANDOMIZE TO 000000"
1080 PRINT "RANDOMIZE TO 000000"
1090 PRINT "RANDOMIZE TO 000000"
1100 PRINT "RANDOMIZE TO 000000"
1110 PRINT "RANDOMIZE TO 000000"
1120 PRINT "RANDOMIZE TO 000000"
1130 PRINT "RANDOMIZE TO 000000"
1140 PRINT "RANDOMIZE TO 000000"
1150 PRINT "RANDOMIZE TO 000000"
1160 PRINT "RANDOMIZE TO 000000"
1170 PRINT "RANDOMIZE TO 000000"
1180 PRINT "RANDOMIZE TO 000000"
1190 PRINT "RANDOMIZE TO 000000"
1200 PRINT "RANDOMIZE TO 000000"
1210 PRINT "RANDOMIZE TO 000000"
1220 PRINT "RANDOMIZE TO 000000"
1230 PRINT "RANDOMIZE TO 000000"
1240 PRINT "RANDOMIZE TO 000000"
1250 PRINT "RANDOMIZE TO 000000"
1260 PRINT "RANDOMIZE TO 000000"
1270 PRINT "RANDOMIZE TO 000000"
1280 PRINT "RANDOMIZE TO 000000"
1290 PRINT "RANDOMIZE TO 000000"
1300 PRINT "RANDOMIZE TO 000000"
1310 PRINT "RANDOMIZE TO 000000"
1320 PRINT "RANDOMIZE TO 000000"
1330 PRINT "RANDOMIZE TO 000000"
1340 PRINT "RANDOMIZE TO 000000"
1350 PRINT "RANDOMIZE TO 000000"
1360 PRINT "RANDOMIZE TO 000000"
1370 PRINT "RANDOMIZE TO 000000"
1380 PRINT "RANDOMIZE TO 000000"
1390 PRINT "RANDOMIZE TO 000000"
1400 PRINT "RANDOMIZE TO 000000"
1410 PRINT "RANDOMIZE TO 000000"
1420 PRINT "RANDOMIZE TO 000000"
1430 PRINT "RANDOMIZE TO 000000"
1440 PRINT "RANDOMIZE TO 000000"
1450 PRINT "RANDOMIZE TO 000000"
1460 PRINT "RANDOMIZE TO 000000"
1470 PRINT "RANDOMIZE TO 000000"
1480 PRINT "RANDOMIZE TO 000000"
1490 PRINT "RANDOMIZE TO 000000"
1500 PRINT "RANDOMIZE TO 000000"
1510 PRINT "RANDOMIZE TO 000000"
1520 PRINT "RANDOMIZE TO 000000"
1530 PRINT "RANDOMIZE TO 000000"
1540 PRINT "RANDOMIZE TO 000000"
1550 PRINT "RANDOMIZE TO 000000"
1560 PRINT "RANDOMIZE TO 000000"
1570 PRINT "RANDOMIZE TO 000000"
1580 PRINT "RANDOMIZE TO 000000"
1590 PRINT "RANDOMIZE TO 000000"
1600 PRINT "RANDOMIZE TO 000000"
1610 PRINT "RANDOMIZE TO 000000"
1620 PRINT "RANDOMIZE TO 000000"
1630 PRINT "RANDOMIZE TO 000000"
1640 PRINT "RANDOMIZE TO 000000"
1650 PRINT "RANDOMIZE TO 000000"
1660 PRINT "RANDOMIZE TO 000000"
1670 PRINT "RANDOMIZE TO 000000"
1680 PRINT "RANDOMIZE TO 000000"
1690 PRINT "RANDOMIZE TO 000000"
1700 PRINT "RANDOMIZE TO 000000"
1710 PRINT "RANDOMIZE TO 000000"
1720 PRINT "RANDOMIZE TO 000000"
1730 PRINT "RANDOMIZE TO 000000"
1740 PRINT "RANDOMIZE TO 000000"
1750 PRINT "RANDOMIZE TO 000000"
1760 PRINT "RANDOMIZE TO 000000"
1770 PRINT "RANDOMIZE TO 000000"
1780 PRINT "RANDOMIZE TO 000000"
1790 PRINT "RANDOMIZE TO 000000"
1800 PRINT "RANDOMIZE TO 000000"
1810 PRINT "RANDOMIZE TO 000000"
1820 PRINT "RANDOMIZE TO 000000"
1830 PRINT "RANDOMIZE TO 000000"
1840 PRINT "RANDOMIZE TO 000000"
1850 PRINT "RANDOMIZE TO 000000"
1860 PRINT "RANDOMIZE TO 000000"
1870 PRINT "RANDOMIZE TO 000000"
1880 PRINT "RANDOMIZE TO 000000"
1890 PRINT "RANDOMIZE TO 000000"
1900 PRINT "RANDOMIZE TO 000000"
1910 PRINT "RANDOMIZE TO 000000"
1920 PRINT "RANDOMIZE TO 000000"
1930 PRINT "RANDOMIZE TO 000000"
1940 PRINT "RANDOMIZE TO 000000"
1950 PRINT "RANDOMIZE TO 000000"
1960 PRINT "RANDOMIZE TO 000000"
1970 PRINT "RANDOMIZE TO 000000"
1980 PRINT "RANDOMIZE TO 000000"
1990 PRINT "RANDOMIZE TO 000000"
2000 PRINT "RANDOMIZE TO 000000"

```

# Rainbird

Now you can see how your computer can mix its standard colors to produce up to 128 different shades.

Some programs which produce this effect use a graphics character with a checker-board pattern in order to produce a pattern of dots to mix the colors. This program allows for this option, as well as two others. You can see that horizontal lines produce a color which is more steady on the screen than the checker-board pattern produces.

