AWESOME GAMES FOR YOUR ATARI COMPUTER

HAL RENKO / SAM EDWARDS

Awesome GAMES for your ATARI computer

Awesome GAMES for your ATARI computer

Hal Renko and Sam Edwards

ADDISON-WESLEY PUBLISHING COMPANY
Reading, Massachusetts · Menlo Park, California
London · Amsterdam · Don Mills, Ontario · Sydney

Cover design by Stuart Hughes
Illustrations by Agnes Lehár-Graham

Typeset by Blackmore Press, Shaftesbury
Printed in Finland by Werner Söderström Osakeyhtiö, Member of Finnprint

ABCDE 8987654 0 201 16477 9

Contents

Because the ATARI printer can't print graphic characters and reverse video, the following codes are used in the listings:

Text between @ are in reverse video.

Text between [ ] should be entered while holding down the control key.

Text between < > indicate a special key e.g. <SHIFT> means press shift key.

Acknowledgements

This book was born at the University of Twente in the Netherlands where we met the following games fanatics: H. Hermens, P. de Vries, A. Stapel, H. Riesebos, A. Rensink, M. Sijbenga, E. Groenhuis, W. Koolhoven, R. Bosch, A. Pelsmaeker, K. Boon, and B. Hoogstraten. We would like to thank them all for the many hours they spent with us, discussing, creating, and writing the programs for the games included here.

Introduction

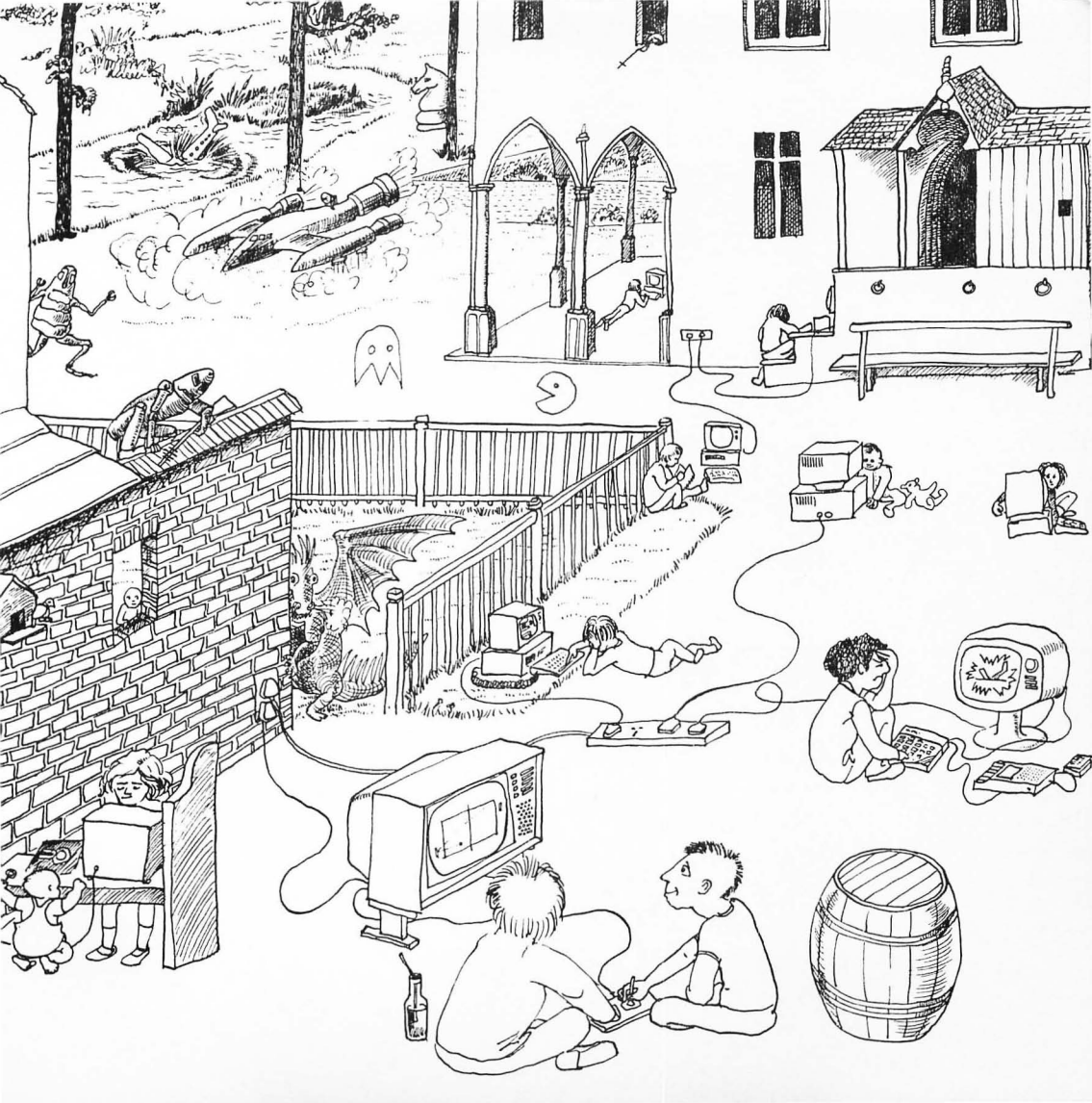
Everybody knows that computers are used for all kinds of serious purposes — financial calculations, business applications and text processing, for instance. They can, however, help us in another way. As the proverb says 'All work and no play makes Jack a dull boy' and it's certainly true that we all need something to keep the cogwheels of our mind turning when we are not considering the 'important' things in life. So we read books, do hobbies, and play games.

Bruegel's wonderful painting 'Children at Play' shows us that this has been true for centuries, if not forever.

This book aims to provide something for everybody from 6 to 96. There are games to play on your own, and others to play with your family and friends. You will find fast action games to test your reflexes and your ability to control many moving objects on the screen; puzzles and brainteasers to get you thinking; board games where you pit your wits against the computer; and, of course, some wonderful arcade-type games which combine aspects of all of these. There are also some very intriguing games which are in a class of their own.

A number of the games have been devised to give interesting and worthwhile results from relatively short listings that will not take much time to enter.

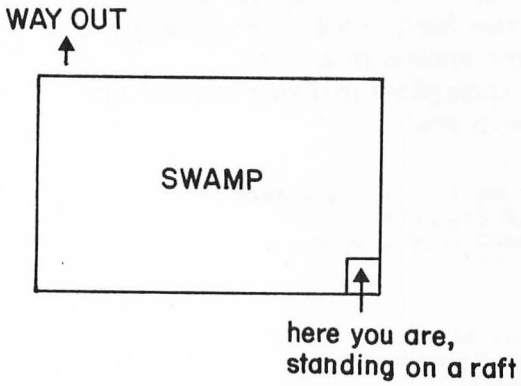
Looking again at 'Children at Play' a thought springs to mind. If Bruegel were alive today, would his painting look something like this . . .



Zombies in the Swamp

Watch out! There are zombies about, and you have to cross the swamp where they live. To do this you must walk across on rafts.

At the start of the game the swamp looks like this:



You are at one corner, and you must reach the diagonally opposite corner to escape. To put a raft next to the one you are standing on, enter T and the computer asks

THROW TO THE?

Enter N for north, S for south, E for east, or W for west, and a raft will appear in the appropriate position (north is at the top of the screen). To move, enter M and the computer will ask

MOVE TO THE?

Again, enter N, S, E, or W as appropriate, to step onto the raft.

So far so good, but beware! From time to time one of those zombies will emerge from the swamp and run across the rafts you have positioned. If you are in its way it will catch you. Luckily, zombies can't see very well so they often fall off the rafts back into the swamp.

To succeed in this exciting game you must develop your own strategy. The longer and more complicated your path the less likely the zombies are to catch you, but, on the other hand, it will take you longer to get across the swamp, giving time for more zombies to appear.

Oh no! Here come those terrifying zombies again. You had better just hope they don't catch you.

```
10 REM #### Zombies In The Swamp ####
20 DIM SW(7,8),C$(1),CS$(4),I$(3)
30 DIM PL$(3),ZO$(3),FL$(3),BL$(3)
40 OPEN #2,4,0,"K:"
50 GRAPHICS 17
60 GOTO 1800
99 REM ### Initialize ###
100 PL$="@□JJJ□@":ZO$="@###@"
110 BL$="    ":FL$="..."
120 FOR C=0 TO 7
130 SW(0,C)=-1:SW(C,0)=-1
140 SW(7,C)=-1:SW(C,8)=-1
150 NEXT C
160 SW(7,8)=-1:SW(0,8)=-1
170 SW(0,1)=1:SW(1,0)=1
180 FOR R=1 TO 6:FOR C=1 TO 7
190 SW(R,C)=0:NEXT C:NEXT R
200 PX=6:PY=7:SW(6,7)=1
210 X=6:Y=7:T$=PL$:GOSUB 1700
220 RETURN
299 REM ### get command ###
300 GET #2,C:C$=CHR$(C)
310 FOR I=1 TO LEN(CS$)
320 IF C$=CS$(I,I) THEN RETURN
```

```
330 NEXT I:GOTO 300
399 REM ### GET direction ###
400 CS$="NESW":GOSUB 300
410 DX=(I=2)-(I=4):DY=(I=3)-(I=1)
420 X=PX+DX:Y=PY+DY
430 IF SW(PX+DX,PY+DY)<>-1 THEN RETURN
440 POSITION 0,23
450 ? #6;"Impossible-TRY AGAIN";
460 POP :GOTO 520
499 REM ### execute command ###
500 POSITION 0,23
510 ? #6;" ???                    ";
520 CS$="TM":GOSUB 300
530 ON I GOTO 600,700
599 REM ### throw ###
600 POSITION 0,23
610 ? #6;"THROW TO THE ?          ";
620 GOSUB 400:SW(X,Y)=1
630 IF X*Y>0 THEN T$=FL$:GOSUB 1700
640 RETURN
699 REM ### move ###
```

```
700 POSITION 0,23
710 ? #6;"MOVE TO THE ?        ";
720 GOSUB 400:LX=X:LY=Y
730 IF SW(X,Y)<>1 THEN POSITION 0,22:? #
6;"into the swamp[AAAL]   AGAIN ?        "
;:GOTO 1200
740 IF X=0 OR Y=0 THEN POSITION 0,22:? #
6;"you made it[AAA],      AGAIN ?       ":G
OTO 1200
750 X=PX:Y=PY:T$=FL$:GOSUB 1700
760 PX=LX:PY=LY
770 X=LX:Y=LY:T$=PL$:GOSUB 1700
780 RETURN
799 REM choose zombie's direction ###
800 ON RND(0)*2+0.5 GOSUB 1300,1400
810 IF SW(ZX+DX,ZY+DY)<1 THEN 900
820 IF SW(ZX+DY,ZY-DX)=1 OR SW(ZX-DY,ZY+
DX)=1 THEN 1000
830 RETURN
899 REM ### swamp ahead ###
900 ON RND(0)*4+0.5 GOSUB 1300,1400,1500
,1600
910 IF SW(ZX+DX,ZY+DY)=-1 THEN 900
920 RETURN
999 REM ### possible move ###
1000 LX=DX:LY=DY
1010 ON RND(0)*4+0.5 GOSUB 1300,1400,150
0,1600
1020 IF DX=-LX AND DY=-LY OR SW(ZX+DX,ZY
+DY)<1 THEN DX=LX:DY=LY
1030 RETURN
1099 REM ### zombie's turn ###
1100 ZX=6:ZY=7:GOTO 1160
1110 IF ZX=PX AND ZY=PY THEN POSITION 0,
22:? #6;"you have just been  eaten[AAAL]
AGAIN ?";:GOTO 1200
1120 GOSUB 800
1130 T$=FL$:GOSUB 1700
1140 ZX=ZX+DX:ZY=ZY+DY
1150 IF SW(ZX,ZY)=0 THEN SW(ZX,ZY)=-1:PO
SITION 3*ZX-2,3*ZY-2:? #6;"0#0":RETURN
1160 X=ZX:Y=ZY:T$=ZO$:GOSUB 1700
1170 GOTO 1110
```

```
1190 REM ### again ###
1200 POKE 764,225:GET #2,X
1210 IF X=89 THEN RUN
1220 GRAPHICS 0:END
1300 DX=0:DY=-1:RETURN :REM north
1400 DX=-1:DY=0:RETURN :REM west
1500 DX=0:DY=1:RETURN :REM south
1600 DX=1:DY=0:RETURN :REM east
1690 REM ### print field ###
1700 FOR I=0 TO 2
1710 POSITION X%3-3,Y%3-3+I:? #6;T$
1720 NEXT I:RETURN
1790 REM ##### main program #####
1800 GOSUB 100:REM initialize #
1810 GOSUB 500:REM player     #
1820 TU=TU+1:IF RND(0)%(PX+PY)>3 OR TU<5
 THEN 1810
1830 GOSUB 1100:REM zombies   #
1840 GOTO 1810
```

Galactic Monsters

There it is, registering on the x-y-Gz radar. At last, you are approaching the Milky Way. In just 2.56 protoseconds you will be safe in your home galaxy!

But there is danger here, and as captain of your ship, you know very well what it is. You keep a watchful eye on the XR6-screen. No alert yet. The tension is unbearable . . . 2.5 protoseconds . . . 2.0 protoseconds . . . 1.5 protoseconds . . . oh no! There it is —

THE VAN ALLEN SQUARE STONE BELT

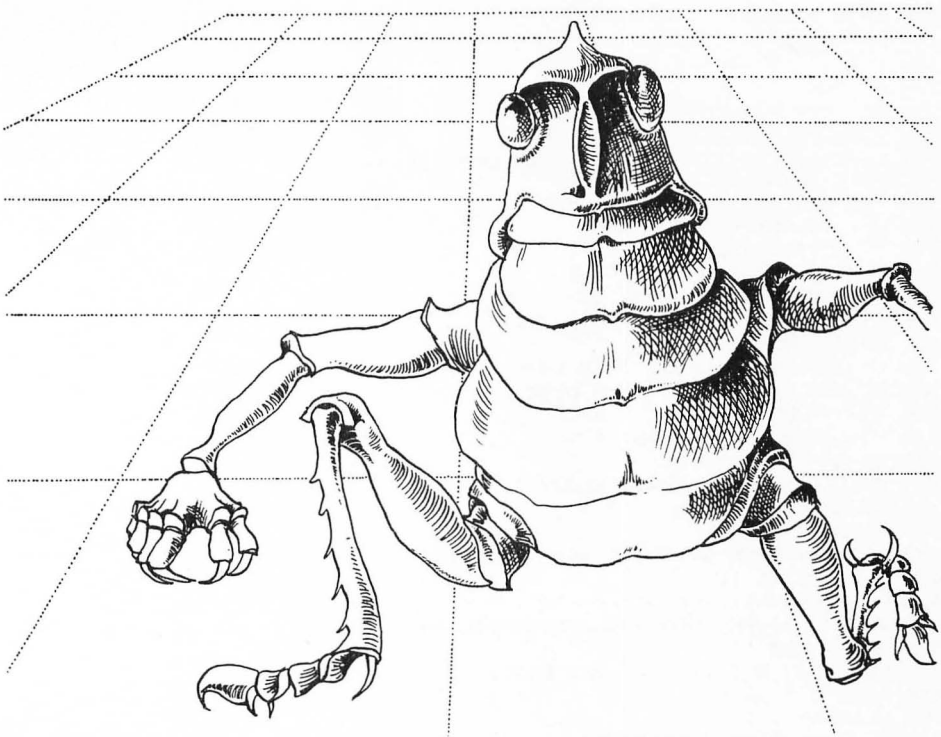
There are a number of these notorious belts, and in them lurk the dreaded Galactic monsters. No weapon can defend you against these vengeful creatures; all you can do is avoid them.

Once the program is started and you see a *VAN ALLEN SQUARE STONE BELT* then you can move yourself by pressing:

- to move one SQUARE upward
+ to move one SQUARE to the left
* to move one SQUARE to the right
= to move one SQUARE downward

Those terrible Galactic monsters will approach closer and closer. Your only hope for survival is to avoid them! If you cross a *VAN ALLEN SQUARE STONE BELT* a second time, the monsters will become more aggressive. Somehow, they seem to anticipate your every move. If they manage to attack you five times, you and your ship are totally destroyed.

Good luck, captain . . . only a few protoseconds to go, and you are home free!



```
10 REM ####### Galactic Monsters #######
20 DIM X(11),Y(11),STIK(9,1)
30 OPEN #2,4,0,"K"
40 GRAPHICS 18
50 GOTO 900
99 REM ### Initialize ###
100 FOR I=0 TO 9
110 READ X,Y:STIK(I,0)=X:STIK(I,1)=Y
120 NEXT I
```

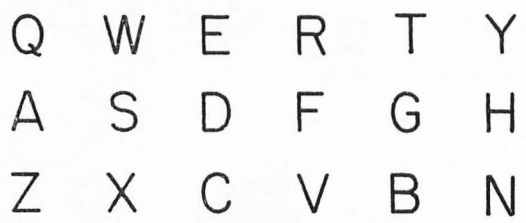
```
130 DATA 1,1,1,-1,1,0,0,0,-1,1
140 DATA -1,-1,-1,0,0,0,0,1,0,-1
150 ? #6;"  galactic monsters"
160 RETURN
199 REM ### player's move ###
200 SOUND 0,20,10,15
210 KEY=STICK(0):IF KEY<>15 THEN DX=STIK
(KEY-5,0):DY=STIK(KEY-5,1):GOTO 290
220 IF PEEK(764)=255 THEN 340
230 GET #2,KEY:DX=0:DY=0
240 IF KEY=42 THEN DX=1:GOTO 290
250 IF KEY=43 THEN DX=-1:GOTO 290
260 IF KEY=45 THEN DY=-1:GOTO 290
270 IF KEY=61 THEN DY=1:GOTO 290
280 GOTO 340
290 IF ABS(10-XP-DX)>5 THEN DX=0
300 IF ABS(6-YP-DY)>5 THEN DY=0
310 TURN=TURN+1
320 POSITION XP,YP:? #6;"#";
330 XP=XP+DX:YP=YP+DY
340 POSITION XP,YP:? #6;"0[J]0";
350 SOUND 0,0,0,0
360 RETURN
399 REM ### move galactic monsters ###
400 FOR T=0 TO 10
410 POSITION X(T),Y(T):? #6;"#";
420 IF RND(0)<PR THEN GOSUB 500:GOTO 440
430 GOSUB 600
440 POSITION X(T),Y(T):? #6;"[C]";
450 NEXT T
460 RETURN
499 REM ### move monster to player ###
500 IF RND(0)>0.8 THEN X(T)=X(T)+SGN(XP-
X(T)):RETURN
510 Y(T)=Y(T)+SGN(YP-Y(T)):RETURN
599 REM ### move monster random ###
600 A=INT(RND(0)*4)
610 X(T)=X(T)+(A=0 AND X(T)<15)-(A=1 AND
 X(T)>5)
620 Y(T)=Y(T)+(A=2 AND Y(T)<11)-(A=3 AND
 Y(T)>1)
630 RETURN
```

```
690 REM ### eaten by monster ? ###
700 FOR T=0 TO 10
710 IF XP<>X(T) OR YP<>Y(T) THEN 780
720 K=K+1:T=10
730 SOUND 0,200,10,15:SOUND 1,200,8,15
740 GOSUB 800
750 SOUND 0,240,10,15:SOUND 1,240,8,15
760 GOSUB 800
770 SOUND 0,0,0,0:SOUND 1,0,0,0
780 NEXT T
790 RETURN
799 REM ### wait ###
800 FOR W=0 TO 100:NEXT W:RETURN
899 REM ##### main program #####
900 GOSUB 100:REM initialize #
910 FOR T=1 TO 11
920 POSITION 5,T:? #6;"############";
930 NEXT T
940 SCORE=SCORE+HP-TURN-5*K
950 HP=HP+20:TURN=0:PR=PR+0.1
960 XP=10:YP=1:POSITION XP,YP:? #6;"@[J]
@";
970 FOR T=0 TO 10
980 X(T)=5+T:Y(T)=8
990 NEXT T
1000 GOSUB 200:REM player   #
1010 GOSUB 700:REM eaten ?  #
1020 IF K>4 THEN 1070
1030 IF YP=11 THEN 910
1040 GOSUB 400:REM monsters #
1050 GOSUB 700:REM eaten ?  #
1060 IF K<=4 THEN 1000
1070 POSITION 1,11
1080 ? #6;"your @score@ is ";SCORE
1090 POSITION 0,0
1100 ? #6;"do you want again <ESC>[*]";
1110 POKE 764,255:GET #2,KEY
1120 IF KEY=89 THEN RUN
1125 IF KEY<>78 THEN 1110
1130 IF KEY<>78 THEN 1110
1140 GRAPHICS 0:END
```

Keyboard Memory

This game uses the computer keyboard to test your memory. Eighteen of the keys each conceal a character, in the same way that a playing card, lying face down, hides its value. There are nine different characters, each hidden by two keys.

The symbols of the keys you must concentrate on are shown on the screen.



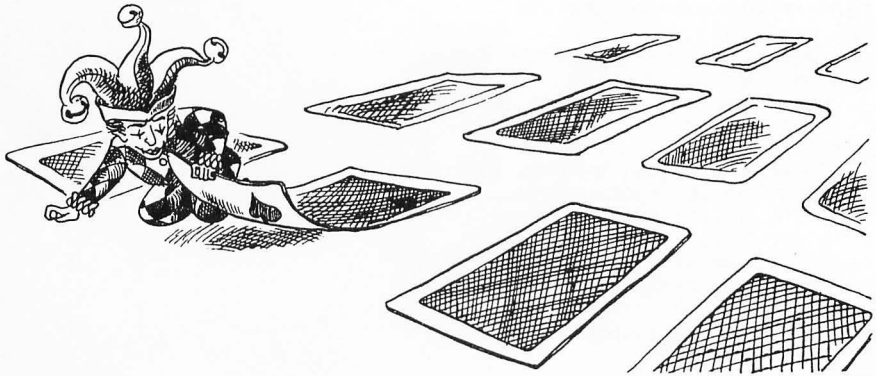
Press a key and the character it is hiding will be shown on the screen: press two, one after the other, and if they are both hiding the same character their symbols will disappear from the screen after you have pressed any other key.

Your first few attempts will obviously be guesses but by memorizing the positions of the characters they show you should soon be able to work out where the pairs are.

```
10 REM ******* Keyboard Memory *******
20 DIM KEY$(18),PIC$(18),KEYPIC$(18),KE$
(1),ASK(1,2)
30 OPEN #2,4,0,"K:"
40 GRAPHICS 18:POKE 756,226
50 SETCOLOR 0,0,0
60 GOTO 800
99 REM Initialize ***
```

```
100 ? #6;"  @keyboard@ @memory@    "
110 KEY$="qwertyasdfghzxcvbn"
120 PIC$="@E,,,,,,PPSSRRTTFFGG@@"
130 FOR KEY=1 TO 18
140 PICN=INT(18*RND(0)+1)
150 IF PIC$(PICN,PICN)<>" " THEN 190
160 PICN=PICN+1
170 IF PICN>18 THEN PICN=PICN-18
180 GOTO 150
190 KEYPIC$(KEY,KEY)=PIC$(PICN,PICN)
200 PIC$(PICN,PICN)=" "
210 NEXT KEY:RETURN
299 REM display keys ***
300 FOR I=0 TO 2:POSITION 4,2*I+4
320 FOR J=1 TO 6
330 C=6*I+J:? #6;KEY$(C,C);" ";
340 NEXT J:NEXT I:RETURN
399 REM input and show card ***
400 FOR NK=0 TO 1
410 GET #2,KEY:KE$=CHR$(KEY+32)
420 FOR I=1 TO 18
430 IF KE$=KEY$(I,I) AND (NK=0 OR I<>ASK
(0,0)) THEN 490
440 NEXT I
450 POSITION 4,11:? #6;"@WRONG@ @INPUT@"
460 SOUND 0,200,10,15:GOSUB 700
470 POSITION 4,11:? #6;"              "
480 GOTO 410
490 ASK(NK,2)=2*INT((I-1)/6)+4
500 ASK(NK,1)=2*(I-3*ASK(NK,2)+13)
510 ASK(NK,0)=I
520 POSITION ASK(NK,1),ASK(NK,2)
530 ? #6;KEYPIC$(I,I)
540 NEXT NK:RETURN
599 REM check cards ***
600 TURN=TURN+1
610 K1=ASK(0,0):K2=ASK(1,0)
620 IF KEYPIC$(K1,K1)<>KEYPIC$(K2,K2) TH
EN SOUND 0,100,10,15:GOSUB 700:RETURN
630 DONE=DONE+1
640 KEY$(K1,K1)=" ":KEY$(K2,K2)=" "
```

```
650 SOUND 0,40,10,15
660 GOSUB 700
670 IF DONE=9 THEN ? " Done in ";TURN;"
turns !":SOUND 0,10,10,15:GOSUB 700:END
680 RETURN
699 REM wait ***
700 FOR W=1 TO 200:NEXT W
710 SOUND 0,0,0,0
720 RETURN
799 REM *** main program ***
800 GOSUB 100:REM initialize      *
810 GOSUB 300:REM display keys    *
820 GOSUB 400:REM input and show  *
830 GOSUB 600:REM check           *
840 GOTO 810
```



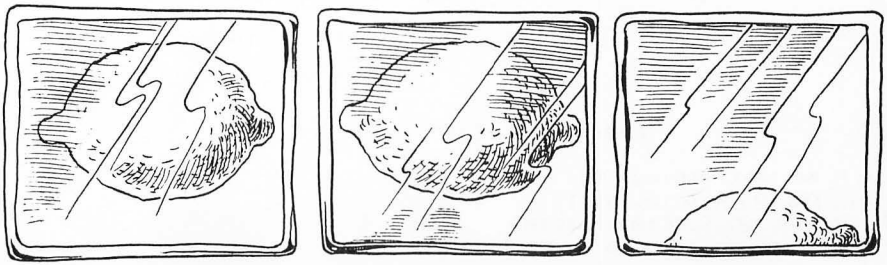
Las Vegas a Go Go

Have you ever watched someone pumping coin after coin into a one-armed bandit, and found yourself wondering what the fascination was? You will soon find out when this program turns your computer into a fabulous Las Vegas-style fruit machine. All the playing instructions you need will appear on the screen. At certain points you will be presented with a list of options, for instance

```
INSERT, HOLD, PLAY, OR END
```

Enter your choice by typing the first letter of the option you want, for instance P keeps your machine playing. The reels are numbered 1, 2, and 3. If you want to hold one or more reels, type in the appropriate number or numbers after you have entered H.

Lights will flash and music play as the wheels whiz around. Have you won this time? Never mind, you are sure to hit the jackpot sooner . . . or later!



```
10 REM ##### Las vegas a gogo #####
20 SETCOLOR 1,0,0:SETCOLOR 2,7,10
30 SETCOLOR 4,12,4:POKE 82,0:POKE 752,1
40 OPEN #2,4,0,"K:"
50 GOTO 2900
99 REM ### initialize ###
100 REM ## screen ##
110 DL=PEEK(560)+256*PEEK(561)+3
120 POKE DL,71:POKE DL+3,6:POKE DL+4,7
130 POKE DL+5,6:POKE DL+25,7
140 ? "<ESC><SHIFT><CLEAR>@% las vegas a
 gogo *@<ESC>[=]<ESC>[=]"
150 ? "                  [H]@      @[J]"
160 ? "                 [H]@        @[J]"
170 ? "                [H]@[QUUE]  [QUUE]@[J
]"
180 ? "               [H]@ @[Y]   @[Y] @[Y]
 @[Y] @[J] @[T]@"
190 ? "               @  [Z]@[UU]@[C] [Z]@[
UU]@[C]   @ @[T]@"
200 ? "               @ [H]@ @[J] [H]@ @[J]
 [H]@ @[J] @ @ @"
210 ? "               @ @    @ @    @ @    @ @
 @ @"
220 ? "               @[J]@[J] [H]@ @[J] [H
]@ @[J] [H]@[H]@ @ @"
230 ? "                 @[J]           [H]@   @
 @"
240 ? "                  @[J]         [H]@    @
 @"
250 ? "                   @              @"
260 ? "                   @         @"
270 ? "                   @         @"
280 ? "                   @         @"
290 ? "                   @         @"
300 ? "                   @         @"
310 ? "                  [H]@         @[J]"
320 ? "                 @             @"
330 REM ## text-strings ##
340 DIM B$(40),R1$(1),IN$(1)
350 DIM S1$(10),S2$(10),S3$(10)
360 REM ## wheel figures & values ##
```

```
370 DIM WF$(13),R(2,13),V(4),HV(3)
380 FOR V=1 TO 13
390 READ R1$,R1,R2:WF$(V)=R1$
400 R(0,V)=0:R(1,V)=R1:R(2,V)=R2
410 NEXT V
420 DATA A,0,0,K,0,0,Q,0,0,J,0,0,#,8,40
430 DATA *,8,40,$,7,30,[,],5,25,[;],3,20
440 DATA [.],3,20,[P],2,10,[S],2,10,@,0,
0
450 REM ## gamble pos. & values ##
460 DIM GP(4,1),GV(4)
470 FOR I=1 TO 4:READ R1,R2
480 GP(I,0)=R1:GP(I,1)=R2
490 NEXT I
500 DATA 33,3,35,5,33,7,30,5
510 REM ## holds, nohold ##
520 DIM HF$(3),NH$(1):HF$="    ":NH$="@ @
"
530 REM ## jokers ##
540 DIM JP(4),JF$(4),JC(4)
550 FOR JO=1 TO 4:JC(JO)=0
560 READ R1,R1$:JP(JO)=R1:JF$(JO)=R1$
570 NEXT JO:NJ=4
580 DATA 15,a,16,k,20,9,21,J
```

```
590 REM ## jackpot ##
600 DIM JA(13)
610 FOR V=1 TO 13:JA(V)=0:NEXT V
620 JA(5)=1:JA(6)=1:JA(7)=1
630 REM ## points ##
640 POSITION 1,1:? "CREDIT:0";
650 POSITION 1,11:? "--@--= 1";
```

```
660 FOR V=12 TO 5 STEP -1
670 ? :? " -";WF$(V,V);WF$(V,V);"- = ";R
(1,V);
680 POSITION 29,24-V:? WF$(V,V);WF$(V,V)
;WF$(V,V);" = ";R(2,V);
690 IF JA(V) THEN ? "+J";
700 NEXT V
710 RETURN
799 REM ### wait ###
800 FOR WW=1 TO W:NEXT WW:RETURN
899 REM ### get input, print s3$ ###
900 FOR W=0 TO 40 STEP 2
910 SOUND 0,W,10,VOL
920 IF PEEK(764)<>255 THEN GET #2,IN:IN$
=CHR$(IN):POSITION X,Y:? S3$;:SOUND 0,0,
0,0:W=99:NEXT W:POP :RETURN
930 NEXT W:RETURN
999 REM ### blink s1$/s2$ and get
        input and end with s3 ###
1000 POSITION X,Y:? S1$;:GOSUB 900
1010 POSITION X,Y:? S2$;:GOSUB 900
1020 GOTO 1000
1099 REM ### add increment to money ###
1100 FOR AD=SGN(IN) TO IN STEP SGN(IN)
1110 SOUND 0,3210-30*SGN(IN)-2*AD,10,15
1120 POSITION 8,1:? MO+AD;
1130 IF MO+AD<100 THEN ? "  ";
1140 FOR W=1 TO 8:SOUND 1,60-5*W,10,10
1150 NEXT W:SOUND 1,0,0,0
1160 NEXT AD:MO=MO+IN:SOUND 0,0,0,0
1170 RETURN
1199 REM ### remove double ###
1200 DO=0:POSITION 26,0:? "         ";
1210 FOR JO=1 TO 4:POSITION JP(JO),6
1220 ? JF$(JO,JO);:JC(JO)=0
1230 NEXT JO:NJ=4:RETURN
1299 REM ### print b$ at bottom ###
1300 POSITION 0,21:? "<ESC><SHIFT><BACK
S>";
1310 POSITION INT(10-LEN(B$)/2),21
1320 ? B$;:RETURN
1399 REM ### jackpot ###
```

```
1400 JA=1:JC=15:POSITION 9,11:? "0";
1410 FOR TI=1 TO 10:POSITION 20,1
1420 ? "@JAC:@                    ";
1430 POSITION 24,1
1440 FOR C=1 TO JC:? "j";
1450 SOUND 0,150-10*TI-2*C,10,15
1460 NEXT C:NEXT TI:SOUND 0,0,0,0
1470 RETURN
1499 REM ### adapt variables ###
1500 LW=( NOT HB)*WI:WI=0
1510 FOR HO=1 TO 3:HF$(HO,HO)=NH$
1520 POSITION 12+3*HO,12:? NH$;
1530 NEXT HO
1540 IF DO THEN 1620
1550 FOR JO=1 TO 4
1560 IF JC(JO)=0 THEN 1610
1570 JC(JO)=JC(JO)-1
1580 IF JC(JO)>0 THEN 1610
1590 NJ=NJ+1
```

```
1600 POSITION JP(JO),6:? JF$(JO,JO);
1610 NEXT JO:GOTO 1630
1620 DC=DC-1:IF DC=0 THEN GOSUB 1200
1630 IF  NOT JA THEN RETURN
1640 JC=JC-1:SOUND 0,250,10,15
1650 W=10:GOSUB 800
1660 POSITION JC+24,1:? " ";
1670 IF JC=0 THEN JA=0:POSITION 20,1:? "
    ";:POSITION 9,11:? " ";:REM 4+1 sp.
1680 SOUND 0,0,0,0:RETURN
1699 REM ### not enough money ###
1700 HB=1:GOSUB 1200
1710 JA=0:POSITION 20,1:? "
      ";:REM 19 spaces.
```

```
1720 B$="INSERT OR END":GOSUB 1300
1730 S1$="INSERT (1)":S2$="@INSERT (1)@"
1740 S3$="            ":X=10:Y=1:VOL=15
1750 GOSUB 1000:IF IN$="P" THEN 1750
1760 RETURN
1799 REM ### hold possible ###
1800 B$="INSERT HOLD PLAY END"
1810 GOSUB 1300:X=15:Y=12:VOL=4
1820 S1$="@ @<ESC>[*]<ESC>[*]@ @<ESC>[*]
<ESC>[*]@ @":S2$="H<ESC>[*]H<ESC>[*]H<ESC
>[*]H<ESC>[*]H"
1830 FOR HO=1 TO 3:I=3*HO-2
1840 S1$(I,I)=HF$(HO,HO):NEXT HO
1850 S3$=S1$:GOSUB 1000:HO=IN-48
1860 IF HO<1 OR HO>3 THEN RETURN
1870 NH=NH+1:IF HF$(HO,HO)=NH$ THEN HF$(
HO,HO)="@H@":GOTO 1830
1880 HF$(HO,HO)=NH$:GOTO 1830
1899 REM ### no hold ###
1900 B$="INSERT, PLAY OR END"
1910 GOSUB 1300
1920 S1$="???":S2$="@???@":S3$="@   @"
1930 X=17:Y=19:VOL=2:GOSUB 1000
1940 RETURN
1999 REM # what to do with winnings ###
2000 B$="":IF  NOT HB THEN B$="HOLD "
2010 B$(LEN(B$)+1)="GAMBLE COLLECT"
2020 GOSUB 1300
2030 S1$="???":S2$="@???@":S3$="@   @"
2040 X=17:Y=19:VOL=2:GOSUB 1000
2050 RETURN
2099 REM ### spin gamble wheels ###
2100 NT=NT+1:HD=0
2110 FOR WD=1 TO 3:IF HF$(WD,WD)=NH$ THE
N POSITION 10+4*WD,9:? " ";:NEXT WD
2120 FOR WD=1 TO 3
2130 SOUND 0,183+24*WD,12,10
2140 IF HF$(WD,WD)<>NH$ THEN HD=1:GOTO 2
250
2150 FI=INT(RND(0)*100+1)
2160 IF FI<5 THEN V(WD)=FI:GOTO 2180
2170 V(WD)=5+(FI>7)+(FI>10)+(FI>13)+(FI>
```

```
23)+(FI)36)+(FI)49)+(FI)68)+(FI)87)
2180 FOR W=1 TO 5*WD
2190 IF W=WD THEN SOUND 1,0,0,0
2200 FOR WS=WD TO 3
2210 IF WF$(WS,WS)=NH$ THEN I=INT(13*RND
(0)+1):POSITION 10+4*WS,9:? WF$(I,I);
2220 NEXT WS:NEXT W:SOUND 1,100,10,15
2230 POSITION 10+4*WD,9
2240 ? WF$(V(WD),V(WD));
2250 NEXT WD:W=30:GOSUB 800
2260 SOUND 0,0,0,0:SOUND 1,0,0,0
2270 RETURN
2299 REM ### take care of jokers ###
2300 JG=0
2310 FOR WD=1 TO 3:V=V(WD)
2320 IF V>4 THEN FV=V:GOTO 2360
2330 JW=WD:JG=JG+1:IF JC(V))0 THEN 2360
2340 POSITION JP(V),6:? " ";
2350 JC(V)=20:NJ=NJ-1
2360 NEXT WD
2370 IF  NOT DD THEN DD=(NJ=0):IF DD THE
N DC=15:POSITION 26,0:? "double [A]";
2380 RETURN
2399 REM ### compute winnings ###
2400 HV=13:ON JG GOTO 2430,2410,2470
2410 IF  NOT (JA AND FV>10) THEN HV=FV
2420 GOTO 2470
2430 V(0)=V(3):V(4)=V(1)
2440 IF JW AND V(JW-1)<>V(JW+1) THEN 247
0
2450 V(0)=15:V(4)=15
2460 HV=V(JW+1-2*(V(JW+1))V(JW-1)))
2470 FOR WD=1 TO 3:HV(WD)=V(WD)
2480 IF V(WD)<5 THEN HV(WD)=HV
2490 NEXT WD
2500 IF HV(1)=HV(2) AND HV(2)=HV(3) AND
JA(HV(2)) THEN GOSUB 1400
2510 FOR WD=1 TO 3:WI=WI+(HV(WD)=13)
2520 NEXT WD:IF JW THEN WI=10*WI
2530 NS=(HV(1)=HV(2))+(HV(2)=HV(3)):WI=W
I+R(NS,HV(2))
2540 IF DD THEN WI=WI+WI
```

```
2550 RETURN
2599 REM ### gamble ###
2600 NG=NG+1:W=1:GV(1)=2*WI:GV(2)=0
2610 GV(3)=INT(3*WI/2):GV(4)=INT(WI/2)
2620 B$="STOP":GOSUB 1300
2630 POSITION 10,1:? "    @GAMBLE@";
2640 FOR LI=2 TO 8:POSITION 29,LI:? "@
       @";:NEXT LI:REM 10 inv-spaces.
2650 R=R+1:IF R>4 THEN R=1
2660 POSITION GP(R,0),GP(R,1):? GV(R);:S
OUND 0,80+10*R,10,15
2670 IF PEEK(764)<>255 AND IN$<>"S" THEN
 GET #2,IN:IN$=CHR$(IN)
2680 IF IN$<>"S" THEN 2700
2690 W=(1+RND(0))*W:GOSUB 800
2700 IF W<200 THEN POSITION GP(R,0),GP(R
,1):? "@    @";:GOTO 2650
2710 FOR LI=2 TO 8:POSITION 29,LI:? "
       ";:NEXT LI:REM 10 spaces.
2720 POSITION 14,1:? "       ";
2730 SOUND 0,0,0,0
2740 GW=GW+GV(R)-WI:WI=GV(R):RETURN
2799 REM ### end of game ###
2800 GRAPHICS 0:POKE 82,2
2810 ? "<ESC>[=]<ESC><TAB>AMOUNT OF MONE
Y :"
2820 ? "<ESC>[=]<ESC><TAB>Put in
   : ";NI
2830 ? "<ESC>[=]<ESC><TAB>Got back
   : ";MO/4
2840 ? "<ESC>[=]<ESC><TAB>Max at one tim
e  : ";MM/4
2850 ? "<ESC>[=]<ESC><TAB>won by gamblin
g  : ";GW/4
2860 ? "<ESC>[=]<ESC><TAB>Number of hold
s  : ";NH
2870 ? "<ESC>[=]<ESC><TAB>Number of gamb
les : ";NG
2880 ? "<ESC>[=]<ESC><TAB>Number of turn
s  : ";NT
2890 END
2899 REM ##### main program #####
2900 GOSUB 100:REM initialize #
```

```
2910 GOSUB 1500:REM adapt      #
2920 IF MO>MM THEN MM=MO
2930 ON 2*HB+(MO>1)+1 GOSUB 1700,1800,17
00,1900
2940 POSITION 12,1:? "         ";
2950 CA=(IN$="I")+2*(IN$="P")+3*(IN$="E"
)
2960 ON CA GOTO 2970,3000,2800:GOTO 2920
2970 NI=NI+1:IN=4:GOSUB 1100
2980 IF MO>MM THEN MM=MO
2990 GOTO 2920
3000 IN=-2:GOSUB 1100
3010 GOSUB 2100:REM spin wheels #
3020 GOSUB 2300:REM jokers      #
3030 GOSUB 2400:REM winnings    #
3040 HB=HD OR MO<2
3050 IF WI>LW THEN 3110
3060 IF  NOT HD THEN 2910
3070 POSITION 12,1:? "YOU LOST";
3080 SOUND 0,250,10,15
3090 W=30:GOSUB 800:SOUND 0,0,0,0
3100 GOTO 2910
3110 POSITION 12,1:? "WON:     ";
3120 POSITION 16,1:? WI;
3130 FOR DU=WI+1 TO 2 STEP -1
3140 SOUND 0,DU,10,15
3150 W=10:GOSUB 800
3160 NEXT DU
3170 GOSUB 2000:REM get instruction #
3180 CA=(IN$="H" AND  NOT HB)+2*(IN$="G"
)+3*(IN$="C")
3190 ON CA GOTO 2910,3210,3200:GOTO 3170
3200 HB=1:IN=WI:GOSUB 1100:GOTO 2910
3210 GOSUB 2600:REM gamble #
3220 HD=-1:LW=0:GOTO 3040
```

Parrot

PARROT is a very simple competition game. Play it on a rainy day with a friend (especially after your friend has had a few beers!).

At the start of the game, your computer impertinently asks you:

Yes, he is calling *you* a parrot! Press any key and the computer immediately displays a letter. You must enter the same letter as fast as you can. Since you copy each letter, you really are a parrot! The game is repeated for about 10 seconds, and then the computer shows your score.