```

2 FOR a=0 TO 7 STEP 2: POKE U
SR "a"+a,BIN 11111111: POKE 158
"a"+a+1,BIN 00000000: NEXT a
3 FOR a=0 TO 7 STEP 2: POKE U
SR "b"+a,BIN 10101010: POKE 159
"b"+a+1,BIN 01010101: NEXT a
4 FOR a=0 TO 7 STEP 2: POKE U
SR "c"+a,BIN 11110000: POKE 160
"c"+a+1,BIN 00001111: NEXT a
6 PAPER 7: INK 1: BORDER 7: C
LS : GO TO 100
10 PAPER 0: INK 7: BORDER 0: C
LS : PRINT AT 16,0;" 0 1 2
3 4 5 6 7 paper"
15 FOR c=0 TO 1
20 PRINT AT 19,12;"bright ";c;
: PRINT #0;"Press any key or 'sp
ace' for the Menu"
25 FOR a=0 TO 7: FOR b=0 TO 7
30 BEEP .003,RND#40-10
35 IF b=0 THEN PRINT AT a#2+1,
b#4;a;AT 0,0;"j.a."
40 IF c=1 THEN PRINT BRIGHT c
: INK a; PAPER b;AT a#2,b#4;"=====
":AT a#2+1,b#4;"=====

```

```

41 IF c=2 THEN PRINT BRIGHT c
: INK a; PAPER b;AT a#2,b#4;"=====
":AT a#2+1,b#4;"=====
42 IF c=3 THEN PRINT BRIGHT c
: INK a; PAPER b;AT a#2,b#4;"=====
":AT a#2+1,b#4;"=====
45 IF b=0 THEN PRINT AT a#2+1,
b#4;a;AT 0,0;"ink"
50 NEXT b: NEXT a
60 IF INKEY$="" THEN GO TO 50
70 IF INKEY$<>" " THEN NEXT c:
GO TO 15
80 GO TO 150
100 PRINT AT 0,1;"128 COLOUR SP
ECTRUM "; OVER 1;AT 0,1;"

```

```

110 PRINT AT 2,1;"This program
demonstrates how the standard e
ight colours of the ZX Spectru
m can be mixed to produce up to
128 different shades."

```

```

120 PRINT "Some programs whic
h use this effect use a graphi
c character with a chessboard t
ype pattern of dots to mix the
colours, this option is provided
but generally a series of horizon
tal lines will work better as
this way the slight dot crawl ef
fect is reduced rather than
exaggerated!"

```

```

130 PRINT AT 19,9;"HIT ANY KEY

```

```

140 PAUSE 0
145 CLS
150 PAPER 7: INK 1: BORDER 7: C
LS : PRINT AT 1,13; INVERSEF 1;"M
ENU"

```

```

160 PRINT AT 5,0;"1:Horizontal
lines ===== HIT a";AT 7,2;"2:Ch
essboard pattern ===== KEY";AT 9
,0;"3:Dashed lines ===== 1
-3"

```

```

170 PRINT AT 13,1;"The colour c
ombination for each shade can be
read from the axes of the chart.
It must be noted that the ink
and paper colours are not inter
changeable, that is for example,
blue ink on red paper is a di

```

```

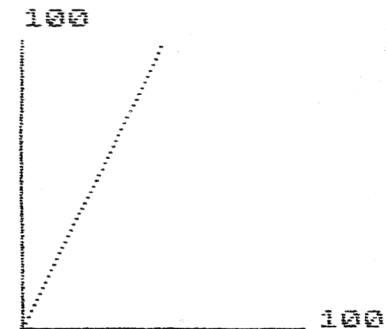
fferent colour to red ink no bl
ue paper."
180 PRINT #0;AT 0,1; INK 1;"The
routines for generating theuser
graphics are in lines 2-4."
190 IF INKEY$(">") THEN GO TO 19
@
200 LET a$=INKEY$
205 IF a$>"3" OR a$<"1" THEN GO
TO 200
210 LET c:=VAL a$: GO TO 10

```

## Line of Best Fit

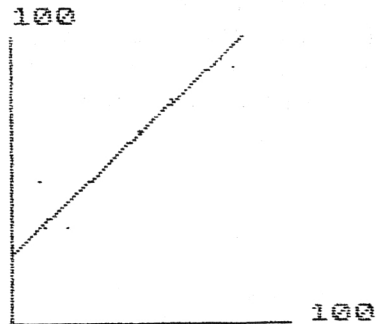
This useful utility program, written by Neal Cavalier-Smith, allows you to enter a number of points. The computer will then plot a line of best fit for these points, and give the equation of this line. You can see it in action in this sample run:

Equation of best fitting line is  
 $Y=2X+0$



Input points were... (1,2)  
 ( 4, 8) (40,80) (23,46)

Equation of best fitting line is  
 $Y=0.93X+24.5$



This is the listing for your own 'line of best fit':

```

10 LET a=0: LET b=0: LET m=0
20 LET k=0: INK 7 : PAPER 0
30 CLS : BORDER 0
40 INPUT "How many points?";n
50 DIM x(n)
60 DIM y(n)
70 FOR c=1 TO n
75 INPUT x(c),y(c)
76 PLOT x(c),y(c)
80 LET a=a+x(c)
90 LET b=b+y(c)
100 NEXT c
110 LET a=a/n: LET b=b/n
120 FOR c=1 TO n
130 LET m=m+(((x(c)-a)*(y(c)-b)
) / ((ABS (x(c)-a))2))
140 NEXT c
145 LET m=m/n
150 LET k=b-m#a
160 LET c=0
195 PLOT 0,0: DRAW 100,0: PRINT
AT 21,13;"100"
196 PLOT 0,0: DRAW 0,100: PRINT
AT 6,6;"100"

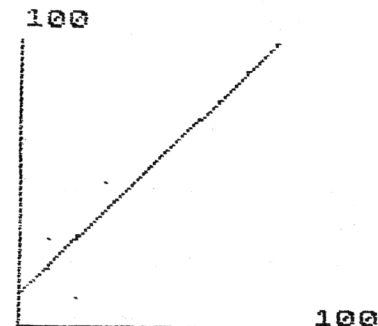
```

```

200 PRINT AT 0,0;"Equation of b
est fitting line is"
210 PRINT AT 0,0;"Y=";INT (m*10
0+.5)/100;"X+";INT (k*10+.5)/10
220 FOR c=1 TO 100
225 LET d=m#c+k
230 IF d<100 THEN PLOT c,d
240 NEXT c
250 STOP

```

Equation of best fitting line is  
 $Y=0.98X+11$





# Hall of Fame

This can be used in any arcade type game. As an example, type in a number above 100, then your name, and in a second or so the computer will calculate and print up the new Hall of Fame. This program was created by David Perry.