How good are you as a parrot? Have a competition with your friends and try to win the title of 'Super-Duper Parrot'. (My record is 19 — I teach parrots!)

```
10 REM ############ parrot ############
20 DIM X$(1)
30 OPEN #2,4,0,"K:"
40 DL=PEEK(560)+PEEK(561)*256
50 POKE DL+13,7:POKE DL+16,7:POKE 82,0
60 ? "<ESC>[<]<ESC>[=]<ESC>[=]<ESC>[=]<E
SC><TAB><ESC><TAB> [H][E[] @[J]":? "<ESC
><TAB><ESC><TAB>[F][@[J]   @"
70 ? "<ESC><TAB><ESC><TAB>  @  [H][@"
80 ? :? ;"<ESC>[=][@are you ready parrot@
":? "<ESC>[=]<ESC><TAB><ESC><TAB><ESC><T
AB>?";
90 GET #2,X:? "<ESC><SHIFT><BACK S><ESC>
[-]<ESC>[-]<ESC><SHIFT><BACK S>    @the
parrot@";
100 POKE 19,0:POKE 20,0
110 X=INT(65+26*RND(0))
120 POSITION 22,10:? CHR$(X+32);" ";
130 FOR W=1 TO 20:NEXT W:SOUND 0,0,0,0
140 IF PEEK(20)+256*PEEK(19)>600 THEN 20
0
150 IF PEEK(764)=255 THEN 140
160 GET #2,G:? CHR$(G);
170 IF G<>X THEN SOUND 0,240,10,15:GOTO
130
180 C=C+1:SOUND 0,20,10,15
190 ? "<ESC><SHIFT><BACK S>";:POSITION 2
2,10:GOTO 110
```

```
200 ? :? "<ESC>[=]<ESC><TAB>Score : ";C;
" in 10 sec."
210 FOR W=100 TO 0 STEP -1
220 SOUND 0,W,10,15:NEXT W:SOUND 0,0,0,0
230 ? "<ESC>[=]<ESC><TAB>Do you want to
play again ";
240 INPUT X$:IF X$<>"N" THEN RUN
250 GRAPHICS 0:POKE 82,0:END
```

Kentucky Derby

Ladies and gentlemen, they are off! Red Arrow is off to a good start . . . Blondish Boy is giving his jockey some trouble . . . and there's the famous Spanish Lady, on the inside track. This is really first class excitement, and we've only just begun! Coming round the first bend . . . Mickey Finn has taken over the lead, hard-pressed by Speedy Gonzales . . .

The Kentucky Derby is a horse-racing game. There are not as many horses in our Kentucky Derby as there are in the real race — in fact, there are only three — but they have to run 10 gruelling laps (with computer horse-racing anything is possible)! At the start of the game the odds are given for each horse. Now you can place your bets. As long as you have money you can bet again and again — these horses never get tired.

After the RUN command we see:

```
YOU HAVE . . . . .
HOW MUCH DO YOU BET ON HORSE 1
HORSE: 1  2  3
ODDS : 2  2  3
BET   :
```

(odds may vary from race to race).

When you have run out of money, your computer cheekily tells you:

```
YOU'RE LIVING ABOVE YOUR MEANS
```

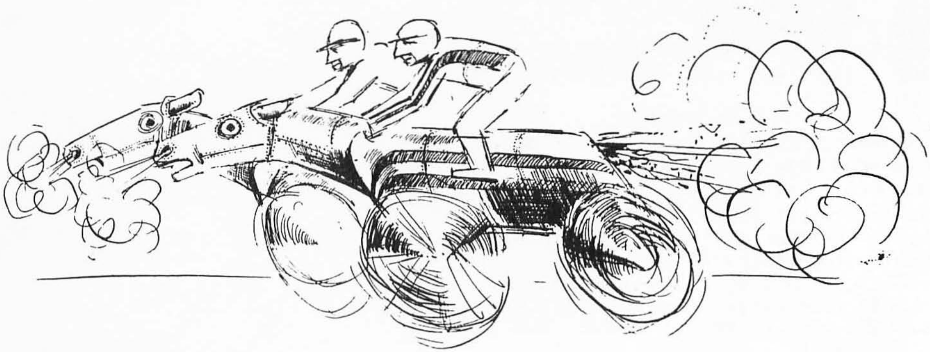
Immediately after you enter all your bets the screen displays the racetrack, with horses at the starting gates. The odds and bets are displayed in the lower part

of the screen.

I put my money on the bobtailed nag
Somebody bet on the bay

```
10 REM ######## Kentucky Derby ########
20 DIM R(3),V(3),X(3),T(3)
30 DIM D(3),P(3),I(3),S(3)
40 OPEN #2,4,0,"K"
50 GOTO 1200
99 REM ### initialize ###
100 VMIN=1:DV=1:VUPG=2:XMAX=19.5
110 ROUNDS=10:CREDIT=100:XH=10
120 XH1=XH-2.5:XH2=XH-0.8:XH3=XH+0.8
130 RETURN
199 REM ### initialize horses ###
200 WINNER=0
210 FOR K=1 TO 3
220 X(K)=0:R(K)=0:D(K)=0
230 P(K)=(INT(5*RND(0)+1)/5)
240 V(K)=P(K)*DV+VMIN:T(K)=2*K
250 NEXT K
260 RETURN
299 REM ### make bets ###
300 PTOT=0
310 FOR I=1 TO 3:P(I)=(V(I)-VMIN)/DV:PTO
T=PTOT+P(I):NEXT I
320 FOR K=1 TO 3:P(K)=INT(PTOT/P(K)):NEX
T K
330 GRAPHICS 2+16:SETCOLOR 1,4,6
340 SETCOLOR 2,13,6:SETCOLOR 3,12,10
350 POSITION 0,9
360 ? #6;"@HORSE:[    Q   R   S]@"
370 ? #6;"@ODDS :@":? #6;"@BET  :@";
380 POSITION 0,0
390 ? #6;"[JJJ] kentucky derby [JJJ]";
400 ? #6;"@     YOU CAN BET@";
410 FOR I=1 TO 3:POSITION 6+4*I,10:? #6;
P(I);:NEXT I
```

```
420 POSITION 0,6
430 ? #6;"HOW MUCH DO YOU BET"
440 ? #6;"ON HORSE   ?";
450 FOR J=1 TO 3:POSITION 0,3
460 ? #6;" YOU HAVE ";CREDIT;"$    "
470 ? #6;"
        ";:REM 38 spaces.
480 L=0:I(J)=0:POSITION 9,7:? #6;J;
490 POSITION 13,7:? #6;"       ";
500 POSITION 13,7
510 GET #2,X:IF X=155 THEN 550
520 X=X-48:IF X<0 OR X>10 THEN 510
530 ? #6;CHR$(X+16);
540 L=L+1:I(J)=I(J)*10+X:GOTO 510
550 IF CREDIT<I(J) THEN POSITION 2,4:? #
6;"@YOU'RE LIVING     ABOVE YOUR MEANS!@
":GOTO 480
560 CREDIT=CREDIT-I(J)
570 POSITION 7+4*J-L,11:? #6;I(J);
580 NEXT J:RETURN
599 REM ### change characterset ###
600 RESTORE :TOP=PEEK(106)-8
610 POKE 204,TOP:POKE 206,224
620 FOR X=1536 TO 1555
630 READ V:POKE X,V
640 NEXT X
650 Q=USR(1536)
660 DATA 104,162,4,160,0,177,205,145
670 DATA 203,200,208,249,230,206,230
680 DATA 204,202,208,242,96
690 RAMSET=TOP*256
700 FOR I=59 TO 63
710 FOR X=RAMSET+I*8 TO RAMSET+I*8+7
720 READ V:POKE X,V:NEXT X:NEXT I
730 DATA 4,7,7,132,124,124,68,130
740 DATA 4,7,12,60,255,208,64,96
750 DATA 4,7,6,60,60,38,68,129
760 DATA 60,60,62,126,126,127,255,255
770 DATA 0,0,0,0,0,0,0,255
780 POKE 756,TOP:RETURN
799 REM ### initialize screen ###
```



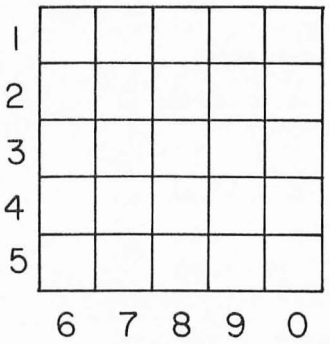
```
800 POSITION 0,7:? #6;"
   YOU HAVE ";CREDIT;"$.";
810 POSITION 0,1
820 FOR K=1 TO 3
830 ? #6;K;":                        ";
840 ? #6;"@[NNNNNNNNNNN]<ESC>[<BACK S>NNN
NNNNNN]@";
850 POSITION 2,T(K)-1:? #6;R(K)
860 POSITION X(K),T(K):? #6;"@[;]@"
870 NEXT K:RETURN
899 REM ### horse race ###
900 FOR Q=1 TO 3:SOUND 0,0,0,0
910 IF D(Q) THEN 1110
920 POSITION X(Q),T(Q)
930 IF S(Q)=0 THEN ? #6;"@[N]@"
940 IF S(Q) THEN ? #6;" "
950 IF S(Q)=1 THEN S(Q)=0:GOTO 1080
960 X(Q)=X(Q)+V(Q)
970 V(Q)=V(Q)+VUPG*RND(0)*(VM-V(Q))
980 IF S(Q)=2 THEN T(Q)=T(Q)+1:S(Q)=0:GO
TO 1080
990 IF (XH1<X(Q)) AND (X(Q)<XH3) THEN 10
60
1000 IF X(Q)<XMAX THEN 1080
```

```
1010 X(Q)=X(Q)-XMAX:R(Q)=R(Q)+1
1020 POSITION 2,T(Q)-1:? #6;R(Q);
1030 IF R(Q)<ROUNDS THEN GOTO 1080
1040 D(Q)=1:IF WINNER=0 THEN WINNER=Q
1050 GOTO 1080
1060 IF X(Q)<XH2 THEN S(Q)=1:GOTO 1080
1070 S(Q)=2:T(Q)=T(Q)-1
1080 SOUND 0,240,10,15
1090 POSITION X(Q),T(Q)
1100 ? #6;CHR$(251+S(Q));
1110 NEXT Q
1120 SOUND 0,0,0,0:RETURN
1199 REM ##### main program #####
1200 GOSUB 100:REM initialize          #
1210 GOSUB 200:REM initialize horses #
1220 GOSUB 300:REM make bets           #
1230 SOUND 0,20,10,15
1240 GOSUB 600:REM characterset        #
1250 GOSUB 800:REM initialize screen #
1260 VM=(V(1)+V(2)+V(3))/3
1270 GOSUB 900:REM horse race          #
1290 IF D(1)+D(2)+D(3)<3 THEN 1260
1300 CREDIT=CREDIT+P(WINNER)*I(WINNER)
1310 POSITION 0,8
1320 ? #6;"@horse @";WINNER;"@ wins  @"
1330 IF CREDIT THEN SOUND 0,100,10,15:FO
R W=1 TO 500:NEXT W:SOUND 0,0,0,0:GOTO 1
210
1340 ? #6:? #6;"YOU LOST ALL YOUR   MONE
Y  !!          ";
1350 GOTO 1350
```

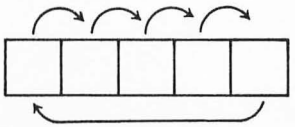
Rainbow Square Dance

A two-dimensional cube? Impossible of course, but this game, played on the computer screen, is similar in many ways to Rubik's Cube.

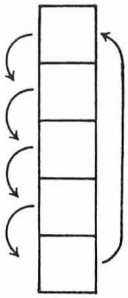
You will see 25 colored squares arranged at random on a 5 × 5 board. The rows of squares on the board are numbered like this:



That is, the horizontal rows are numbered 1 to 5 and the vertical rows 6, 7, 8, 9, and 0. When you enter the number of a row, the squares in that row are moved one position. In a horizontal row the squares move to the right, and in a vertical row they move downwards. As a square disappears off one end of a row it reappears at the other end, like this



in a horizontal row, or this

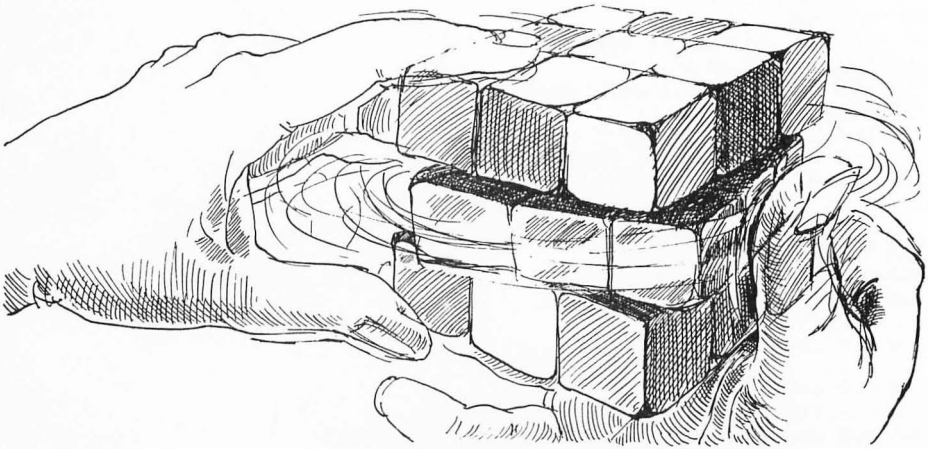


in a vertical row.

The object of all this? You must rearrange the squares to form five horizontal stripes of one color each. Sounds simple . . . but is it?

```
10 REM ##### Rainbow Square Dance #####
20 DIM A(5,5)
30 OPEN #2,4,0,"K"
40 GOTO 600
99 REM ### initialize ###
100 FOR I=0 TO 4:READ A
110 FOR J=0 TO 4:A(I,J)=A:NEXT J
120 NEXT I
130 DATA 3,35,163,0,131
199 REM ### initialize screen ###
200 GRAPHICS 17
210 ? #6;"@rainbow square dance@":? #6
220 ? #6;"   6  7  8  9  0":? #6
230 FOR I=1 TO 5
240 POSITION 0,3*I+2:? #6;I
250 POSITION 18,3*I+2:? #6;I;
260 NEXT I
270 POSITION 0,20:? #6;"   6  7  8  9  0
"
280 FOR I=0 TO 4:FOR K=0 TO 2
290 POSITION 2,4+3*I+K
300 FOR J=0 TO 4:FOR L=0 TO 2
```

```
310 ? #6;CHR$(A(I,J));
320 NEXT L:NEXT J:NEXT K:NEXT I
330 RETURN
399 REM ### shift row ###
400 FOR I=5 TO 1 STEP -1
410 A(A,I)=A(A,I-1)
420 NEXT I
430 A(A,0)=A(A,5)
440 FOR K=0 TO 2
450 POSITION 2,4+A*3+K
460 FOR J=0 TO 4:FOR L=0 TO 2
470 ? #6;CHR$(A(A,J));
480 NEXT L:NEXT J:NEXT K:RETURN
499 REM ### shift column ###
500 FOR I=5 TO 1 STEP -1
510 A(I,A)=A(I-1,A)
520 NEXT I
530 A(0,A)=A(5,A)
540 FOR I=0 TO 4:FOR K=0 TO 2
550 POSITION 2+3*A,4+I*3+K
```



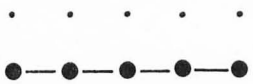
```
560 FOR L=0 TO 2:? #6;CHR$(A(1,A));:NEXT
 L
570 NEXT K:NEXT I:RETURN
600 REM ##### main program ###
610 LEV=30
620 GOSUB 100:REM initialize #
630 FOR W=1 TO LEV
640 A=INT(RND(0)*5)
650 ON INT(1+2*RND(0)) GOSUB 400,500
660 NEXT W
670 POSITION 0,23
680 ? #6;"enter row :  ";
690 POSITION 11,23
700 GET #2,A:? #6;CHR$(A);:A=A-48
710 IF ABS(A-4.5)>=5 THEN 670
720 IF A=0 THEN A=10
730 IF A>5 THEN A=A-6:GOSUB 500:GOTO 670
740 A=A-1:GOSUB 400
750 GOTO 670
```

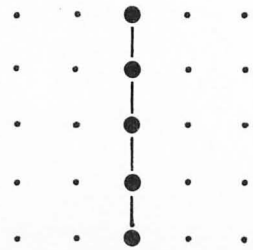
Qui Vive

To be 'on the qui vive' means to be alert and watchful. This game is called *QUI VIVE* because to play it successfully you must always be on the look out for winning situations. The rules are very simple but the program needed to implement them is anything but. In fact, it presents quite a challenge to a games programmer.

QUI VIVE was invented by Eugene de Wolf and is played on a square 5 × 5 board. Each player has five checkers and must try to arrange them into one of seven symmetrical patterns. These are:

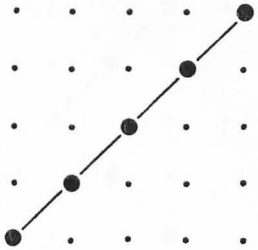
a horizontal line, e.g.



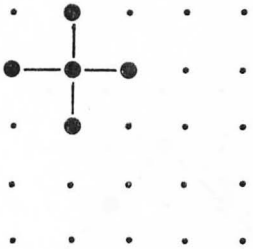


a vertical line, e.g.

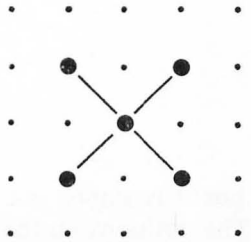
a diagonal, e.g.



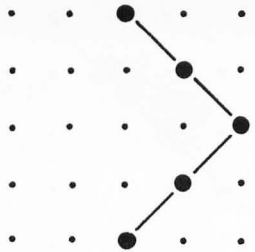
a rectangular cross, e.g.



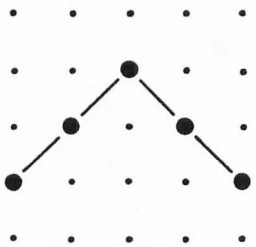
a skew cross, e.g.



a vertical wing, e.g.



a horizontal wing, e.g.



At the start of the game the board is empty and you and the computer take turns to place checkers on it. The positions on the board are labelled like this:

To place a checker at any particular position simply enter the appropriate coordinates: for instance for position 21 enter A1, for position 12, B3, etc. Your checkers are indicated by P and the computer's by C.

If all ten checkers have been placed on the board and no one has won, the game continues, with you and the computer taking turns to move checkers to try and gain a winning position. To move one of your checkers you enter the coordinates first of its present position and then of the position to which you wish to move it.

You must keep a look out for chances to make a winning pattern yourself, at the same time making sure that you are blocking any winning moves by the computer.

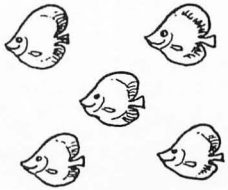
The program is one of the most interesting in this book. It contains a list of not only all 42 possible winning patterns but also over 100 particularly strong situations from which a player has a chance of making either one of two winning patterns. During the game the computer monitors this list in the light

of the situation on the board and assesses what its best move is.

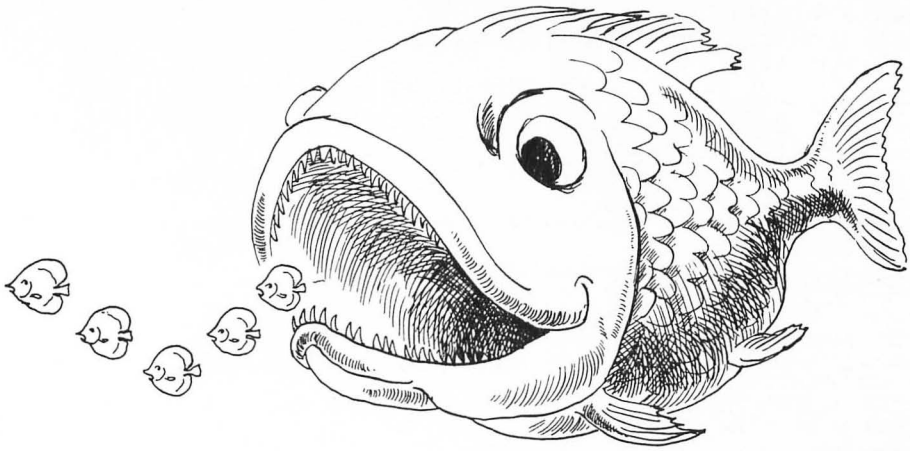
We advise you to play this game on an ordinary checkerboard against one of your friends before you take on the computer. You will soon see what a superb game it is.

```
10 REM ###### Q U I - U I V E ######
20 DIM SF(41,4),SFD(41),X$(27)
30 DIM DF(107,1),DFF(107,6),DFP(107)
40 DIM CC(4,1),B(24),CDF(30),PDF(30)
50 POKE 82,5:? "<ESC><CLEAR><ESC><ESC>@
Please wait @"
60 GOTO 600
99 REM ### compute points of CDF (Comput
er Double Figures) ###
100 P1=(B(DFF(DF,0)) OR B(DFF(DF,1)))+(B
(DFF(DF,2)) OR B(DFF(DF,3)))
110 P=(B(DFF(DF,4))+B(DFF(DF,5))+B(DFF(D
F,6)))/10
120 IF P1<2 AND P=3 THEN P=P1+2:RETURN
130 P=P1+P:RETURN
199 REM ### compute points of PDF (Playe
r Double Figures) ###
200 NGP1=0:NGP2=0:NGC1=0:NGC2=0
210 D1=B(DFF(DF,0)):D2=B(DFF(DF,1))
220 IF D1=1 OR D2=1 THEN NGP1=1
230 IF D1=10 OR D2=10 THEN NGC1=10
240 D1=B(DFF(DF,2)):D2=B(DFF(DF,3))
250 IF D1=1 OR D2=1 THEN NGP2=1
260 IF D1=10 OR D2=10 THEN NGC2=10
270 GV=B(DFF(DF,4))+B(DFF(DF,5))+B(DFF(D
F,6))
280 P=GV+NGP1+NGP2+NGC1+NGC2
290 IF GV=30 AND (NGC1=0 OR NGC2=0) THEN
 P=P-10
300 IF GV=3 AND (NGP1=0 OR NGC2=0) THEN
P=P-1
310 RETURN
399 REM ### determine strategic value of
 move ###
400 IF NCDF=0 THEN 490
410 MC=0:NMC=0:BEST=0
```

```
420 FOR CDF=0 TO NCDF-1
430 DF=CDF(CDF):GOSUB 100
440 IF P>4 THEN PP=CP:PT=CT:BEST=1:CDF=N
CDF:NEXT CDF:RETURN
450 IF P=MC THEN NMC=NMC+1
460 IF P>MC THEN MC=P:NMC=1
470 NEXT CDF
480 IF MC<MP OR (MC=MP AND NMC<NMP) THEN
 RETURN
490 IF NPDF=0 THEN 550
500 IF A=0 THEN PP=CP:PT=CT
510 FOR PDF=0 TO NPDF-1
520 DF=PDF(PDF):GOSUB 200
530 IF P=4 THEN PDF=9:NEXT PDF:RETURN
540 NEXT PDF
550 IF MP=MC AND NMP=NMC AND RND(0)<0.5
AND A=1 THEN RETURN
560 MP=MC:NMP=NMC:PP=CP:PT=CT:A=1
570 RETURN
599 REM ##### main program #####
600 GOSUB 2600:REM init-SF (Single Figur
es)
610 GOSUB 3300:REM init-DF (Double Figur
es)
620 GOSUB 3500:REM com-DFF (Double Figur
es Fields)
630 CC=0:PC=0
640 ? "Do you want to begin ";:INPUT X$
650 IF X$(1,1)="Y" THEN GOSUB 1300:GOTO
960
660 PP=12:GOTO 950
670 FOR I=0 TO 4:CC(I,1)=0:NEXT I
680 GOSUB 1800:REM owner-SF's          #
690 IF P=5 THEN ? "@you win@<ESC>[2J":GO
```