```
1 REM HALL OF FAME
2 REM (C) DAVID PERRY 1983
3 REM
4 REM NUM=NUMBER OF SCORES
5 REM
6 REM
9 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS
10 LET NUM=10: DIM N(NUM+1): D
IM N$(NUM+1,8): FOR N=1 TO NUM:
LET N(N)=1100-(N*100): LET N$(N)
="SPECTRUM": NEXT N
15 CLS : INPUT "FOR AN EXAMPLE
ENTER YOUR SCORE          SCO
RE:":SC
20 IF SC<=N(10) THEN GO TO 20
0
30 LET NUM=11: IF SC>=N(NUM) T
HEN INPUT "ENTER 8 INITIALS! ":P
$: IF LEN P$>8 THEN GO TO 20
35 PRINT AT 7,0: FLASH 1: BRIG
HT 0: INK 7: PAPER 2:"THIS WILL
ONLY TAKE A FEW SECS!"
40 IF SC>=N(NUM) THEN LET N(N
UM)=SC: LET N$(NUM)=P$
```

```
50 FOR A=1 TO (NUM-1): LET B$=
N$(A): LET C$=N$(A+1): LET B=N(A
): LET C=N(A+1): IF B<C THEN LE
T N(A)=C: LET N(A+1)=B: LET N$(A
)=C$: LET N$(A+1)=B$
60 NEXT A: FOR N=1 TO NUM-1: I
F N(N)<N(N+1) THEN GO TO 50
70 NEXT N
80 CLS
90 PRINT AT 2,4:"H A L L O F
F A M E !"
100 PRINT AT 3,4:"=====
===== "
110 FOR N=1 TO NUM-1: PRINT AT
N+5,7: INK 6: "(" : INK 2:N: INK
6: ")":AT N+5,12: INK 7:N(N): PRI
NT AT N+5,17: INK 5:N$(N): NEXT
N
120 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
PRINT AT 17,8: PAPER 2: INK 7:"
PRESS ANY KEY! ": LET I=0
130 LET I=I+1: IF I>7 THEN LET
I=0
140 PRINT AT 2,4: INK I:"H A L
L O F F A M E !"
150 BEEP .01,I*7: PAUSE 2: IF I
NKEY$="" THEN GO TO 130
160 GO TO 15
200 PRINT AT 10,4:"SORRY SCORE
TO LOW!!!"
210 PRINT AT 13,7:" PRESS A KEY
!"
220 PAUSE 0: RUN
```

# Spec File

This outstanding database program comes from Michael Briggs. It stores data in a very compact way, ensuring maximum use is made of the memory.

When the program is run, you'll be asked to enter your name, the file title, field titles and the like. Note that an entry cannot be longer than 256 characters, and a field must be 28 characters or less.

A menu will be displayed, with the following options:

1/ BEGIN NEW FILE: This runs the program, clearing all data

2/ ENTER RECORD(S): This will ask you to enter a record, field by field. It will then sort the record into alphabetical order with the rest of the file. If you make a mistake when entering any field, simply enter "Q" to return to the menu

3/ SEARCH: This searches through the file for either a string of letters or a numbered record

4/ PRINT ALL RECORDS: This prints the first field of all files and allocates a specific entry number to them

5/ SAVE FILE: This saves the file under the title of the file. If the title is an invalid file name, the word "index" will be used

6/ STATUS: This will tell you various facts about the state of the program, such as much space is left in the file, how many entries there are, and how long the computer has been running the file

You'll find that even though this sounds complex when you're reading about it, you'll find it is really quite simple to run the program.