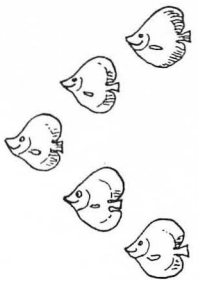


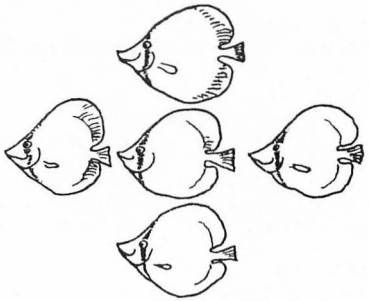
```
TO 1100
700 IF WSF(>-1 THEN GOSUB 1700:GOSUB 140
0:? "@1 win@<ESC>[2J":GOTO 1100
705 ? "@ My turn, please wait @"
710 GOSUB 2000:REM owner-OF's          #
720 GOSUB 2300:REM interesting OF's    #
730 A=0:BF=0:EF=24:MP=0:NMP=0
740 IF OM(>-1 THEN BF=OM:EF=OM
750 CP=BF-1
760 CP=CP+1:IF CP>EF THEN 950
770 IF B(CP)<>0 THEN 940
790 GOSUB 2400:REM find CDF            #
800 IF SK<9 AND A=1 THEN 940
810 B(CP)=10
820 IF CC=5 THEN 860
830 GOSUB 400:REM Strategic value      #
840 IF BEST THEN BEST=0:GOTO 950
850 GOTO 930
860 FOR CTI=0 TO 4
870 IF CC(CTI,1) THEN 920
880 CT=CC(CTI,0):B(CT)=0
890 GOSUB 400:REM Strategic value      #
900 IF BEST THEN BEST=0:CTI=9:NEXT CTI:G
OTO 950
910 B(CT)=10
920 NEXT CTI
930 B(CP)=0
940 GOTO 760
950 GOSUB 1400:REM computer move       #
960 GOSUB 1500:REM player move         #
970 GOTO 670
1099 REM ### again ? ###
1100 ? "Do you want to play again ";
1110 INPUT X$:IF X$="Y" THEN FOR I=0 TO
24:B(I)=0:NEXT I:GOTO 630
1120 END
1199 REM ### get input ###
1200 INPUT X$:IF LEN(X$)<>2 THEN 1240
1210 IF X$(1,1)<"A" OR X$(1,1)>"E" OR X$
(2,2)<"1" OR X$(2,2)>"5" THEN 1240
1220 X=ASC(X$(1,1))-5%VAL(X$(2,2))-40
1230 RETURN
```



```
1240 ? "<ESC>[2]Wrong input try again ";
1250 GOTO 1200
1299 REM ### show board ###
1300 FOR I=0 TO 4:? 5-I;
1310 FOR J=0 TO 4:K=I*5+J
1320 IF B(K)=0 THEN ? "0 0";
1330 IF B(K)=10 THEN ? "0[T]0";
1340 IF B(K)=1 THEN ? "[T]";
1350 NEXT J:? :NEXT I
1360 ? "<ESC>[-]<ESC>[-]<ESC>[-]<ESC>[*]
<ESC>[*]<ESC>[*]<ESC>[*]<ESC>[*]<ESC>[*]
<ESC>[*]I am    : 0[T]0"
1370 ? "<ESC>[*]<ESC>[*]<ESC>[*]<ESC>[*]
<ESC>[*]<ESC>[*]<ESC>[*]You are: [T]<ESC
>[=]":? " ABCDE":?
1380 RETURN
1399 REM ### computer move ###
1400 IF CC<5 THEN PT=CC:CC=CC+1:GOTO 146
0
1410 ? "I take up  : ";CHR$(65+PT-INT(PT
/5)*5);5-INT(PT/5)
1420 B(PT)=0
```

```
1430 FOR CTI=0 TO 4
1440 IF CC(CTI,0)=PT THEN PT=CTI:CTI=9
1450 NEXT CTI
1460 ? "I put down : ";CHR$(65+PP-INT(PP
/5)*5);5-INT(PP/5)
1470 B(PP)=10:CC(PT,0)=PP
1480 GOSUB 1300:RETURN
1499 REM ### player move ###
1500 IF PC<5 THEN 1540
1510 ? "Which checker do you take up ";
1520 GOSUB 1200:PT=X
1530 IF B(PT)<>1 THEN ? "<ESC>[2]Not pos
sible":GOTO 1510
1540 ? "Where do you put your checker ";
1550 GOSUB 1200:PP=X
1560 IF B(PP)<>0 THEN ? "<ESC>[2]Not pos
sible":GOTO 1540
1570 IF PC=5 THEN B(PT)=0
1580 IF PC<5 THEN PC=PC+1
1590 B(PP)=1
1600 GOSUB 1300:RETURN
1699 REM #### winning ###
1700 SF=WSF:GOSUB 2100:PP=OM
1710 FOR CTI=0 TO 4:CT=CC(CTI,0)
1720 FOR I=0 TO 4
1730 IF SF(SF,I)=CT THEN I=9:NEXT I:NEXT
 CTI:RETURN
1740 NEXT I:PT=CT
1750 CTI=9:NEXT CTI:RETURN
1799 REM ### determine owner of SF's ###
         ### and obligatory move (OM)###
         ### and checker pending (CC)###
1800 OM=-1:WSF=-1
1810 FOR SF=0 TO 41
1820 P=B(SF(SF,0))+B(SF(SF,1))+B(SF(SF,2
))+B(SF(SF,3))+B(SF(SF,4))
1830 IF P=0 OR P=10 OR P=20 OR P=30 THEN
 SFO(SF)=10:NEXT SF:RETURN
1840 IF P=40 THEN WSF=SF:NEXT SF:RETURN
1850 IF P>20 OR P<3 OR P=11 OR P=12 THEN
 SFO(SF)=0:NEXT SF:RETURN
1860 SFO(SF)=1
```





```
1870 IF P=4 AND OM=-1 THEN GOSUB 2100
1880 IF P=14 THEN GOSUB 2200
1890 IF P=5 THEN SF=99
1900 NEXT SF:RETURN
1999 REM ### determine owner of DF's ###
2000 NCDF=0:NPDF=0
2010 FOR DF=0 TO 107
2020 O1=SFO(DF(DF,0)):O2=SFO(DF(DF,1))
2030 IF O1<>O2 OR O1=0 OR O2=0 THEN DFP(
DF)=0:NEXT DF:RETURN
2040 GOSUB 200:DFP(DF)=P
2050 IF P=30 OR P=40 THEN CDF(NCDF)=DF:N
CDF=NCDF+1:NEXT DF:RETURN
2060 IF P=4 OR P=14 THEN PDF(NPDF)=DF:NP
DF=NPDF+1:NEXT DF:RETURN
2070 NEXT DF:RETURN
2099 REM ### determine obligatory ###
         ### move (OM)           ###
2100 FOR I=0 TO 4
2110 IF B(SF(SF,I))=0 THEN OM=SF(SF,I):I
=9
2120 NEXT I:RETURN
2199 REM ### determine pended ###
         ### checker (CC)    ###
2200 FOR I=0 TO 4
2210 IF B(SF(SF,I))=10 THEN F=I:I=9
2220 NEXT I
2230 FOR J=0 TO CC-1
2240 IF SF(SF,F)=CC(J,0) THEN CC(J,1)=1:
```

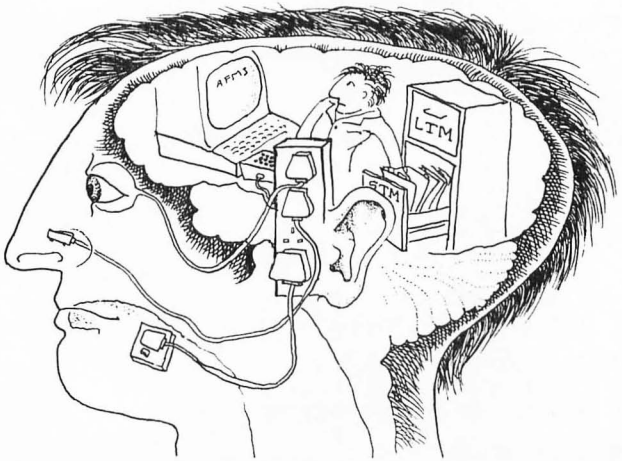
```
J=9
2250 NEXT J:RETURN
2299 REM determine interesting DF's ###
2300 IF NCDF>5 THEN RETURN
2310 I=20
2320 FOR DF=0 TO 107
2330 IF DFP(DF)<>I THEN NEXT DF:GOTO 237
0
2340 IF SFO(DF(DF,0))=10 AND SFO(DF(DF,1
))=10 THEN CDF(NCDF)=DF:NCDF=NCDF+1
2350 IF NCDF=6 THEN DF=108:NEXT DF:RETUR
N
2360 NEXT DF
2370 IF I=0 THEN RETURN
2380 I=I-10
2390 GOTO 2320
2399 REM ### find CDF ###
2400 IF OM()-1 THEN SK=9:RETURN
2420 IF NCDF=0 THEN 2470
2430 FOR CDF=0 TO NCDF-1:DF=CDF(CDF)
2440 FOR SK=0 TO 6
2450 IF DFF(DF,SK)=CP THEN SK=9:CDF=NCDF
:NEXT SK:NEXT CDF:RETURN
2460 NEXT SK:NEXT CDF
2470 IF NPDF=0 THEN SK=0:RETURN
2490 FOR PDF=0 TO NPDF-1:DF=PDF(PDF)
2500 FOR SK=0 TO 6
2510 IF DFF(DF,SK)=CP THEN SK=9:PDF=NPDF
2520 NEXT SK:NEXT PDF:RETURN
2599 REM ### initialize SF ###
2600 FOR I=0 TO 41:READ X$
2610 FOR J=1 TO 5
2620 SF(I,J-1)=ASC(X$(J,J))-65
2630 NEXT J:NEXT I:RETURN
2699 REM ### wings (^U)()   ###
2700 DATA UQMSY,PLHNT,KGCIO
2710 DATA AGMIE,FLRNJ,KQWSD
2720 DATA AGMQU,BHNRV,CIOSW
2730 DATA EIMSY,DHLRX,CGKQW
2799 REM ### times (X)      ###
2800 DATA ACGKM,BDHLN,CEIMO
2810 DATA FHLPR,GIMQS,HJNRT
```

```
2820 DATA KMQUW,LNRVX,MOSWY
2899 REM ### plus (+)       ###
2900 DATA BFGHL,CGHIM,DHIJN
2910 DATA GKLMQ,HLMNR,IMNOS
2920 DATA LPQRV,MQRSW,NRSTX
2999 REM ### diagonals (\/) ###
3000 DATA AGMSY,EIMQU
3099 REM ### columns (|)    ###
3100 DATA AFKPU,BGLQV,CHMRW
3110 DATA DINSX,EJOTY
3199 REM ### rows (-)       ###
3200 DATA ABCDE,FGHIJ,KLMNO
3210 DATA PQRST,UVWXY
3299 REM ### initialize DF ###
3300 FOR I=0 TO 107 STEP 12:READ X$
3310 FOR J=0 TO 11:FOR K=0 TO 1
3320 L=2*J+K+1
3330 DF(I+J,K)=ASC(X$(L,L))-49
3340 NEXT K:NEXT J:NEXT I:RETURN
3399 REM ### double figures ###
3400 DATA 171:1A1C1E1M101P2>202B2J
3410 DATA 393<3=3?3G474:4=4?4A4G40
3420 DATA 4P505B5D5J696<6C6E6M7=7A
3430 DATA 7C7I7O7P8>8B8D8J9?9E9K:?
3440 DATA :A:E:K:O:P;>;@;D;J<=<C<I
3450 DATA =G=I=O>F>H>J?G?K?P@F@J@L
3460 DATA AGAIAKAMAOAPBHBJBNCICMCP
3470 DATA DJDLDNEKEMEOFRFWGSGWHTHW
3480 DATA IRIXJSJXKTKXLRLYMSMYNTNY
3499 REM ### compute DFF ###
3500 FOR I=0 TO 24:B(I)=0:NEXT I
3510 FOR DF=0 TO 107:CF=4:NCF=0
3520 FOR I=0 TO 4:B(SF(DF(DF,0),I))=1:NE
XT I
3530 FOR I=0 TO 4:SF=SF(DF(DF,1),I)
3540 IF B(SF) THEN DFF(DF,CF)=SF:CF=CF+1
:B(SF)=0:NEXT I:GOTO 3560
3550 DFF(DF,NCF)=SF:NCF=NCF+1:NEXT I
3560 FOR I=0 TO 4:SF=SF(DF(DF,0),I)
3570 IF B(SF) THEN DFF(DF,NCF)=SF:NCF=NC
F+1:B(SF)=0
3580 NEXT I:NEXT DF:RETURN
```

STM

The nature of memory is certainly one of the most interesting topics in psychology. To remember something has been defined as 'to show in present responses some signs of earlier learned responses.' Why, though, do we remember some things and forget others? We may recognize someone we haven't seen in years, or call to mind a tune having heard only a few notes of it, but find it very difficult to remember, for instance, a telephone number. There are in fact two distinct types of memory: short term memory (STM) and long term memory (LTM). If you want to find out more about this intriguing topic we recommend that you read 'Introduction to Psychology' by Hilgard & Atkinson.

This game tests your short term memory. Play it with your friends and family and see who can remember most. You will see this 'menu' on the screen:



DO YOU WANT TO PLAY WITH
1. LETTERS?
2. NUMBERS?
3. 0 OR 1 ONLY?
4. OR DO YOU WANT TO STOP?

Now you make your choice by entering 1, 2, 3 or 4. Say you entered 1 to play with letters. A letter will now appear on the screen but only for a very short time. You have to enter that letter. The computer will then show you two letters which you have to enter, then three, and so on. Obviously, as the number of letters increases, remembering them all becomes more difficult. What is the longest string of characters you can remember? Can you beat our record of eight?

```
10 REM ############ S T M ############
20 DIM B$(20),A$(20)
30 GRAPHICS 0
40 ? "<ESC>{=J<ESC><TAB>Do you want to p
lay with"
50 ? "<ESC>{=J<ESC><TAB>1. Letters ?"
60 ? "<ESC>{=J<ESC><TAB>2. Numbers ?"
70 ? "<ESC>{=J<ESC><TAB>3. 0 OR 1 only ?
"
80 ? "<ESC>{=J<ESC><TAB>4. Or do you wan
t to stop ?"
90 ? "<ESC>{=J<ESC><TAB>";:INPUT A
100 ON A GOTO 110,120,130,350:GOTO 90
110 A=26:B=65:GOTO 140
120 A=10:B=48:GOTO 140
130 A=2:B=48
140 GRAPHICS 2:POKE 19,0:POKE 20,0
150 ? #6;" *****  s t m  *****"
160 C=1
170 A$=""
180 FOR D=1 TO C
190 B$=CHR$(INT(B+RND(0)*A))
```

```
200 POSITION 10,4:? #6;B$;
210 A$(LEN(A$)+1)=B$
220 FOR W=1 TO 100:NEXT W
230 POSITION 10,4:? #6;" ";
240 NEXT D
250 ? "Answer :";:INPUT B$
260 IF B$=A$ THEN C=C+1:GOTO 170
270 POSITION 0,4
280 ? #6;"wrong "
290 ? #6;"IT WAS : ";A$
300 ? #6;"SCORE  : ";C-1
310 ? #6;"TIME   : ";INT((PEEK(19)*256+P
EEK(20))/0.6)/100;" SEC."
320 ? "Do you want to play again ";:INPU
T A$
330 IF A$="Y" THEN 140
340 RUN
350 GRAPHICS 0:END
```

One to Five

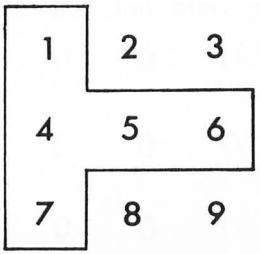
This exasperating game, which we saw first on a CASIO calculator, takes quite some logical reasoning to solve. It is played on a board with nine positions set out like this:

1 2 3

4 5 6

7 8 9

A horizontal and a vertical line drawn through any one position will cross five positions in all. For instance lines drawn through position 4 will also cross positions 1, 5, 6, and 7.



The positions on the board are occupied by a random pattern of digits between 0 and 5. If you enter one of the positions (using the key shown on the screen) the digit on that position, and all the digits on the horizontal and vertical lines going through that position, are increased by 1 (except 5 which becomes 0). For instance if we had

2 2 4

1 5 2

1 5 4

and entered 4, the board would change to

3 2 4

2 0 3

2 5 4

The puzzle is solved when the board looks like this:

0 0 0

0 0 0

0 0 0

```
10 REM ######### One To Five #########
20 DIM IL$(9),B(2,2),IN$(1)
30 OPEN #2,4,0,"K:"
40 GRAPHICS 18
50 IL$="QWEASDZXC"
60 FOR R=0 TO 2:FOR C=0 TO 2
70 B(R,C)=0:NEXT C:NEXT R
80 GOTO 600
99 REM ### screen ###
100 ? #6;"*   one to five   *"
110 ? #6;"  /----+----+----\":GOSUB 200
120 ? #6;"  +----+----+----+":GOSUB 200
130 ? #6;"  +----+----+----+":GOSUB 200
140 ? #6;"  \----+----+----/"
150 FOR R=0 TO 2:FOR C=0 TO 2
160 POSITION 3+5*C,2+3*R
170 ? #6;IL$(3*R+C+1,3*R+C+1);
180 NEXT C:NEXT R
190 GOSUB 500:RETURN
200 ? #6;"  !    !    !    !"
210 ? #6;"  !    !    !    !"
220 RETURN
299 REM ### get instruction ###
300 GET #2,IN:IN$=CHR$(IN+128)
310 FOR IN=1 TO 9
320 IF IN$=IL$(IN,IN) THEN RETURN
330 NEXT IN:GOTO 300
399 REM ### execute move ###
400 R=INT((IN-1)/3):C=IN-3*R-1
410 B(R,C)=B(R,C)-1
420 FOR I=0 TO 2
430 B(R,I)=B(R,I)+1
440 IF B(R,I)=6 THEN B(R,I)=0
450 B(I,C)=B(I,C)+1
460 IF B(I,C)=6 THEN B(I,C)=0
470 NEXT I:RETURN
499 REM ### output board ###
500 NZ=0
510 FOR R=0 TO 2:FOR C=0 TO 2
520 POSITION 5+5*C,3+3*R
530 ? #6;CHR$(B(R,C)-48+64);
```

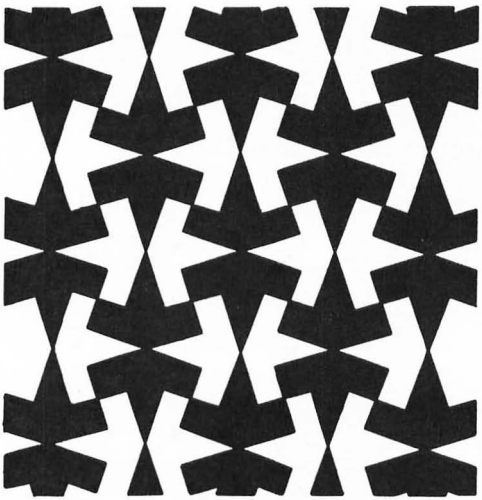
```
540 NZ=NZ+(B(R,C)=0)
550 NEXT C:NEXT R:RETURN
599 REM ##### main program #####
600 FOR TI=1 TO 9
610 IN=INT(9*RND(0)+1)
620 GOSUB 400:REM execute move    #
630 NEXT TI
640 GOSUB 100:REM screen          #
650 GOSUB 300:REM get instruction #
660 GOSUB 400:REM execute move    #
670 NM=NM+1
680 GOSUB 500:REM output board    #
690 IF NZ<9 THEN 650
700 POSITION 0,11:? #6;"@done in @";NM;"
@ moves@";
710 GOTO 710
```

Escher

The work of the Dutch graphic artist M.C. Escher (1902- 1972), based as it is on symmetry and mathematical forms, appeals particularly to computer programmers.

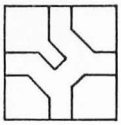
Now with the help of the computer you can produce your own patterns based on the same principles used by Escher.

Consider the following pattern:

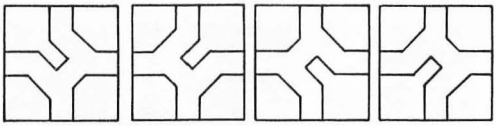


It is based on a mosaic in the Alhambra palace in Spain. If you look closely you can see that all the tiles are the same shape and that they are arranged so that the 'inverse video' of the white tiles shows the same pattern rotated

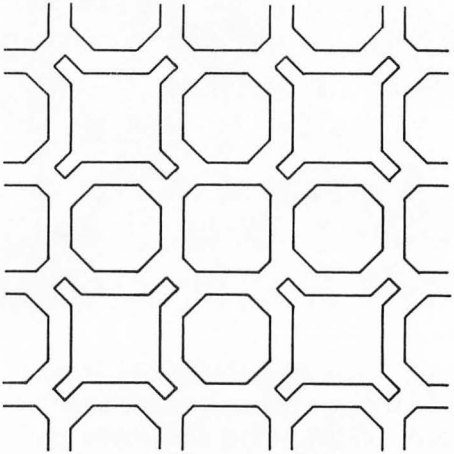
through 90 degrees. Escher studied such patterns and invented a game based on them. He devised a patterned tile such as



and arranged to form a continuous pattern.



which he rotated through four positions



This program does something very similar. After you have filled a 'basic' square with a pattern the computer rotates and shifts it to fill the screen with your own 'Escher'.

First the computer asks for the graphic mode:

Graphic modes:
1. very LOW resolution
2. LOW resolution
3. HIGH resolution
4. very HIGH resolution

and you enter your choice, for instance 2.

The the computer asks you to enter the size of your basic square, which you then have to fill, row by row, with colored pixels. The size of the pixels depends on the resolution you have chosen. Each key of the keyboard represents one colored pixel. You can delete the last entered pixel with the backspace key.

After you have filled your basic square the computer asks:

Enter length of board (max . . . .):?

and

Enter width of board (max . . . .):?

Now you fill the board with basic squares by pressing the N, W, E or S-key. These keys indicate the direction of each basic square, as it is displayed on the screen. (N = north, W = west, E = east, S = south). If you want to change the dimensions of the board, press the START key!

```
10 REM ##### Escher Game #####
20 DIM X$(1)
30 OPEN #2,4,0,"K:"
40 GOTO 900
99 REM ### initialize ###
100 GRAPHICS 0
110 ? "<ESC>[=]<ESC><TAB>E S C H E R   G
 A M E"
120 ? "<ESC>[=]<ESC>[=]<ESC><TAB>Graphic
s modes :"
130 ? "<ESC>[=]<ESC><TAB>1. very LOW res
olution"
140 ? "<ESC>[=]<ESC><TAB>2. LOW resoluti
on"
150 ? "<ESC>[=]<ESC><TAB>3. HIGH resolut
ion"
160 ? "<ESC>[=]<ESC><TAB>4. very HIGH re
solution"
170 ? "<ESC>[=]<ESC><TAB>Which mode do y
ou want :";
180 INPUT MO:IF MO=1 THEN 200
190 FOR I=5 TO 4*MO:READ X:NEXT I
200 READ MO:READ MC:READ MR:READ OF
210 ? :? "<ESC><TAB>Enter size of basic"
220 ? "<ESC><TAB>square (max ";INT((MR+1
)/2);") :";
230 INPUT SI
240 IF SI<1 OR SI>(MR+1)/2 THEN 210
250 DIM SQ(SI-1,SI-1)
260 DATA 3, 40,20,4, 5, 80, 46,2
270 DATA 7,160,95,1, 8,320,191,1
280 RETURN
299 REM ### initialize screen ###
300 DL=PEEK(560)+256*PEEK(561)+3
310 POKE DL,70:POKE DL+3,6
320 REM Define your own colors !
330 SETCOLOR 0,2,8
340 SETCOLOR 1,12,10
350 SETCOLOR 2,9,4
360 SETCOLOR 3,4,6
370 SETCOLOR 4,0,0
380 RETURN
```

```
399 REM ### input basic square ###
400 RESTORE
410 READ MOD:READ NC:READ NR:READ OFF
420 IF SI>(NR+1)/2 THEN 410
430 SC=(NC-SI)/2:SR=(NR-SI)/2
440 GRAPHICS MOD+16:GOSUB 300
450 FOR R=0 TO SI-1:RP=R+SR+OFF
460 POKE 87,1:POSITION 0,0
470 ? #6;"enter data for basicsquare ,ro
w ";R+1;"  ";
480 POKE 87,MOD
490 FOR C=0 TO SI-1:CP=C+SC:COL=1
500 COLOR 2*COL:PLOT CP,RP
510 FOR W=1 TO 25
520 IF PEEK(764)<>255 THEN 540
530 NEXT W:COL= NOT COL:GOTO 500
540 GET #2,X:X=X+208:IF X<>334 THEN 580
550 COLOR 0:PLOT CP,RP:C=C-2
560 IF C<-1 THEN R=R-1:RP=RP-1:C=SI-2
570 NEXT C
580 SQ(R,C)=X:COLOR X:PLOT CP,RP
590 NEXT C:NEXT R:RETURN
599 REM ### draw escher picture ###
600 ? "<ESC><CLEAR><ESC><ENTER>Enter lengt
h of board (max ";
610 ? INT(MC/SI);") :";:INPUT LE
620 IF LE<1 OR LE>MC/SI THEN 600
630 ? " Enter width of board (max ";
640 ? INT(MR/SI);") :";:INPUT WI
650 IF WI<1 OR WI>MR/SI THEN 630
660 SC=(MC-SI*LE)/2:SR=(MR-SI*WI)/2
670 GRAPHICS MO+16:GOSUB 300
680 POKE 87,2
690 ? #6;"@<JJ>@  @e s c h e r@  @<JJ>@"
700 ? #6;" draw with n<L>w<L>e<L>s<N>"
710 POKE 87,MO
720 FOR RO=OF+SR TO (WI-1)*SI+OF+SR STEP
 SI
730 FOR CO=SC TO (LE-1)*SI+SC STEP SI
740 IF PEEK(53279)=6 THEN 600
750 IF PEEK(764)=255 THEN 740
760 GET #2,X:X$=CHR$(X):X=0
```

```
770 X=(X$="N")+2*(X$="E")+3*(X$="S")+4*(
X$="W")
780 IF  NOT X THEN 740
790 FOR R=0 TO SI-1:FOR C=0 TO SI-1
800 COLOR SQ(R,C)
810 ON X GOSUB 860,870,880,890
820 NEXT C:NEXT R:NEXT CO:NEXT RO
830 POKE 87,1:POSITION 0,1
840 ? #6;"@[J]@  @d r a w i n g@  @[J]@"
850 RETURN
860 PLOT CO+C,RO+R:RETURN :REM <ESC>[-]
870 PLOT CO+SI-1-R,RO+C:RETURN :REM <ESC
>[*]
880 PLOT CO+SI-1-C,RO+SI-1-R:RETURN :REM
 <ESC>[=]
890 PLOT CO+R,RO+SI-1-C:RETURN :REM <ESC
>[+]
899 REM ##### main program #####
900 GOSUB 100:REM initialize          #
910 GOSUB 400:REM input basic square  #
920 GOSUB 600:REM draw escher picture #
930 IF PEEK(53279)<>6 THEN 930
940 POKE 764,255:GOTO 920
```

Genius at Work

Play this simple game to find out how good you are at thinking mathematically. After you have entered the number of players, the computer displays seven numbers and a larger 'target' number and asks

OPERATION?

You must pick two of the numbers and one of the four simple mathematical functions:

addition +
subtraction —
multiplication ×
division /

For instance, if you enter 7 + 18, the computer asks:

Is this your final result?

If you enter N, the computer calculates 7 + 18 = 25 and replaces the number 7 and 18 in the initial list by 25. Now you choose again. Your task is to get a final result as near to the target number as you can.

The character of this game is simplicity, so the whole family can play. When one player has finished their go, press any key to start again. Now you can find out if there is a genius in your family!

```
10 REM ######## Genius At Work ########
20 DIM D(7),F$(10),Z$(1)
30 T=0:X=0:R=0
```

```
40 GOTO 800
99 REM ### random numbers ###
100 FOR I=1 TO 7
110 D(I)=INT(10*RND(0)+1)
120 IF RND(0)<0.25 THEN D(I)=INT(4*RND(0
)+1)*25
130 NEXT I
140 F=INT((9*RND(0)+1)*100)
150 RETURN
199 REM ### display ###
200 ? "<ESC>[<CLEAR><ESC>[=]Try to appro
ach this number: ";F
210 ? "<ESC>[=]You have got these number
s:<ESC>[=]"
220 FOR I=1 TO 7
230 IF D(I)<>0 THEN ? D(I);" ";
240 NEXT I
250 ?
260 RETURN
299 REM ### input operation ###
300 ? "<ESC>[=]Enter operation ";:INPUT
F$
310 FOR N=1 TO LEN(F$)
320 E=ASC(F$(N))
330 IF (41<E) AND (E<48) THEN A=VAL(F$(1
,N-1)):B=VAL(F$(N+1,LEN(F$))):RETURN
340 NEXT N
350 GOTO 300
399 REM ### compute new numbers ###
400 FOR J=1 TO 7
410 IF D(J)=A THEN D(J)=0:GOTO 430
420 NEXT J:GOTO 460
430 FOR I=1 TO 7
440 IF D(I)=B THEN ON E-41 GOSUB 500,510
,0,520,0,530:VALF=D(I):RETURN
450 NEXT I:D(J)=A
460 ? "<ESC>[2J<ESC>[2J":X=0:POP :GOTO 8
10
500 D(I)=A*B:RETURN
510 D(I)=A+B:RETURN
520 D(I)=A-B:RETURN
530 D(I)=INT(A/B):RETURN
```

```
599 REM ### compute points ###
600 ? "<ESC>[=]Is this your final result
 ";
610 INPUT Z$
620 IF Z$<>"Y" THEN POP :GOTO 810
630 ? :? "You have: ";VALF;" on ";F
640 X=5-ABS(F-VALF)
650 IF X<0 THEN X=0
660 IF X=5 THEN ? "<ESC>[=]WELL DONE!"
670 T=T+X:R=R+1
680 ? "<ESC>[=]This makes ";X;" points"
690 ? "<ESC>[=]Now you have ";T;" points
";
700 ? " in ";R;" rounds"
710 ? "<ESC>[=]An average of ";INT(T*100
/R)/100;" points per round"
720 ? "<ESC>[=]Again ";:INPUT Z$
730 IF Z$="Y" THEN POP :GOTO 800
740 RETURN
799 REM ##### main program ###
800 GOSUB 100:REM random numbers       #
810 GOSUB 200:REM display              #
820 GOSUB 300:REM input operation      #
830 GOSUB 400:REM compute new numbers  #
840 GOSUB 600:REM compute points       #
850 ? "<ESC>[<CLEAR>]":END
```

Shark Hunt

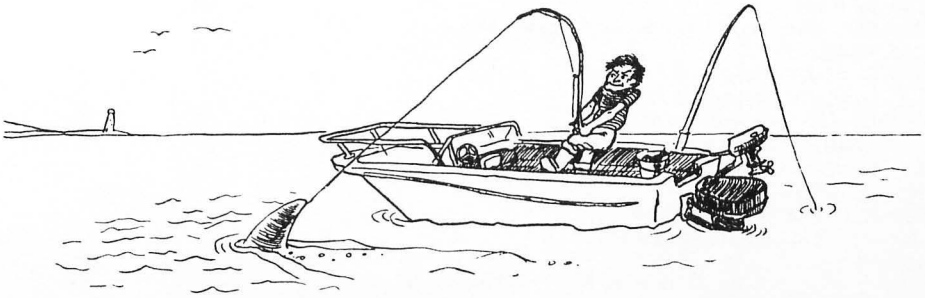
For five days now you have been adrift in your boat searching the seven seas. Suddenly you see a slight ripple on the mirror-like surface of the ocean — there it is, your quarry, that terror of the deep — the shark. Move your boat using the cursor keys or the joystick. Your echo-sounder will tell you how close you are to the shark — the nearer you get the higher the note it gives. Stray too far from the shark and you are told

SORRY; YOU WENT TOO FAR

When you think you are close enough you can take a shot at the shark by pressing the space bar or pressing the red button on your joystick. If you hit the target you will see the message

THAT'S IT! CONGRATULATIONS!

Unfortunately you only have 200 ergs of energy to use. Moving your boat and firing your gun both cost energy. If you use up all your energy the computer tells you

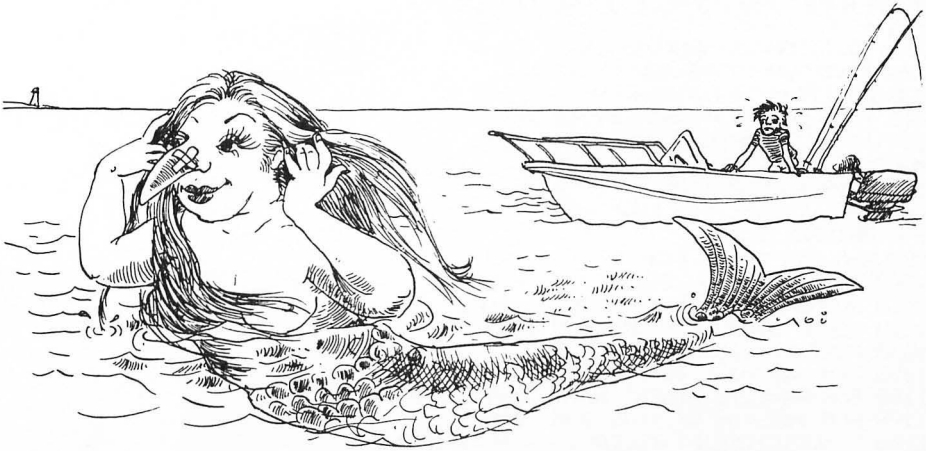


YOU RAN OUT OF ENERGY

At the end of the game you will be given information on your performance as a shark hunter.

```
10 REM ###### S h a r k   H u n t ######
20 DIM CO$(81),R$(1),X$(3),Y$(3)
30 DIM TU(3,8),STIK(9,1)
40 OPEN #2,4,0,"K:"
50 POKE 752,1
60 GOTO 1500
99 REM ### initialize ###
100 CO$="Sorry, you went too far    @Tha
t's it! Congratulations!@You ran out of
energy      "
110 X$="l r":Y$="U d"
120 FOR TU=0 TO 2:FOR NO=0 TO 7
130 READ R:TU(TU,NO)=R
140 NEXT NO:NEXT TU
150 DATA 204,193,204,193,204,210,204
160 DATA 193,89,125,147,176,193,204
170 DATA 218,218,24,21,18,15,12,9,6,3
180 FOR I=0 TO 9
190 READ X,Y:STIK(I,0)=X:STIK(I,1)=Y
200 NEXT I
210 DATA 3,3,3,-3,3,0,0,0,-3,3
220 DATA -3,-3,-3,0,0,0,0,3,0,-3
230 RETURN
299 REM ### initialize one game ###
300 NG=NG+1
310 SX=INT(70*RND(0))-35
320 SY=INT(70*RND(0))-35
330 DI=ABS(SX)+ABS(SY)
340 MD=DI:SD=DI:SH=0:H1=0:NS=0
350 EN=200:VX=0:VY=0
360 RETURN
399 REM ### screen ###
400 ? "<ESC>[<J<ESC>[=J"
410 ? "          @                    @
          @  S H A R K   H U N T  @"
```

```
415 ? "@                                  @"
420 ? "<ESC>[=]<ESC><TAB>Amount of energ
y: ";EN;" ergs"
430 POSITION 2,17
440 ? "Move by using the cursor keys or<
ESC><TAB><ESC><TAB><ESC><TAB><ESC><TAB>
    the joystick"
450 ? "Shoot by pressing the space bar o
r<ESC><TAB><ESC><TAB><ESC><TAB><ESC><TAB
>     the red button"
460 POSITION 16,13
470 ? "Direction";
480 POSITION 2,22
490 ? "Press <@START@> to start game";
500 IF PEEK(53279)<>6 THEN 500
510 POSITION 2,22:? "<ESC><SHIFT><BACK S
><ESC><SHIFT><INSERT>";
520 RETURN
599 REM ### input speed ###
600 VX=0:VY=0
610 IF PEEK(764)=255 THEN 670
620 GET #2,IN
```



```
630 VX=3*(IN=42)-(IN=43))
640 VY=3*(IN=61)-(IN=45))
650 SH=(IN=32)
660 RETURN
670 IN=STICK(0):IF IN<>15 THEN VX=STIK(I
N-5,0):VY=STIK(IN-5,1)
680 SH= NOT STRIG(0)
690 RETURN
699 REM ### compute distance etc. ###
700 IF  NOT SH THEN 750
710 NS=NS+1:EN=EN-2*DI
720 IF EN<=0 THEN EN=0:RETURN
730 HI=(DI*RND(0)<=1)
750 SX=SX-VX:SY=SY-VY
760 EN=EN-(VX<>0)-(VY<>0)-1
770 IF EN<=0 THEN EN=0:RETURN
780 SX=SX+SGN(SX):SY=SY+SGN(SY)
790 DI=ABS(SX)+ABS(SY)
800 IF DI<MD THEN MD=DI
810 RETURN
899 REM ### output ###
900 POSITION 20,9:? "@ @<ESC>[=]<ESC><[+
]<ESC>[+]@ @ @ @<ESC>[=]<ESC>[+]<ESC>[+]
@ @";
910 POSITION 20+SGN(VX),10
920 ? X$(SGN(VX)+2,SGN(VX)+2);
930 POSITION 20,10+SGN(VY)
940 ? Y$(SGN(VY)+2,SGN(VY)+2);
950 IF SH THEN GOSUB 1000
960 SOUND 0,DI,10,15
970 W=100:GOSUB 1100
980 POSITION 25,6:? EN;" ergs  ";
990 RETURN
999 REM ### shoot ###
1000 FOR I=220 TO 250 STEP 5
1010 SOUND 1,I,10,15:W=2:GOSUB 1100
1020 SOUND 1,0,0,0:W=5:GOSUB 1100
1030 NEXT I:RETURN
1099 REM ### wait ###
1100 FOR WW=1 TO W:NEXT WW:RETURN
1199 REM ### end of this game ###
1200 ? "<ESC><SHIFT><CLEAR><ESC>[=]";CO$
```

```
(27*CA+1,27*(CA+1));"<ESC>[=]"
1210 FOR NO=0 TO 7
1220 SOUND 0,TU(CA,NO),10,15
1230 W=25:GOSUB 1100
1240 NEXT NO:SOUND 0,0,0,0
1250 ? "Starting distance          : ";SD
;" m."
1260 ? "Number of shots            : ";NS
1270 ? "You approached within      : ";MD
;" m."
1280 IF HI THEN ? "You hit it from
     : ";DI;" m."
1290 RE=50-DI/2
1300 IF HI THEN RE=50+EN/4+SD/10
1310 RE=INT(RE)
1320 IF RE<0 THEN RE=0
1330 IF RE>100 THEN RE=100
1340 TR=TR+RE
1350 IF RE>MR THEN MR=RE
1360 ? "<ESC>[=]Rewards, on a scale of 1
00:"
1370 ? "This game                  : ";RE
1380 ? "Average over ";NG;" games      :
 ";INT(TR/NG)
1390 ? "Maximum                    : ";MR
1400 RETURN
1499 REM ##### main program #####
1500 GOSUB 100:REM initialize         #
1510 GOSUB 300:REM initialize 1 game #
1520 GOSUB 400:REM screen             #
1530 GOSUB 600:REM input              #
1540 GOSUB 700:REM computing          #
1550 GOSUB 900:REM output             #
1560 CA=3*(EN=0)+2*HI+(DI>100)-1
1570 IF CA<0 THEN 1530
1580 GOSUB 1200:POKE 764,255
1590 ? "<ESC>[=]Another game ? ";:GET #2
,IN
1600 IF IN=89 THEN 1510
1610 ? "<ESC>[<CLEAR>]":END
```

Shakespearian Shuffle

Shakespeare, one of the world's greatest writers, and chess, the king of games, are combined in this unusual puzzle.

Letters are arranged on a chessboard like this:

T	O		B	E		O	R
N	O	T		T	O		B
E		T	H	A	T		I
S		T	H	E		Q	U
E	S	T	I	O	N	.	
W	I	L	L	I	A	M	
S	H	A	K	E	S	P	E
A	R	E		I	6	0	3

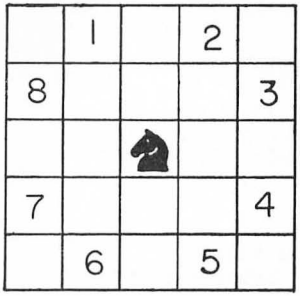
A chesspiece, the knight, then moves around the board as it would in a normal game of chess. As it jumps from one square to another, the letters or symbols on the squares are exchanged. In this way the text on the board is jumbled up. The computer will ask you

LEVEL?

and you enter the number of moves you want the knight to make. Obviously,

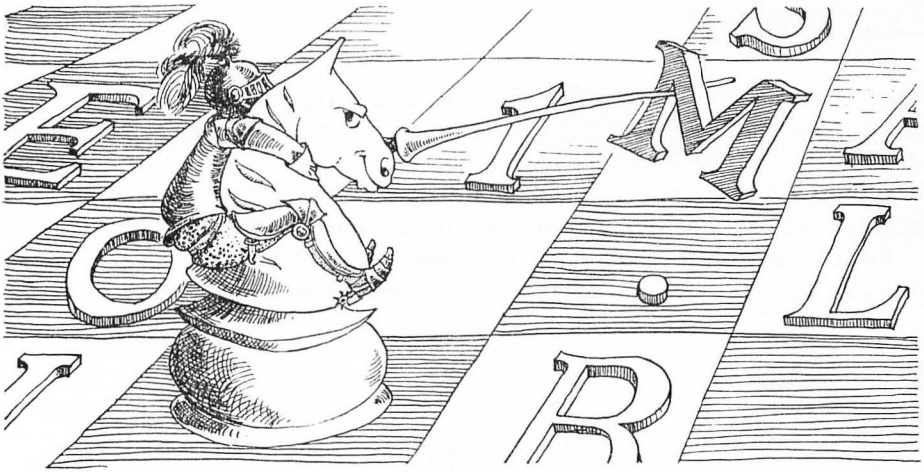
the higher this number the more mixed up the board becomes.