```
      : REM MICHAEL BRIGGS
      2 POKE 23609,40: BORDER 6: IN
K 1: PAPER 7: CLS : DIM Z$(35000
): LET Z=1: LET CO=0
      8 INPUT "FORENAME...":C$
      9 INPUT "SURNAME...":S$
     10 INPUT "TITLE OF FILE ":T$
     20 INPUT "NUMBER OF FIELDS ":F
     30 DIM A$(F,20): FOR A=1 TO F:
PRINT "ENTER TITLE FOR FIELD ":
A: INPUT B$: LET A$(A,2 TO )=B$:
LET A$(A,1)=CHR$ LEN B$: PRINT
":-": INK 2:B$: PRINT : NEXT A
     70 PAUSE 50
     100 POKE 23609.40: BORDER 6: CL
S : INK 0: PLOT 0,0: DRAW 255,0:
DRAW 0,175: DRAW -255,0: DRAW 0
,-175: INK 3
```

```

110 PRINT INK 0: BRIGHT 1: AT 1
,13-LEN T#/2:T#: " MENU"
120 PRINT AT 3,3: " NO 1.BEGIN
NEW FILE."
130 PRINT AT 5,3: " NO 2. ENTER
NEW RECORD(S)."
140 PRINT AT 7,3: " NO 3. SEARCH
RECORDS."
150 PRINT AT 9,3: " NO 4. PRINT
ALL RECORDS."
160 PRINT AT 11,3: " NO 5. SAVE
FILE."
165 PRINT AT 13,3: " NO 6. FILE
STATUS."
170 PRINT INK 2: AT 15,8: "@ MIC
HAEL BRIGGS"
180 PRINT INK 1: AT 17,11-(LEN
C#+LEN S#)/2: "FILE FOR ":C#: " ":
S#
190 INPUT "CHOICE...":C
195 CLS
196 INK 1
200 IF C<C AND C<8 THEN GO TO
C*1000
210 GO TO 100
1000 PRINT AT 10,2: "ARE YOU SURE
":C#: " ?"
1010 INPUT Y#
1020 IF Y#(1)="Y" OR V#(1)="Y" T
HEN RUN
1030 GO TO 100
2000 INK 2: FOR A=1 TO F
2010 PRINT AT A*2-1,13-(CODE A#(
A,1)/2): "ENTER ":A#(A,2 TO )
2020 NEXT A

```

```

2021 PRINT AT 21,0: INK 1: "ENTER
~Q~ TO RETURN TO MENU."
2030 LET E#=""
2040 FOR A=1 TO F
2050 PRINT AT A*2-1,13-(CODE A#(
A,1)/2): "ENTER ": FLASH 1:A#(A,2
TO 1+CODE A#(A))
2060 INPUT F#
2061 IF F#="Q" THEN GO TO 100
2070 LET E#=E#+CHR# LEN F#+F#
2080 PRINT AT A*2-1,13-(CODE A#(
A,1)/2): "ENTER ":A#(A,2 TO 1+COD
E A#(A))
2090 PRINT INK 0: AT A*2,16-LEN
F#/2:F#
2100 NEXT A
2110 LET X1=1: FOR A=1 TO CO: LE
T X#=Z#(X1+2 TO X1+CODE Z#(X1+1)
+1)
2115 IF X#>E#(2 TO ) THEN GO TO
2400
2120 LET X1=X1+CODE Z#(X1)
2125 NEXT A
2126 LET E#=CHR# (LEN E#+1)+E#:
LET Z#(Z TO Z+LEN E#)=E#
2131 LET Z=Z+LEN E#
2140 LET CO=CO+1: CLS : GO TO 20
00
2400 LET E#=CHR# (LEN E#+1)+E#
2410 LET Z#(X1+LEN E# TO Z+LEN E
#)=Z#(X1 TO Z)
2420 LET Z#(X1 TO X1+LEN E#-1)=E
#
2430 GO TO 2131
2999 REM SEARCH ROUTINE

```