You will be shown the board after the knight has made his moves. Your task is to unscramble the text by moving the knight back around the board. Enter a number between 1 and 8 to move the knight, as shown.




```
10 REM ##### Shakespearian shuffle #####
20 DIM HX(7),HY(7),A$(64),B$(64),W$(1)
30 OPEN #2,4,0,"K:"
40 GOTO 500
99 REM ### initialize ###
100 XO=1:YO=1
110 READ B$:A$=B$
120 FOR K=0 TO 7
130 READ A,B:HX(K)=A:HY(K)=B
140 NEXT K
150 RETURN
199 REM ### knight's move ###
200 XS=HX(M):YS=HY(M)
210 IF XS+XO>8 OR XS+XO<1 OR YS+YO>8 OR
YS+YO<1 THEN M=INT(RND(0)*8):GOTO 200
220 XN=XO+XS:YN=YO+YS
230 RETURN
299 REM ### change letters ###
300 P=XN+(YN-1)*8:V=XO+(YO-1)*8
310 W$=B$(P,P):B$(P,P)=B$(V,V):B$(V,V)=W
$
```

```
320 POSITION 1+XO,1+YO
330 ? #6;B$((YO-1)*8+XO,(YO-1)*8+XO)
340 POSITION 1+XN,1+YN:? #6;"t";
350 XO=XN:YO=YN
360 RETURN
399 REM ### display ###
400 POSITION 0,0:? #6;"shakespeare shuff
le ";:? #6
410 FOR K=1 TO 8
420 ? #6;"  ";B$((K-1)*8+1,K*8)
430 NEXT K
440 POSITION 15,3:? #6;"8 1"
450 POSITION 14,4:? #6;"7   2"
460 POSITION 16,5:? #6;"@#@"
470 POSITION 14,6:? #6;"6   3"
480 POSITION 15,7:? #6;"5 4"
490 RETURN
499 REM ##### main program #####
500 ? "<ESC>[<CLEAR>]<ESC>[=]<ESC>[=]Ent
er level :";:INPUT L
510 GRAPHICS 18
520 GOSUB 100:REM initialize      #
530 GOSUB 400:REM display         #
540 FOR K=1 TO L
550 M=INT(RND(0)*8)
560 GOSUB 200:REM knight's move   #
570 GOSUB 300:REM change letters  #
580 NEXT K
590 POSITION 0,11:? #6;"enter move :
";
600 POSITION 12,11:GET #2,M:? #6;CHR$(M)
:M=M-49:IF M<0 OR M>7 THEN 590
610 GOSUB 200:REM knight's move   #
620 GOSUB 300:REM change letters  #
630 IF A$<>B$ THEN 590
640 POSITION 0,0:? #6;"congratulations!
 "
650 POSITION 0,10:? #6;"@do you want to
play again@   ?    ";
660 GET #2,M:IF M=89 THEN RUN
670 END
680 DATA @to be ornot to  be[L] thatis t
```



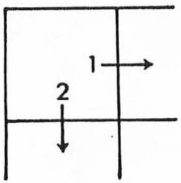
```
he question[N]william shakespeare [QWPS]
@
690 DATA 1,-2,2,-1,2,1,1,2,-1,2,-2,1,-2,
-1,-1,-2
```

Explosion

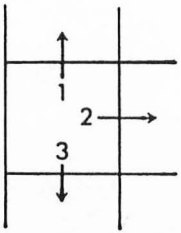
Most computer versions of *EXPLOSION* set out the board for two or more opponents to play on. With this program you are pitted against the computer itself. Are you up to the challenge?

EXPLOSION is played on a board of 3 × 3 or 4 × 4 squares. The computer will ask you to enter the size of the board you want.

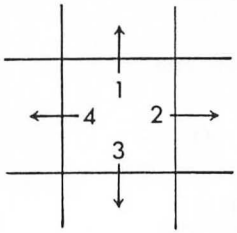
Each square on the board has a capacity equal to the number of squares directly adjoining it. This means that corner corner squares have a capacity of 2



edge squares have a capacity of 3



and central squares have a capacity of 4.

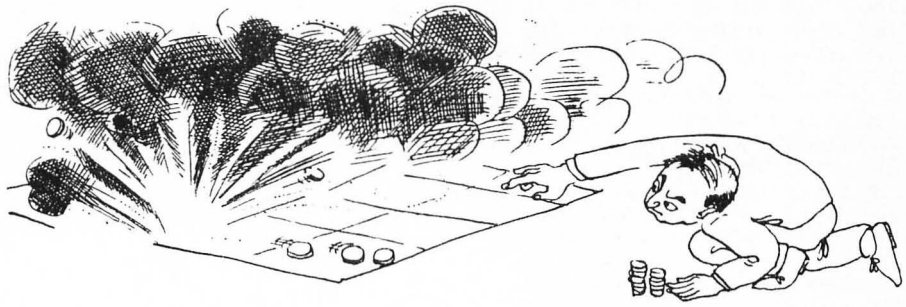


You and the computer have checkers, of opposite colors, that you place on the board in turn. You may place a checker on any empty square, or on any square which already has one or more of your own checkers on it. A square will 'explode' when the number of checkers it carries reaches its capacity. An exploding square empties, its checkers spreading out, one to each neighboring square. These checkers will 'take over' any opposing checkers on these squares.

As the game progresses the explosions get bigger and bigger. Eventually the whole board will explode in one color. If it's your color you've won!

```
10 REM ##### Explosion #####
20 DIM RB(5,5),SB(5,5),ST(5,5),AN$(1)
30 OPEN #2,4,0,"K:"
40 GOTO 1300
99 REM ### copy SB to RB ###
100 FOR X=1 TO SI:FOR Y=1 TO SI
110 SB(X,Y)=RB(X,Y)
120 NEXT Y:NEXT X:RETURN
199 REM ### copy RB to SB ###
200 FOR X=1 TO SI:FOR Y=1 TO SI
210 RB(X,Y)=SB(X,Y)
220 NEXT Y:NEXT X:RETURN
299 REM ### add to neighbours ###
300 SB(EX,EY)=PL*(ABS(SB(EX,EY))+1)
```

```
310 IF DI AND ST(EX,EY) THEN GOSUB 700
320 RETURN
399 REM ### execute explosion ###
400 SB(X,Y)=SB(X,Y)+PL
410 NE=0
420 EX=X:EY=Y:IF DI THEN GOSUB 700
430 XP=0
440 FOR X=1 TO SI:FOR Y=1 TO SI
460 IF ABS(SB(X,Y))<ST(X,Y) THEN 560
470 XP=1:NE=NE+1
490 IF DI THEN GOSUB 800
500 SB(X,Y)=SB(X,Y)-ST(X,Y)*PL
510 EX=X:EY=Y:IF DI THEN GOSUB 700
520 EX=X:EY=Y+1:GOSUB 300
530 EX=X+1:EY=Y:GOSUB 300
540 EX=X:EY=Y-1:GOSUB 300
550 EX=X-1:EY=Y:GOSUB 300
560 NEXT Y:NEXT X
570 ET=(NE>SI*SI)
580 IF XP AND  NOT ET THEN 430
590 RETURN
599 REM ### evaluate situation ###
600 EN=0
610 FOR X=1 TO SI:FOR Y=1 TO SI
620 EN=EN+SB(X,Y)
630 IF -SB(X,Y)<ST(X,Y)-1 THEN 660
640 EN=EN-2
650 EN=EN+10*((SB(X+1,Y)=ST(X+1,Y)-1)+(S
B(X,Y+1)=ST(X,Y+1)-1)+(SB(X-1,Y)=ST(X-1,
Y)-1)+(SB(X,Y-1)=ST(X,Y-1)-1))
660 NEXT Y:NEXT X
670 RETURN
699 REM ### output one value ###
700 POSITION 3+3*EX,2+2*EY
710 ? #6;CHR$(ABS(SB(EX,EY))+48+(5+2*PL)
*32*(SB(EX,EY)<>0))
720 RETURN
799 REM ### output explosion ###
800 SB=SB(X,Y):EX=X:EY=Y
810 FOR I=15 TO 1 STEP -1
820 SOUND 0,80,8,I
```



```
830 SB(X,Y)=-16:GOSUB 700
840 FOR W=1 TO 8:NEXT W
850 SB(X,Y)=SB:GOSUB 700
860 FOR W=1 TO 5:NEXT W
870 NEXT I:SOUND 0,0,0,0
880 RETURN
899 REM ### player's move ###
900 POSITION 0,11:? #6;"YOUR TURN X: ?";
910 GET #2,MX:MX=INT(MX)-48
920 IF MX<1 OR MX>SI THEN 910
930 POSITION 10,11:? #6;"Y";
940 GET #2,MY:MY=INT(MY)-48
950 IF MY<1 OR MY>SI THEN 940
960 IF RB(MX,MY)<0 THEN 900
970 POSITION 0,11:? #6;"                ";
980 GOSUB 100:REM sb -> rb              #
990 X=MX:Y=MY:DI=1
1000 GOSUB 400:REM execute explosion    #
1010 IF ET THEN 1600:REM end            #
1020 GOSUB 200:REM rb -> sb             #
1030 RETURN
1099 REM ### computer's move ###
1100 BE=1000
1110 FOR TX=0 TO SI:FOR TY=0 TO SI
1120 IF RB(TX,TY)>0 THEN 1190
1130 GOSUB 100:REM sb -> rb             #
1140 X=TX:Y=TY:DI=0
```

```
1150 GOSUB 400:REM execute explosion  #
1160 IF ET THEN MX=TX:MY=TY:GOTO 1200
1170 GOSUB 600:REM evaluate situation #
1180 IF EN<BE OR EN=BE AND RND(0)<0.4 TH
EN BE=EN:MX=TX:MY=TY
1190 NEXT TY:NEXT TX
1200 GOSUB 100:REM sb -> rb           #
1210 X=MX:Y=MY:DI=1
1220 GOSUB 400:REM execute explosion  #
1230 IF ET THEN 1600:REM end          #
1240 GOSUB 200:REM rb -> sb           #
1250 RETURN
1299 REM ### main program ###
1300 GOSUB 1400:REM initialize        #
1310 IF CM THEN 1330
1320 PL=1:GOSUB 900:REM player's move #
1330 PL=-1:GOSUB 1100:REM comp's move #
1340 NT=NT+1
1350 GOTO 1320
1399 REM ### initialize ###
1400 ? "<ESC>[<CLEAR><ESC>[=Enter size
 of board (3 or 4) ";
1410 INPUT SI:SI=INT(SI)
1420 IF SI<3 OR SI>4 THEN 1400
1430 FOR X=0 TO 5:FOR Y=0 TO 5
1440 RB(X,Y)=0:SB(X,Y)=0:ST(X,Y)=0
1450 NEXT Y:NEXT X
1460 FOR X=1 TO SI:FOR Y=1 TO SI
1470 ST(X,Y)=4-(X=1)-(X=SI)-(Y=1)-(Y=SI)
1480 NEXT Y:NEXT X
1490 ? "<ESC>[=who starts (you or me) "
;
1500 INPUT AN$:CM=(AN$="Y")
1510 GRAPHICS 18
1520 ? #6;"@[JJJ]  explosion  [JJJ]@"
1530 POSITION 5,2:? #6;"X1 X2 X3";
1540 IF SI=4 THEN ? #6;" X4";
1550 FOR Y=1 TO SI
1560 ? #6:? #6:? #6;"  Y";Y;
1570 FOR X=1 TO SI:? #6;"  ";RB(X,Y);
1580 NEXT X:NEXT Y
1590 RETURN
```

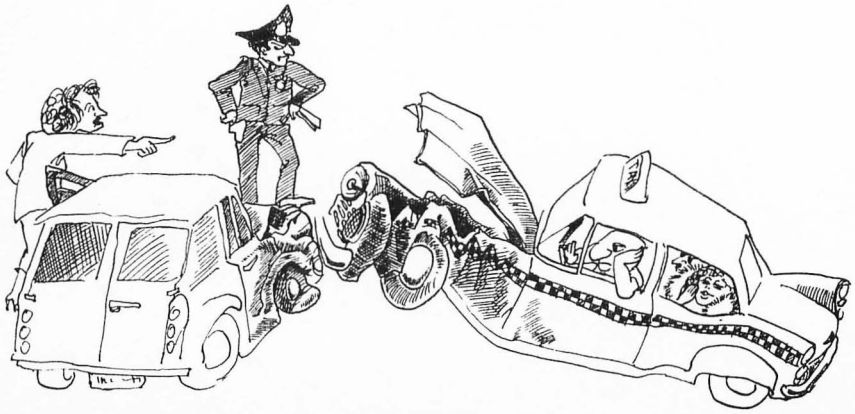
```
1599 REM ### end ###
1600 POSITION 0,1
1610 ? #6;"an eternal explosionresulted
L] so ";
1620 IF PL=1 THEN ? #6;"@you@    ";:GOTO
1640
1630 ? #6;"@!@      ";
1640 ? #6;" have won in ";NT;" turns";
1650 GOTO 1650
```

New York, New York

This original and exciting game puts you in a helicopter high over New York, looking down on the city's streets. At the moment they are deserted, but soon traffic will appear, and it's up to you to keep it moving. You do this by controlling the city's traffic lights. When a car (represented by a square) reaches a red light it will stop. Each traffic light bears a symbol — pressing the key bearing that symbol changes the light to green and the car will continue on its journey.

At the start of the game enter a level of difficulty, after which you will see the streets and cars. The game proceeds in rounds, during each of which the cars move 'one step'.

You start with 200 points. You gain 40 points for every car that gets right across the city. However, if two cars collide at a crossroads you lose 200 points,



and every time a car has to stop at a traffic light you lose 10 points. If you run out of points the game ends — if not it lasts 200 rounds. The more points you get, the better traffic cop you are!

```
10 REM ###### New York, New York ######
20 DIM CR(4),LN(4),LI(4)
30 DIM CA(8),SP(8),VC(8),S(8),LU(8)
40 DIM C$(16),K$(1)
50 DIM D$(2),S$(26),L$(35)
60 OPEN #6,0,0,"S:"
70 GOTO 1000
99 REM ### initialize ###
100 FOR C=1 TO 4:READ D:CR(C)=D:NEXT C
110 DATA 234,246,438,426
120 FOR L=0 TO 3:READ M:LI(L)=M:NEXT L
130 DATA 32,37,133,128
140 FOR C=1 TO 8:READ D:SP(C)=D:NEXT C
150 DATA 8,20,502,490,160,352,158,350
160 FOR C1=1 TO 7 STEP 2
170 READ V,LU
180 VC(C1)=V:VC(C1+1)=V
190 LC(C1)=LU:LC(C1+1)=LU
200 NEXT C1
210 DATA 32,-1,-32,2,2,33,-2,-32
220 C$="@ [Y] [Y]@[Y]@ @[Y]@ @[UUUU]@[UU
UU]@"
230 D$="@  @"
240 S$="          @    @         @    @"
250 L$="  @
   @"
260 ? "<ESC>{<CLEAR>}<ESC>{=}Level (1-3)
 ";:INPUT LE
270 LE=INT(LE)
280 IF LE<1 OR LE>3 THEN GOTO 260
290 LE=16-4*LE
300 PU=200
310 FOR G=1 TO 8:S(G)=0:NEXT G
320 FOR G=1 TO 4:LN(G)=0:NEXT G
330 RETURN
399 REM ### screen ###
```

```
400 GRAPHICS 0:POKE 752,1:POSITION 11,1
402 ? "New York, New York<ESC>[=]<ESC>[=
]<ESC>[=]"
410 FOR R=0 TO 3:? S$:NEXT R:? L$:? L$
420 FOR R=0 TO 3:? S$:NEXT R:? L$:? L$
430 FOR R=0 TO 3:? S$:NEXT R
440 POSITION 9,7:? "Q"
450 POSITION 30,7:? "W"
460 POSITION 9,18:? "A"
470 POSITION 30,18:? "S"
480 FOR C=1 TO 4:L=0:GOSUB 570
490 POSITION 1+Y,E:? "@[T]@"
500 FOR L=1 TO 3:GOSUB 570
510 POSITION 1+Y,E:? "[T]"
520 NEXT L:NEXT C
530 POSITION 5,3:? "Points:"
540 POSITION 25,3:? "Rounds:"
550 POSITION 7,22:? "LIGTHS: Red = @[T]@
, Green = [T]"
560 RETURN
570 X=CR(C)+L1(L):E=INT(X/32)
580 P=32*E:Y=X-P
590 RETURN
699 REM ### car drives ###
700 A=INT(LE*RND(0))
710 IF S(C)=0 AND A<>1 THEN RETURN
720 IF S(C)=0 AND A=1 THEN CA(C)=SP(C):R
EM new car
730 IF S(C)<>3 AND S(C)<>9 THEN GOTO 760
740 X=4+CA(C)+LO(C)-32*INT((CA(C)+LO(C))
/32):Y=5+INT((CA(C)+LO(C))/32):POSITION
X,Y:GET #6,S:POSITION X,Y:PUT #6,S
750 IF S=148 THEN PU=PU-10:SOUND 0,80,10
,15:FOR W=1 TO 20:NEXT W:SOUND 0,0,0,0:R
ETURN
760 POSITION 4+CA(C)-32*INT(CA(C)/32),5+
INT(CA(C)/32):? D$
770 CA(C)=CA(C)+V(C)
780 S(C)=S(C)+1
790 IF S(C)=15 THEN S(C)=0:PU=PU+40:FOR
W=80 TO 30 STEP -1:SOUND 0,W,10,15:NEXT
W:SOUND 0,0,0,0:RETURN
```

```
800 X=4+CA(C)-32*INT(CA(C)/32):Y=5+INT(C
A(C)/32):POSITION X,Y:GET #6,S:POSITION
X,Y:PUT #6,S:POSITION X,Y
810 IF S=160 THEN ? C$(2*C-1,2*C):RETURN

820 SOUND 0,100,0,15:FOR W=1 TO 200:NEXT
 W:SOUND 0,0,0,0
830 PU=PU-200
840 S(C)=0
850 RETURN
899 REM ### change ligths ###
900 CR=(K=47)+2*(K=46)+3*(K=62)+4*(K=63)
910 IF CR=0 THEN RETURN
920 GOSUB 970:? "[CT]"
930 LN(CR)=LN(CR)+1
940 IF LN(CR)=4 THEN LN(CR)=0
950 GOSUB 970:? "@[CT]@"
960 RETURN
970 POSITION 1+CR(CR)+LI(LN(CR))-32*INT(
(CR(CR)+LI(LN(CR)))/32),INT((CR(CR)+LI(L
N(CR)))/32)
980 RETURN
999 REM ##### main program #####
1000 GOSUB 100:REM initialize #
1010 GOSUB 400:REM screen     #
1020 FOR C=1 TO 8
1030 GOSUB 700:REM car drives #
1040 K=PEEK(764):POKE 764,255
1050 IF K<>255 THEN GOSUB 900:REM change
 ligths
1060 NEXT C
1070 NR=NR+1
1080 IF PU<0 THEN PU=0
1090 POSITION 13,3:? PU;"    "
1100 POSITION 33,3:? NR
1110 IF PU>0 AND NR<200 THEN GOTO 1020
1120 POSITION 5,22:POKE 752,0
1130 ? "<ESC><SHIFT><BACK S><ESC>[2]<ESC
><TAB><ESC><TAB>Once again ";:INPUT K$
1140 IF K$="Y" THEN RUN
1150 END
```

Key

Searching for a key you have lost can be an aggravating experience at the best of times, but when you have to find it as quickly as possible and, what's more, it's hidden inside a computer the whole thing becomes very exasperating, but also very challenging.

The computer has stored inside it a string of 20 ones and zeroes.

1 0 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 0

These figures can be shifted cyclically, that is, digits are moved from the lefthand end to the right. This is done three times and the numbers in each column added, for instance

row 1 → 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1
row 2 → 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0 1 0 1
row 3 → 0 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1
sums → 0 2 2 1 2 1 2 2 2 2 3 0 3 0 3 1 1 3 0 3

This is all kept hidden from you: it is in fact the key you must find.

What you are shown are the three rows of numbers, each shifted again. For instance if the top row is shifted two positions, the middle row four positions and the bottom row eight positions you will see

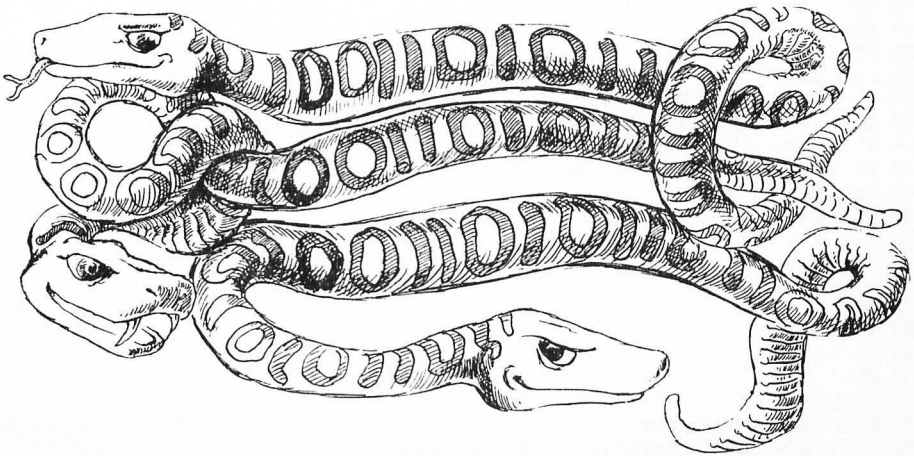
1 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0
0 1 0 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 1
1 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0 1 0 1 0
2 0 0 0 0 1 1 2 1 1 1 1 2 3 3 1 0 1 2 2

You are not shown the sum of the new columns but the difference for each column, between this sum and the first 'hidden' sum. In the first column, for instance, the hidden sum was 0 and the new sum is 2 so you see 2; in the second column the hidden sum was 2 and so is the new sum so you see 0. You must now shift the three rows until they are the same as the hidden key, when, of course, all the differences will be 0. The rows are numbered (the top row is 1, the middle 2 and the bottom 3) so that if you enter

```
ROW = 2
STEPS = 1
```

the middle row will shift one position to the left.