```

3000 PRINT TAB 11:"SEARCH MODE":
AT 2,0:"ENTER...":AT 3,0:" ^N^ I
F ENTERING RECORD NUMBER.  ^~ I
F ENTERING CHARS ONLY.": INPUT "
KEYWORD":K#
3004 LET M#="" ": IF K$(1)="N" TH
EN LET M#=K$(1): LET K#=K$(2 TO
)
3005 PRINT TAB 8:"KEYWORD=":K#
3006 LET RET=3040
3010 LET X=1: FOR A=1 TO CO
3015 LET X1=X: LET V#=Z$(X TO (C
ODE Z$(X)+X-1))+ " ": IF M#="N" T
HEN GO TO 3300
3016 FOR J=1 TO LEN V#-LEN K#
3020 IF V$(J TO J-1+LEN K#)=K# T
HEN GO TO 3200
3030 NEXT J
3040 LET X=X1+LEN V#-1: NEXT A
3050 PRINT " " INK 0:"PRESS A KE
Y": PAUSE 0: GO TO 100
3200 PRINT " " INK 4:"ENTRY NO...
":A: LET X2=X1: LET X1=X1+1: FOR
K=1 TO F
3210 PRINT INK 2:A$(K,2 TO CODE
A$(K)+1):"...": PRINT INK 1:TA
B 32-CODE Z$(X1):Z$(X1+1 TO X1+C
ODE (Z$(X1)))
3220 LET X1=X1+1+CODE Z$(X1)
3225 NEXT K
3240 LET X1=X2: GO TO RET
3300 IF A<>VAL K# THEN GO TO 30
40
3310 LET RET=3050
3320 GO TO 3200

```

```

4000 PRINT "ANY KEY WILL STOP TH
E SCROLLING DISPLAY UNTIL KEY IS
RELEASED."
4005 PAUSE 0
4010 PRINT AT 18,16-LEN T$/2: IN
K 2:T#
4020 INK 1: PRINT
4030 PRINT
4040 LET X=1: FOR A=1 TO CO
4049 PRINT AT 21,0: POKE 23692.-
1
4050 PRINT "0000":AT 21,4-LEN (S
TR# A):A; " ":Z$(X+2 TO X+1+CODE
(Z$(X+1)))
4060 LET X=X+(CODE Z$(X))
4065 IF INKEY$<>" " THEN GO TO 4
065
4070 NEXT A
4075 PRINT " "PRESS ANY KEY TO R
ETURN TO MENU."
4080 PAUSE 0: GO TO 100
5000 PRINT "I SHALL SAVE THE FIL
E UNDER THE NAME OF"
: PRINT AT 10,16-LEN T$/2: FLASH
1:T#
5005 IF T#="" OR LEN T#>10 THEN
PRINT AT 10,1:" IN
DEX " ": SAVE "INDEX"
LINE 100: GO TO 100
5010 SAVE T# LINE 100
5020 GO TO 100
6000 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
6010 PRINT AT 1,12-(LEN T#)/2: B
RIGHT 1: INK 2:T#:" STATUS"

```