How few turns will it take you to find the key? It has been done in as few as 10.



```
10 REM ############## K e y ##############
20 DIM T(4,20),A$(20)
30 GRAPHICS 2
40 GOTO 600
99 REM ### initialize ###
100 ? #6;"   ***  k e y  ***"
110 ? #6;"TURN : ";NT
120 A$="10100101011101010110"
130 GOSUB 200:REM randomize rows #
140 FOR K=1 TO 20
150 T(4,K)=T(1,K)+T(2,K)+T(3,K)
160 NEXT K
170 GOSUB 200:REM randomize rows #
180 RETURN
199 REM ### randomize rows ###
200 FOR J=1 TO 3
210 R=INT(RND(0)*19+1)
220 FOR K=1 TO 20
230 DIS=K+R
240 IF DIS>20 THEN DIS=DIS-20
250 T(J,K)=VAL(A$(DIS,DIS))
260 NEXT K
270 NEXT J
280 RETURN
299 REM ### input ###
300 ? "Row :";:INPUT J
310 IF J<1 OR J>3 THEN 300
320 ? "Steps :";:INPUT S
330 IF S<1 OR S>19 THEN 320
340 RETURN
399 REM ### shift row ###
400 FOR I=1 TO S
410 H=T(J,1)
420 FOR K=1 TO 19
430 T(J,K)=T(J,K+1)
440 NEXT K
450 T(J,20)=H
460 NEXT I
470 RETURN
499 REM ### output ###
500 POSITION 7,1:? #6;NT;:? "<ESC>E<CLEA
```

```
R>]":NZ=0
510 FOR K=1 TO 20
520 POSITION K-1,3:? #6;T(1,K)
530 POSITION K-1,4:? #6;T(2,K)
540 POSITION K-1,5:? #6;T(3,K)
550 V=ABS(T(4,K)-T(3,K)-T(2,K)-T(1,K))
560 IF V=0 THEN NZ=NZ+1
570 POSITION K-1,7:? #6;V
580 NEXT K
590 RETURN
599 REM ### main program ###
600 GOSUB 100:REM initialize #
610 GOSUB 500:REM output     #
620 IF NZ=20 THEN POSITION 0,9:? #6;"goo
d in ";NT;" turns":END
630 NT=NT+1
640 GOSUB 300:REM input      #
650 GOSUB 400:REM shift row  #
660 GOTO 610
```

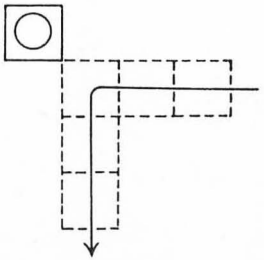
Black Box

Armed only with a laser beam you must discover the whereabouts of a handful of atoms hidden in a vast black box.

The box consists of $8 \times 8 \times 8$ cubes. Atoms can be hidden in any of the cubes, apart from those in the outer layer. However, there are never more than five atoms in the box. These atoms will reflect or divert laser beams according to the following rules.

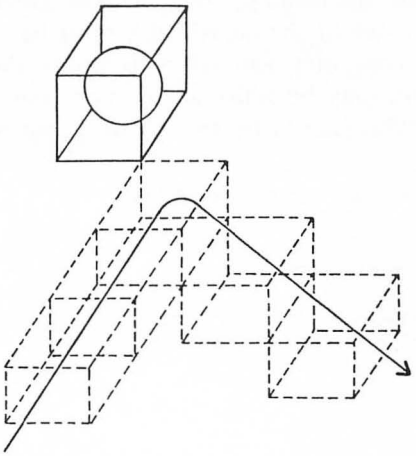
—a beam which strikes an atom is reflected straight back in the opposite direction

—a beam which is set to pass through a cube directly adjacent to an atom will be reflected at right angles

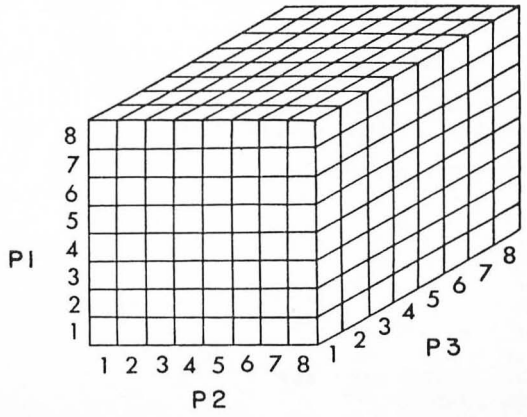


—a beam which is set to pass through a cube diagonally adjacent to an atom will be reflected in a direction which can be obtained by adding two

right-angled reflections



The box is numbered like this:



The computer will ask you to enter P1, P2, and P3, the coordinates of the position where you want the beam to enter the box. Obviously, this must be on the surface, so at least one of the coordinates must be 1 or 8. When you have entered the shot the computer will tell you where the beam has emerged. Remember that a beam may be reflected by more than one atom. After you have had a few shots you should be able to work out where the atoms are.

```
10 REM ####### B l a c k   B o x #######
20 DIM MK(5,3),HH$(1)
30 GOTO 500
99 REM ### compute result ###
100 S1=0:S2=0:S3=0
110 ON (P1=1)+(P1=8)*2+(P2=1)*3+(P2=8)*4
+(P3=1)*5+(P3=8)*6 GOSUB 240,250,260,270
,280,290
120 FOR A1=-1 TO 1
130 FOR A2=-1 TO 1
140 FOR A3=-1 TO 1
150 FOR T=1 TO 5
160 IF ((MK(T,1)=P1+A1) AND (MK(T,2)=P2+A2
) AND (MK(T,3)=P3+A3)) THEN S1=S1-A1:S2=S
2-A2:S3=S3-A3:T=5
170 NEXT T
180 NEXT A3:NEXT A2:NEXT A1
190 S1=SGN(S1):S2=SGN(S2):S3=SGN(S3)
200 P1=P1+S1:P2=P2+S2:P3=P3+S3
210 IF (P1=0)+(P1=9)+(P2=0)+(P2=9)+(P3=0
)+(P3=9)=0 THEN 120
220 ? "Result: ";P1-S1;" ";P2-S2;" ";P3-
S3
230 RETURN
240 S1=1:RETURN
250 S1=-1:RETURN
260 S2=1:RETURN
270 S2=-1:RETURN
280 S3=1:RETURN
290 S3=-1:RETURN
299 REM ### input coordinates ###
300 ? "Enter coordinates"
```

```
310 ? "P1= ";:INPUT P1
320 IF P1<1 OR P1>8 THEN 310
330 ? "P2= ";:INPUT P2
340 IF P2<1 OR P2>8 THEN 330
350 ? "P3= ";:INPUT P3
360 IF P3<1 OR P3>8 THEN 350
370 RETURN
399 REM ### initialize ###
400 FOR T=1 TO 5
410 FOR S=1 TO 3
420 MT,S)=INT(RND(0)*6+2)
430 NEXT S
440 NEXT T
450 ? " <ESC>[<CLEAR>]<ESC>[=]===  B L A
 C K  B O X  ===<ESC>[=]"
460 RETURN
499 REM ##### main program #####
500 GOSUB 400:REM initialize          #
510 ? "Please enter shot"
520 GOSUB 300:REM input coordinates #
530 IF (P1=1)+(P1=8)+(P2=1)+(P2=8)+(P3=1
)+(P3=8)<>1 THEN 520
540 GOSUB 100:REM compute result     #
550 ? :? "Shoot or guess ";:INPUT AW$
560 IF AW$="S" THEN 510
570 ? "Please enter guess"
580 GOSUB 300:REM input coordinates #
590 FOR T=1 TO 5
600 IF (MT,1)=P1) AND (MT,2)=P2) AND (
MT,3)=P3) THEN ? "Right !":GOTO 550
610 NEXT T
620 ? "Wrong !":GOTO 550
```

Treasure Hunt

Have you ever dreamed of going in search of hidden treasure? Of journeying through wild and hostile countryside, living off the land and sleeping rough, until you reach the remote and forbidding land where your glittering prize is hidden? If so, this is the game for you.

You will be taken high up into the Rocky Mountains and given a map which shows where the treasures you seek are hidden, and what they are worth. To reach them you must make your way along narrow twisting paths — one

false step means certain death — by using the I, J, K and M keys. You discover the treasures simply by treading on them.

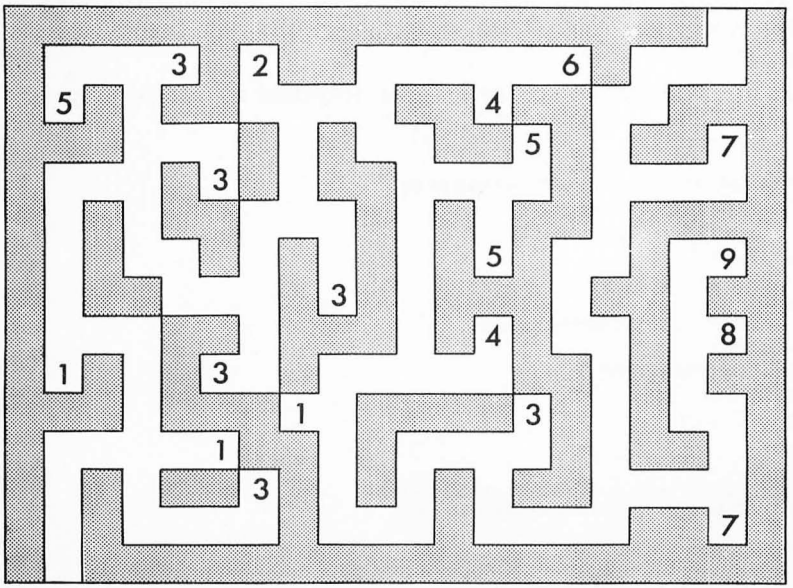
But wait . . . it's not that easy. You didn't really think it would be that simple, did you? The treasures you seek are very carefully guarded by some extremely vicious and terrifying birds who will attack you if you are not careful. If they get you in their sights you will be paralyzed with fear. The only thing you can do to try and stop them is to press the S key.

If you still feel up to the challenge start your search in the lower left hand corner of the map. Collect as much treasure as you can and take it away through the upper right hand exit. This will give you bonus points, and another chance to journey along the paths, picking up treasure. In fact you can make as many treasure-seeking trips as you can get away with. Unfortunately each time you go through you disturb more of those appalling birds, who become increasingly aggressive.

The risks are high, but so are the rewards, so gather up your courage and off you go!

```
10 REM ######### Treasure Hunt #########
20 DIM T$(15),B$(300)
30 DIM PL$(1),SP$(1),MU$(1),BI$(1)
40 OPEN #2,4,0,"K:"
50 GRAPHICS 17
60 GOTO 1000
99 REM ### change characterset ###
100 TOP=PEEK(106)-8
110 POKE 204,TOP:POKE 206,224
120 FOR U=1536 TO 1555
130 READ V:POKE U,V
140 NEXT U
150 Q=USR(1536)
160 DATA 104,162,4,160,0,177,205,145,203
,200,208,249,230,206,230,204,202,208,242
,96
170 RAMSET=TOP*256
180 FOR U=RAMSET+3*8 TO RAMSET+5*8+7
```

```
190 READ V:POKE U,V
200 NEXT U
210 POKE 756,TOP
220 RETURN
230 DATA 255,255,255,255,255,255,255,255
240 DATA 57,107,254,60,60,254,123,49
250 DATA 189,189,153,255,60,60,36,102
299 REM ### initialize & screen ###
300 MU$="@#@":BI$="[D]":PL$="@[E]@":SP$=
" "
310 ? #6;"<ESC>[<CLEAR>]";PL$;BI$;"@ tre
asure hunt @";PL$;BI$
320 ? #6;"RUNS : ";NC
330 ? #6;"BOOTY: ";P2
340 RESTORE 400
```



```
350 FOR Y=0 TO 19
360 READ T$:B$(15*Y+1)=T$:? #6;T$
370 NEXT Y
380 X=1:Y=19:POSITION X,Y+3:? #6;PL$;
390 RETURN
399 REM ### screen data ###
400 DATA @###############@ @#@
410 DATA @#@7   @#@80#@90#@ 70#@ @#@
420 DATA @##@ @#@     @#@ @##@ @#@
430 DATA @##@ @########@ @#@  @#@
440 DATA @#@       @#@      @##@
450 DATA @#@ @####@   @####@60#@
460 DATA @#@ @#@ 30######@ 50#@ @#@
470 DATA @#@   @#@ 40#@5  @#@4 @#@
480 DATA @####@ @#@ @#####@ @##@ @#@
490 DATA @#@   @#@        @#@ @#@
500 DATA @#@ @####@ @######@   @#@
510 DATA @#@      @#@4  @##@ @##@
520 DATA @#####@1@#####@     @##@
530 DATA @#@ 30##@      @##@ 20#@
540 DATA @#@ @#@1@#@30#@ @##@3 @###@
550 DATA @#@ @#@ @###@  @##@ @#@30#@
560 DATA @#@       @#@       @#@
570 DATA @###@ @##@ @###@ @##@ @#@
580 DATA @#@   @#@1      @#@5 @#@
590 DATA @#@ @###############@
599 REM ### player's move ###
600 TI=RND(0)*(400-30*NC)
610 FOR DU=1 TO TI
620 IF PEEK(764)=255 THEN 750
630 GET #2,IN
640 UX=(IN=42)-(IN=43)
650 UY=(IN=61)-(IN=45)
660 LOCATE X+UX,Y+UY+3,ST
670 IF ST=ASC(MU$) THEN 750
680 POSITION X,Y+3:? #6;SP$;
690 X=X+UX:Y=Y+UY
700 PO=ST-48
710 IF PO<0 OR PO>9 THEN 750
720 B$(Y*15+1+X,Y*15+1+X)=SP$
730 P1=P1+PO:P2=P2+PO
```

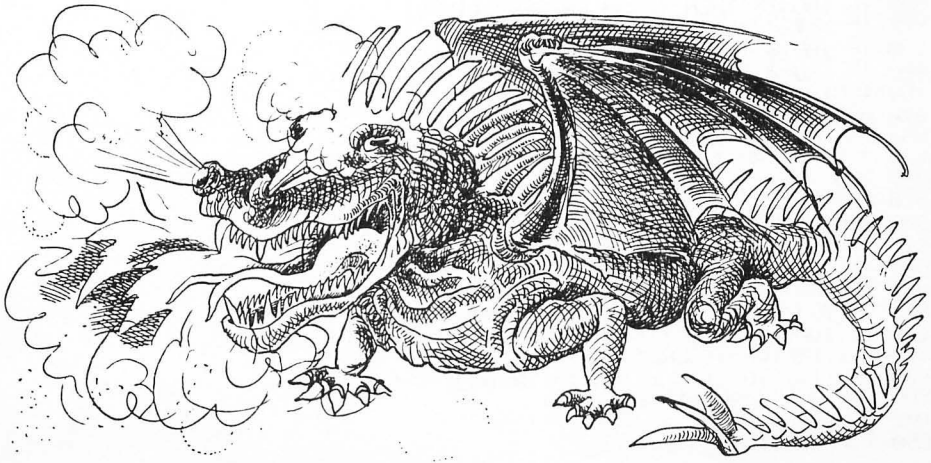
```
740 POSITION 7,2:? #6;P2;" DOLLAR";
750 POSITION X,Y+3:? #6;PL$;
760 IF Y=0 THEN DU=TI
770 NEXT DU
780 IF Y=19 THEN 600
790 RETURN
799 REM ### bird attacks ###
800 DT=25-5*NC:BX=19
810 POSITION BX,Y+3:? #6;BI$;:GOTO 840
820 BX=BX-1
830 POSITION BX,Y+3:? #6;BI$;" ";
840 FOR W=1 TO DT:NEXT W
850 IF BX>0 AND PEEK(764)<>33 THEN 820
860 IF BX<=X THEN POSITION 4,23:? #6;"en
d of game";:GOTO 860
870 POSITION 0,Y+3
880 ? #6;B$(Y*15+1,(Y+1)*15);"      ";
890 P1=P1+1
900 POSITION X,Y+3:? #6;PL$;
910 RETURN
999 REM ##### main program #####
1000 GOSUB 100:REM characterset         #
1010 GOSUB 300:REM initialize & screen #
1020 GOSUB 600:REM player's move        #
1030 IF Y=0 THEN NC=NC+1:P2=P2+NC*P1:P1=
0:GOTO 1010
1040 GOSUB 800:REM bird attacks         #
1050 GOTO 1020
```

I.T. — The Adventure of the Century

In this crazy adventure you will sink into the bowels of the earth and meet a very strange creature who lives there. His name is, of course, I.T. which stands for Intra Terrestrial.

If you have never played an adventure game before don't worry about the rules — there aren't any! You have to work everything out for yourself as you journey beneath the earth's surface, performing extraordinary tasks and facing terrifying dangers as you go. If it all gets too much for you enter *HELP* and see what the computer comes up with.