```

6020 PRINT AT 3,2;"COMPUTER ON T
IME  ":"INT (((PEEK 23672+256*PE
EK 23673)+65536*PEEK 23674)/50):
" SECS"
6030 PRINT AT 5,10-(LEN S#+LEN C
#)/2;"FILE FOR ";C#;" ";S#
6040 PRINT AT 7,8;"NO.OF RECORDS
 ":"CO
6050 PRINT AT 9,2;"CHARACTER SPA
CE LEFT  ":"35000-Z
6060 PRINT AT 11,4;"CHARACTERS I
N FILE  ":"Z
6070 PRINT AT 13,3;"PROGRAM:-@ M
ICHAEL BRIGGS"
6075 PRINT AT 20,5;"PRESS ANY KE
Y FOR MENU"
6080 PRINT AT 3,2;"COMPUTER ON T
IME  ":"INT (((PEEK 23672+256*PE
EK 23673)+65536*PEEK 23674)/50):
" SECS"
6085 IF INKEY$<>"" THEN GO TO 6
100
6090 GO TO 6080
6100 GO TO 100

```

# STRUCTURED PROGRAMMING



## Sketching an outline

Many times I've written articles and chapters in books which are supposedly going to improve the programming skill of those who read them. But every time I produce such a list of 'things you really should be doing when you program', or explain the material to someone, I am reminded of an old story about a farmer.

He was approached by a young man selling correspondence courses in 'Effective Farming'. "Don't you see," the young salesman said as part of his pitch, "that if you take this course you'll know so much more about farming?". The farmer replied "I don't even farm now as good as I know how."

So it goes, I suggest, for advice on programming. I know full well that I do not "program as good as I know how". Many times I break all the rules, wading straight into coding a complex program without even a thought for the ideas of 'structured programming'.

So I suggest you keep in mind, when reading this appendix, that I do not really believe that you - or anyone else - is really going to take the ideas here as rules which must be obeyed, come hell or high water. The best way to approach this,

and any other material in a similar vein which you come across, is to read it carefully, and make your own assessment on each suggestion given. Then, just apply the ideas which seem most sound to you.

The basic idea I'm outlining in this appendix is that of approaching structured programming by 'sketching an outline'.

The idea is simple, but very valuable in order to help you write complicated and involved programs, such as many of those in this book. Of course, you may well be already creating very complex programs without using anything like the idea I'm about to outline. Even if you are, I suggest you think carefully about these ideas, so that you can see that they may make your job easier.

The fundamental idea of structured programming lies hidden in the phrase 'top down programming'. This suggests that you start the process of programming by first stating in words the broad aim of the program you are about to write. You follow this up by writing a series of notes - each of which will eventually be a subroutine - which cover each of the main tasks which the computer will carry out.

To clarify this last paragraph, I'll give you a concrete example. Here are the steps you could follow when trying to write a Noughts and Crosses program. (Note that

'structured programming' produces listings which are much longer than might otherwise be the case. However, this greater length is more than offset by the fact that programs built up in this way are very simple to debug and improve, and their structure can be readily understood by other programmers. You may not be able to make such claims with confidence about some of your current programs.)

SET UP INITIAL BOARD

COMPUTER MOVE

(A) CHECK IF MIDDLE SQUARE EMPTY,  
IF NOT MOVE THERE

(B) CHECK IF THERE IS A COMPUTER  
WINNING MOVE, IF SO MAKE IT

(C) CHECK IF HUMAN WILL WIN ON NEXT  
MOVE, IF SO BLOCK

(D) IF NO MOVE MADE, CHECK TO SEE  
IF A RANDOM MOVE CAN BE MADE, IF SO  
MAKE IT, IF NOT DECLARE A DRAW

PRINT BOARD

ACCEPT PLAYER MOVE

PRINT BOARD

CHECK IF PLAYER HAS WON, IF SO STOP

GOTO 'COMPUTER MOVE'

As you'll see if you take a few moments to study this sketched outline, you can tell exactly which steps the computer will follow; the decisions it will be called to make; and the alternative results of those decisions.



You have already completed the first step towards writing a Noughts and Crosses program. You'll see, by the way, that it doesn't matter at all if you do not yet know how you are going to get your computer to carry out the necessary steps. All that matters is that you know that the steps must, in some fashion, be executed in due course.

The second step in the process consists of turning the sketched outline into a series of subroutine calls. In languages which are more structured than the BASIC you have on your computer (such as the BASIC provided on the BBC Micro, or Pascal) it is relatively easy to call a number of subroutines within an endless loop, with a structure like DO/WHILE or REPEAT/UNTIL. These allow you to repeat a series of program steps indefinitely until a certain condition is satisfied, or the state of a pointer or flag changes.

In the BASIC we have here, however, we have to be content with the humble, and much-maligned GO TO. (Much of the reaction against GO TO, which many 'serious programmers' see as the greatest insult to the art of true programming ever developed, came from early and relatively primitive versions of BASIC, in which you can only follow an IF/THEN with a GO TO. This meant the program leapt all over the place, leading to 'spaghetti code' which was horrendously difficult to interpret.)

The second stage is to turn our sketched outline into a series of subroutine calls, endlessly cycled by a GO TO as follows:

```
10 REM NOUGHTS AND CROSSES
20 GOSUB 9000:REM INITIALISE
30 GOSUB 1000:REM COMPUTER MOVE
40 GOSUB 8000:REM PRINT BOARD
50 GOSUB 2000:REM ACCEPT PLAYER MOVE
60 GOSUB 8000:REM PRINT BOARD
70 IF human has not won AND computer
   has not won THEN GO TO 30
80 PRINT "CONGRATULATIONS"
```

Now, as you can see, we have the 'framework' for a workable Noughts and Crosses program, even though we do not yet have a clue as to how the program will actually work. We can now set about writing the program from first principles.

There are two further advantages of this 'outline' approach. If there is something we cannot, at this stage, actually program (like the basis upon which the computer finds out who has won) we can simply put in a PRINT statement within the subroutine like PRINT "CHECKING FOR WIN" and continue to use the program, as we work on it. Then, each time the computer should be checking for a win, it will print up CHECKING FOR WIN. This means you can continue working on the program, without being needlessly held up on a minor

subroutine which you cannot, for the moment, solve.

The second advantage comes towards the end of the program development stage, the debugging stage. I always find this the most frustrating, and in many ways, the least rewarding aspect of programming. Although I can often get a program working reasonably well fairly quickly (although my first chess in BASIC too nearly six months), to get the program from 'working reasonably well' to 'performing without faults under all conditions' can take as long as it took to get the first version even working at all. However, when you set up the program in the way described in this section, you'll see that the debugging stage can be greatly simplified.

For example, you may find in your Noughts and Crosses program that the computer tends to ignore the bottom right hand corner, when a move into this position would enable it to win, or to block a win from the human. From the 'subroutine loop' we set up before, we know the computer's move must be made somewhere between lines 1000 and 1999. This immediately narrows down the search. If you've been clever, and have made each subroutine a series of further subroutine, each constructed in a similar way to our major subroutine, it would be even easier to track down the bug.

Let's look at this idea - making each subroutine a series of further subroutines - a little more closely. We'll look at the subroutine starting at line 1000, the one in which the computer actually makes its move, as this is the most important (and difficult to program) of all those in this game.

The subroutine could begin as follows:

```
2000 REM COMPUTER MOVE
2010 LET MOVE=0:REM IF THIS BECOMES 1 A
      VALID MOVE HAS BEEN FOUND
2020 GOSUB 2200:REM CHECK IF MIDDLE
      SQUARE IS EMPTY
2030 IF MOVE=1 THEN RETURN
2040 GOSUB 2400:REM CHECK IF A POSSIBLE
      WINNING MOVE EXISTS
2050 IF MOVE=1 THEN RETURN
2060 GOSUB 2600:REM CHECK IF A POSSIBLE
      HUMAN WIN CAN BE BLOCKED
2070 IF MOVE=1 THEN RETURN
2080 GOSUB 2800:REM CHECK IF ANY MOVE AT
      ALL CAN BE MADE
2090 IF MOVE=1 THEN RETURN
2100 REM A RETURN WITH MOVE=0 MEANS NO
      FURTHER MOVES ARE POSSIBLE
2110 RETURN
```

As I said earlier, programs constructed using the 'endless loop of subroutines' coupled with 'subroutines within subroutines' make listings which may be far longer than usual. However, there is usually little need to worry about running

out of memory (as was the case on the ZX81 and the T/S 1000 when counting bytes became the dominant feature of our programming lives). The extra typing involved in producing the longer listings will be amply repaid by the extra clarity your programs attain, and the much shorter time it will take to produce a debugged masterpiece of which you can be proud.

You'll find if you take the trouble to start with a 'sketched outline', the whole process of constructing a major program is much, much simpler than might otherwise be the case. You'll also find that the time involved will be much more constructively spent than it would have been if you had just waded into the programming without taking the time to do your sketching first.

Ignore the old farmer, and try now to program 'as good as you know how'.