So, type in the game, take out a good insurance policy, give the *RUN* command, and off you go!



```
10 REM ############## I. T. ##############
100 GOSUB 5000
110 IHD=8:NKAM=0
120 GOTO 140
130 GOSUB 500
140 GOSUB 1000
150 GOSUB 800
160 IF CWD<6 THEN GOTO 130
170 GRAPHICS 0
180 ? "     CONGRATULATIONS!      "
190 ? "   YOU KILLED YOURSELF    "
200 STOP
500 REM ### input ###
510 NKAM=0:IHD=0
530 ? "What would you like to do"
540 INPUT YY$
545 IF LEN(YY$)=0 THEN GOTO 530
550 FOR I=0 TO 14
560 IF H$(I*11+1,I*11+LEN(YY$))=YY$ THEN
 IHD=I+1:I=15
570 NEXT I
580 IF IHD=13 THEN ? "ON OR OFF?":INPUT
YY$:IF YY$(2)="F" THEN IHD=14
590 IF IHD>6 THEN RETURN
600 IF IHD=0 THEN ? "I DON'T UNDERSTAND
YOU":GOTO 500
610 NKAM=K(IKM,IHD):IF NKAM<>0 THEN RETU
RN
620 ? "YOU CAN'T GO IN THAT DIRECTION":G
OTO 500
670 RETURN
800 REM OUTPUT
810 ? "YOU'RE IN THE ";R$(IKM*15-14,15*I
KM)
811 IF IKM>24 OR BLL=1 THEN GOTO 820
812 ? "YOU CAN'T SEE MUCH HERE":GOTO 830
820 IF IVW<>0 THEN ? "THERE IS A ";V$(12
*IVW-11,12*IVW);"HERE"
830 FOR I=1 TO 6:IF K(IKM,I)<>0 THEN ? "
YOU CAN GO ";H$(I*11-10,I*11)
840 NEXT I
850 ? :? :RETURN
```

```
1000 REM ****** OPERATIONS ********
1010 IF NKAMK>0 THEN IKM=NKAM
1020 IVW=0
1030 FOR I=1 TO 12
1040 IF V(I)=IKM THEN IVW=I:I=12
1050 NEXT I
1060 IPS=0
1070 FOR I=1 TO 6
1080 IF P(I,1)=IKM THEN IPS=I:I=6
1090 NEXT I
1100 REM **  GENERAL OPERATIONS  **
1110 IF (IKM<12) AND (CRI<>8) THEN GOSUB
 4860
1120 CLL=CLL+BLL
1200 REM ** PERSONAL OPERATIONS **
1210 IF IHD<7 THEN GOTO 1240
1220 GOSUB 1300+(IHD-7)*200
1230 REM HANDLERS
1240 IF IPS=0 THEN RETURN
1250 GOSUB 3000+(IPS-1)*200
1260 RETURN
1300 REM ****   KILL   ****
1310 IF (IPS=0) OR (IPS>4) THEN PRINT "T
IME IS THE ONLY THING YOU CAN KILL   HER
E!":RETURN
1320 LET IVWB=7
1330 GOSUB 4800
1340 IF IBZ=0 THEN ? "I WON'T LET YOU AT
TACK WITH BAREHANDS;THAT'S SUICIDE":RETU
RN
1350 PRINT "SO YOU WANT TO KILL THE";P$(
1+10*IPS-10,10*IPS)
1360 ? "YOU ATTACK IT WITH YOUR SWORD ";
1370 IF RND(0)<0.5 THEN ? "BUT YOUR VICT
IM MOVES AWAY QUICKLY":GOTO 1395
1380 ? "AND YOU GIVE YOUR VICTIM A TERRI
BLE BLOW"
1390 LET P(IPS,3)=P(IPS,3)-1
1395 IF RND(0)>0.2 THEN GOTO 1440
1400 ? "THIS IS YOUR CHANCE:YOU CAN HIT
IT    AGAIN BEFORE IT RECOVERS"
1410 ? "DO YOU WANT TO?"
1411 INPUT YY$
```

```
1420 IF YY$(1)<>"Y" THEN GOTO 1440
1430 ? "YOU RAISE YOUR SWORD AGAIN ";:GO
TO 1370
1440 IF P(IPS,3)>1 THEN RETURN
1450 IF P(IPS,3)=1 THEN ? "YOUR VICTIM I
S BADLY WOUNDED":RETURN
1460 ? "YOU KILLED YOUR VICTIM"
1470 P(IPS,1)=0:IVW=8+IPS
1480 V(IVW)=IKM:IPS=0
1490 RETURN
1500 REM ****  HELP  ****
1510 ? "POSSIBLE COMMANDS:"
1520 FOR I=1 TO 15
1530 ? H$(1+11*I-11,11*I)
1540 NEXT I
```

```
1550 ?
1560 RETURN
1700 REM ****  TAKE  ****
1710 IF IVW<>0 THEN GOTO 1740
1720 ? "THERE IS NOTHING TO TAKE HERE"
1730 RETURN
1740 ? "YOU TAKE THE ";V$(1+IVW*12-12,IV
W*12)
1751 B(IBMAX)=IVW
1760 IBMAX=IBMAX+1
1770 V(IVW)=0
1780 IVW=0
1790 RETURN
1900 REM ****  DROP  ****
1909 ? "WHAT DO YOU WANT TO DROP?"
1911 INPUT YY$
1912 L=LEN(YY$)
1920 IVWB=0
1930 FOR I=1 TO 12
1940 IF YY$=V$(1+(I-1)*12,12*(I-1)+L) TH
EN IVWB=I:I=12
1950 NEXT I
1960 IF IVWB<>0 THEN GOTO 1990
1970 ? "I DON'T UNDERSTAND YOU"
1980 RETURN
1990 GOSUB 4800
2000 IF IBZ<>0 THEN GOTO 2030
2010 ? "YOU DON'T HAVE A ";YY$
2020 RETURN
2030 ? "YOU DROPPED THE ";V$(1+12*B(IBZ)
-12,12*B(IBZ))
2040 IVW=B(IBZ)
2050 V(B(IBZ))=IKM
2060 B(IBZ)=B(IBMAX-1)
2070 IBMAX=IBMAX-1
2080 ? "DO YOU WANT TO DROP MORE?"
2090 INPUT YY$
2091 IF YY$(1,1)="Y" THEN GOTO 1900
2095 RETURN
2100 REM *** INVENT ***
2110 IF IBMAX<>1 THEN GOTO 2140
2120 ? "YOU OWN ONLY YOUR CLOTHES"
```

```
2130 GOTO 2180
2140 ? "YOU HAVE THE FOLLOWING THINGS:"
2150 FOR I=1 TO IBMAX-1
2160 ? V$(1+12*B(I)-12,12*B(I))
2170 NEXT I
2180 ? "YOU CAN SURVIVE ";5-CWD;" MORE W
OUNDS"
2190 RETURN
2300 REM ***  BANDAGE  ***
2310 IVWB=6
2320 GOSUB 4800
2330 IF IBZ<>0 THEN GOTO 2360
2340 ? "YOU WON'T MANAGE THAT WITHOUT BA
NDAGE"
2350 RETURN
2360 ? "IT DOESN'T LOOK VERY HOPEFUL"
2370 ? "YOU'LL  NEED ALL YOU HAVE"
2380 SOUND 0,244,10,10
2390 ? "SO THAT WILL HOLD FOR A WHILE"
2400 CWD=0
2410 B(IBZ)=B(IBMAX-1)
2420 IBMAX=IBMAX-1
2430 RETURN
2500 REM ***  LANTERN ON ***
2510 IVWB=1
2520 GOSUB 4800
2530 IF IBZ<>0 THEN GOTO 2560
2540 ? "GET A LANTERN FIRST"
2550 RETURN
2560 ? "YOU TURNED THE LANTERN ON"
2570 BLL=1
2580 RETURN
2700 REM **** LANTERN OFF ****
2710 IVWB=1
2720 GOSUB 4800
2730 IF IBZ<>0 THEN GOTO 2760
2740 ? "YOU DO NOT EVEN HAVE A LANTERN"
2750 RETURN
2760 ? "THE LANTERN IS OFF"
2770 BLL=0
2780 RETURN
2900 REM *****  READ  ******
```

```
2905 IVWB=4
2910 GOSUB 4800
2915 IF IBZ<>0 THEN GOTO 2930
2920 ? "YOU CAN'T READ A BOOK IF YOU DON
'T HAVE ONE"
2925 RETURN
2930 IF BLL=1 OR IKM>24 THEN GOTO 2955
2940 ? "IT IS TOO DARK TO READ HERE"
2950 RETURN
2955 ? "THERE IS A RECIPE FOR COOKIES IN
 THIS BOOK"
2960 ? "IT SAYS:TAKE ONE DEAD HELLHOUND
AND   SOME WHEAT;DROP THIS ON A BARBECUE
    AND WAIT FOR A FEW MINUTES"
2980 ? "THAT'S ALL"
```

```
2990 RETURN
3000 REM ******** I.T. ********
3010 IF IVW=3 THEN P(1,2)=3:V(IVW)=0:IVW
=0
3020 GOTO 3020+P(1,2)*10
3030 ? "I.T.(THE INTRATERRESTIAL) IS HER
E"
3031 ? "HE'S TALKING URGENTLY,"
3032 ? "BUT YOU DON'T UNDERSTAND HIM"
3033 P(1,2)=2
3034 RETURN
3040 ? "I.T. GESTURES THAT HE IS VERY"
3042 ? "THIRSTY. HE LOOKS RATHER DESPERA
TE"
3043 RETURN
3050 ? "I.T. DRINKS THE WATER AS HE NEED
S"
3051 ? "IT VERY MUCH.AFTER A WHILE HE ST
ARTS"
3052 ? "TALKING AGAIN.NOW YOU CAN"
3053 ? "UNDERSTAND HIM"
3054 ? "he says :PLEASE GO DOWN INTO THE
  CAVES AND FIND THE NULLITY BOMB"
3055 ? "SOME CRAZY PROFESSOR WANTS TO BL
OW UP THE EARTH WITH IT"
3056 ? "I HAD A FIGHT WITH THE MONSTER T
HAT  PROTECTS IT"
3057 ? "NOW I'M TOO WEAK;SO ONLY YOU C
AN  SAVE THE EARTH FROM DESTRUCTION"
3058 ? "SUDDENLY I.T. COLLAPSES"
3059 P(1,2)=4:RETURN
3060 ERROR-  ? "IT LOOKS LIKE I.T. IS I
N COMA"GOTO3050
3061 ? "NOW YOU HAVE TO DO IT ALL ALONE"
3062 ? "GOOD LUCK"
3063 P(1,2)=5
3064 RETURN
3070 ? "I.T. IS HERE "
3071 ? "HE IS IN COMA"
3072 RETURN
3083 RETURN
3200 REM **  DRAGON  **
3210 IF IVW=8 THEN P(2,2)=5:V(IVW)=0:IVW
```

```
=0
3220 GOTO 3220+(10*P(2,2))
3230 ? "THERE IS AN ENORMOUS MONSTER HER
E"
3231 ? "ITS EYES ARE ROLLING"
3232 ? "IT YELLS:ARE YOU COOKIE?"
3233 P(2,2)=2
3234 RETURN
3240 ? "THE MONSTER YELLS LOUDER AND LOU
DER:"
3241 ? " ARE  YOU  COOKIE?"
3242 P(2,2)=3
3243 RETURN
3250 ? "IT KEEPS YELLING AND BECOMES RAT
HER  AGRESSIVE"
3252 P(2,2)=4
3253 RETURN
3260 ? "THE MONSTER GIVES YOU A TERRIBLE
 BLOW"
3261 ? "YOUR HEAD IS SPINNING"
3262 CWD=CWD+1
3263 P(2,2)=INT(3*(RND(0))+2)
3264 RETURN
3270 ? "THE MONSTER SAYS SURPRISED:COOKI
ES?"
3271 ? "IT STARTS EATING AT ONCE"
3272 ? "AN ENORMOUS  BONG  AND IT FALLS
ASLEEP"
3273 K(16,6)=1
3274 P(2,2)=6
3275 RETURN
3280 ? "THE MONSTER IS ASLEEP"
3281 RETURN
3400 REM ****  SNAKE  ****
3410 ? "THERE IS A SNEAKY SNAKE HERE"
3420 IF RND(0)<0.4 THEN RETURN
3430 IF (IBMAX=1) OR (RND(0)<0.5) THEN G
OTO 3470
3440 IBMAX=IBMAX-1
3450 V(B(IBMAX))=13+INT(12*RND(0))
3460 ? "WITH A QUICK MOVE IT SNATCHES
     SOMETHING"
```

```
3470 ? "AND IT SNEEKS AWAY"
3480 P(3,1)=P(3,1)+3
3490 IF P(3,1)>24 THEN P(3,1)=P(3,1)-6
3500 RETURN
3600 REM ***  HELLHOUND  ***
3620 GOTO 3620+P(4,2)%10
3630 ? "THERE IS A GIANT HELLHOUND HERE"
3631 ? "IT LOOKS LIKE HE WANTS YOU FOR D
INNER"
3632 P(4,2)=2
3633 RETURN
3640 ? "THE HELLHOUND ATTACKS YOU AND BI
TES YOU VIOLENTLY"
3641 CWD=CWD+1
3642 P(4,2)=3
3643 RETURN
3650 ? "THE HELLHOUND GROWLS AND SEEMS T
O     PREPARE FOR ANOTHER ATTACK"
3651 P(4,2)=2+INT(RND(0)%2)
3652 RETURN
3800 REM ****   BOMB     ****
3810 ? "THE NULLITY BOMB IS HERE."
3820 ? "THERE ARE THREE WIRES BETWEEN TH
E"
3830 ? "BOMB AND THE TIME MECHANISM:A GR
EEN"
3831 ? "ONE (G),A YELLOW ONE (Y) AND A R
ED ONE (R)"
3832 ? :? "YOU MUST DISCONNECT TWO OF TH
EM TO    STOP IT"
3840 ? "WHICH WILL BE THE FIRST ONE?"
3845 INPUT XX$
3850 ? "AND THE SECOND ONE?"
3855 INPUT YY$
3857 CB=0
3860 Q$="YRG"
3861 FOR T=1 TO 3
3865 IF XX$(1,1)=Q$(T,T) OR YY$(1,1)=Q$(
T,T) THEN CB=CB+1
3870 NEXT T
3880 IF CB<2 THEN ? "WATCH OUT! WRONG IN
PUT!":GOTO 3840
3900 IF ABS(ASC(XX$)-ASC(YY$))=11 THEN G
```

```
OTO 3930
3905 GRAPHICS 0:? "ENORMOUS EXPLOSION
MUSHROOM CLOUD!!"
3910 END
3930 ? "CONGRATULATIONS!!!  YOU SUCCEEDE
D     WHERE EVERYONE ELSE FAILED"
3950 END
4000 REM ** BARBECUE **
4010 IF IVW=0 THEN GOTO 4060
4030 IF (V(2)=IKM) AND (V(12)=IKM) THEN
IVW=8:V(8)=IKM:V(2)=0:V(12)=0:P(6,2)=2
4060 GOTO 4060+10*P(6,2)
4070 ? "THERE IS A GIANT BARBECUE HERE"
4071 ? "WITH A LARGE FIRE UNDER IT"
4072 RETURN
4080 ? "AN ENORMOUS FLASH LIGHTS THE PLA
CE"
4081 ? "AND A PENETRATING SMELL FILLS YO
UR NOSE"
4082 P(6,2)=3
4083 RETURN
4090 ? "EVERYTHING IS QUIET NOW"
4091 ? "EVEN THE TERRIBLE SMELL FADES"
4092 P(6,2)=1
4093 RETURN
4800 REM ** LOOK IN OWNINGS **
4810 IBZ=0
4820 FOR I=1 TO IBMAX-1
4830 IF B(I)=IVWB THEN LET IBZ=I:I=IBMAX
-1
4840 NEXT I
4850 RETURN
4860 REM ****  PUZZLE  **
4865 CRO=CRO+1
4870 IF IKM=D(CRO) THEN CRI=CRI+1
4875 IF CRO<8 THEN RETURN
4880 IF CRI=8 THEN GOTO 4915
4885 PRINT "STRANGE THINGS ARE HAPPENING
     "
4890 PRINT "AN ABSOLUTE DARKNESS COVERS
YOU AND   IT FEELS AS IF AN INVISIBLE FO
RCE IS  LIFTING YOU"
```

```
4895 ? "FOR A MOMENT YOU ARE UNCONSCIOUS
"
4900 CRO=1:CRI=1
4905 IKM=1
4910 RETURN
4915 ? "YOU HEAR A STRANGE SOUND LIKE SO
METHING IS BEING PUSHED AWAY"
4920 ? "NOW IT HAS STOPPED"
4925 K(2,1)=3
4930 RETURN
5000 REM ### initialization ###
5010 ? "<ESC><SHIFT><CLEAR><ESC>[=]<ESC>
<TAB><ESC><TAB>@          @"
5020 ? "<ESC><TAB><ESC><TAB>@   I. T.
@"
5030 ? "<ESC><TAB><ESC><TAB>@          @
 "
5040 ? "<ESC><TAB><ESC><TAB>@Please wait
@"
5050 DIM K(36,6),R$(540),H$(165)
5060 DIM P(6,3),P$(60),V(12),V$(144)
5070 DIM B(12),D(8)
5100 REM ### connections ###
5110 FOR I=1 TO 36
5120 K(I,1)=I+1:K(I,2)=I-1:K(I,5)=0
5130 K(I,3)=I+4:K(I,4)=I-4:K(I,6)=0
5140 NEXT I
5150 FOR I=0 TO 24 STEP 12
5160 FOR J=1 TO 9 STEP 4
5170 K(I+J+3,1)=0:K(I+J,2)=0
5180 NEXT J
5190 FOR J=1 TO 4
5200 K(I+J+8,3)=0:K(I+J,4)=0
5210 NEXT J:NEXT I
5230 K(1,5)=16:K(7,5)=15
5240 K(32,6)=13:K(13,5)=32
5250 K(35,6)=18:K(18,5)=35
5260 FOR I=1 TO 15
5270 READ V:READ W:LET K(V,W)=0
5290 NEXT I
5300 REM ## initialize parameters ###
5310 IKM=36:IBZ=1:IBMAX=1:CKM=0
```

```
5340 CLL=0:BLL=0:CWD=0:CRO=0:CRI=0
5400 REM ### fill arrays ###
5405 DIM YY$(15):DIM XX$(15)
5409 H$=" ":H$(165)=" ":H$(2)=H$
5410 FOR I=1 TO 15
5420 READ YY$:H$((I-1)*11+1,I*11)=YY$
5430 NEXT I
5435 V$=" ":V$(144)=" ":V$(2)=V$
5440 FOR I=1 TO 12
5450 READ YY$:V$((I-1)*12+1,I*12)=YY$
5460 NEXT I
5465 P$=" ":P$(60)=" ":P$(2)=P$
5470 FOR I=1 TO 6
5480 READ YY$:P$((I-1)*10+1,I*10)=YY$
5490 NEXT I
5495 R$=" ":R$(540)=" ":R$(2)=R$
5500 FOR I=1 TO 36
5510 READ YY$:R$((I-1)*15+1,I*15)=YY$
5520 NEXT I
5530 FOR I=1 TO 12:READ D:V(I)=D:NEXT I
5560 FOR I=1 TO 6:READ D,E
5570 P(I,1)=D:P(I,2)=1:P(I,3)=E
5590 NEXT I
5600 FOR I=1 TO 8:READ E:D(I)=E:NEXT I
5700 RETURN
6000 DATA EAST,WEST,NORTH,SOUTH,UP,DOWN,
KILL,HELP,TAKE,DROP,INVENTORY,BANDAGE,LA
NTERN ON,LANTERN OFF,READ
6010 DATA LANTERN,WHEATPILE,WATERSACK,CO
OKBOOK,LEAFLET,BANDAGES,SWORD
6011 DATA COOKIE,BODY OF I.T.,GIANT CORP
SE,SLICED SNAKE,DEAD HOUND
6020 DATA I.T.,MONSTER,SNAKE,HELLHOUND,B
OMB,BARBECUE
6030 DATA RESET CAVE,T-CAVE,SECRET CORRI
DOR,CONTROL ROOM,O-CAVE,I-CAVE,SPACE CAV
E,BLACK ROOM,P-CAVE,E-CAVE,N-CAVE
6032 DATA EMPTYNESS,SMALL CAVE,ROCKY CAV
E,SMELLY CAVE,DRAGON CAVE,SNAKE CAVE,YEL
LOW CAVE,STREAM BANK,STINKY PLACE
6033 DATA FOOD CAVE,FINAL CAVE,COLORED C
AVE,ICE CAVE,OPEN PLACE,WOODS,WOODS,WOOD
S,WOODS,WOODS,WOODS,WOODS,WOODS
```

```
6035 DATA WOODS,WOODS,WOODS
6040 DATA 33,30,28,21,14,15,13,0,0,0,0,0
6050 DATA 34,2,16,15,17,4,29,2,8,1,25,1
6060 DATA 1,5,9,10,11,7,6,2
```

The Wolf and the Five Little Goats

A Grimm's fairy tale? No, an intriguing board game! This game is played on a checkerboard displayed on the screen. As the game begins, the goats are scattered throughout the lower half of the board and are represented by little squares containing a number. The wolf stands at the upper left-hand corner of the board.

Off we go — you are the wolf, and the computer controls the goats. You win if you eat three goats, and the computer wins if one of the goats eats you. (These wolf-eating goats are amazing creatures!)

You may start. On the lower part of the screen you see:

```
STILL . . . MOVES
```

for example

```
STILL 2 MOVES
```

The number of moves always lies between 1 and 3, and tells you how many steps you may take during your turn. The wolf — in other words, you — may move horizontally or vertically, but never diagonally. Each move must be entered by using the I, J, K and M keys as cursor keys. If the *last* move of a series of moves brings you on a square with a goat, the goat is yours. You may never cross a square that has a goat on it.

There are 5 goats in all, and they may jump over each other. Goats can move only in one direction. The number of steps they may move is shown on the goats themselves, and is always between 1 and 5. For instance, if a goat bears the number 3 he can move

3 steps to the left, or
3 steps to the right, or
3 steps forward, or
3 steps backward

He cannot move 1 step forward and 2 steps to the left. If you (the wolf) have moved to a new position the computer shows

NOW IT'S MY TURN

When it is your turn, the computer shows

STILL . . . MOVES

At the end of the game the computer either tells you

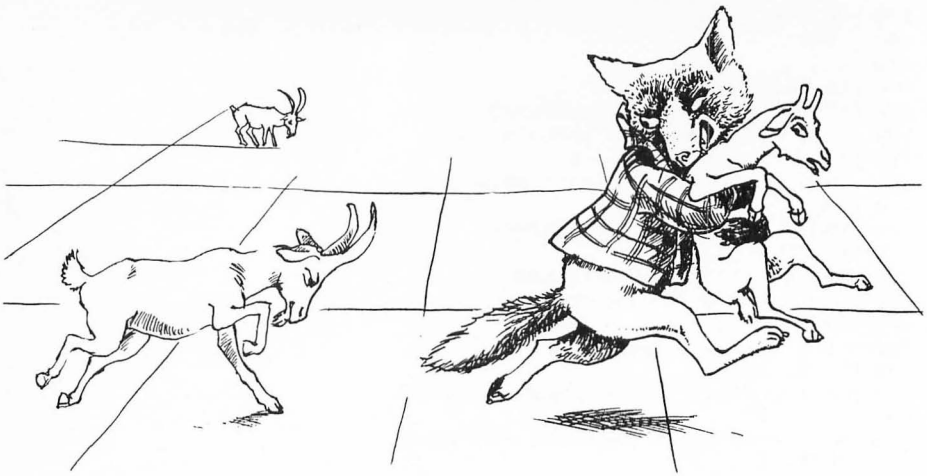
YOU WIN!

or worse

YOU LOSE

```
10 REM ##### The Wolf And       #####
       ##### The 5 Little Goats #####
20 DIM P(4,2),T$(1)
30 OPEN #2,4,0,"K:":OPEN #3,4,0,"S:"
40 GRAPHICS 18
50 GOTO 1000
50 GOTO 1000
99 REM ### initialize ###
100 RBMIN=1:RBMAX=10:KBMIN=5:KBMAX=14
110 RW=RBMIN:KW=KBMIN:MAX=-400:BEAT=0
120 ? #6;"@   wolf[CF][LU] goats@"
```

```
130 FOR I=0 TO 9
140 ? #6;"     ..........."
150 NEXT I
160 FOR I=0 TO 4
170 P(I,0)=INT(5*RND(0))+RBMIN+5
180 P(I,1)=INT(10*RND(0))+KBMIN
190 P(I,2)=INT(5*RND(0))+1
200 POSITION P(I,1),P(I,0):? #6;P(I,2)
210 NEXT I
220 POSITION KW,RW:? #6;"W":BW=2
230 RETURN
299 REM ### input and move ###
300 FOR BW=BW TO 1 STEP -1
310 POSITION 2,11
320 ? #6;" still ";BW;" moves  "
330 GET #2,MOVE
340 DRW=(MOVE=61 AND RW<RBMAX)-(MOVE=45
AND RW>RBMIN)
350 DKW=(MOVE=42 AND KW<KBMAX)-(MOVE=43
AND KW>KBMIN)
360 IF DRW+DKW=0 THEN 330
370 POSITION KW+DKW,RW+DRW:GET #3,X
380 IF CHR$(X)<>"." AND BW<>1 THEN 330
390 POSITION KW,RW:? #6;"."
400 RW=RW+DRW:KW=KW+DKW
410 POSITION KW,RW:? #6;"W"
420 NEXT BW
430 FOR I=0 TO 4
440 IF RW=P(I,0) AND KW=P(I,1) AND P(I,2
)<>0 THEN P(I,2)=0:BEAT=BEAT+1
450 NEXT I
460 BW=INT(3*RND(0)+1):POSITION 2,11
470 IF BEAT=3 THEN ? #6;"@   you win [A]
  @":GET #2,X:END
480 ? #6;"@now it[G]s my turn@"
490 RETURN
499 REM ### status ###
500 STAT=0
510 IF RS<RBMIN OR RS>RBMAX OR KS<KBMIN
OR KS>KBMAX THEN RETURN
520 D=ABS(RW-RS)+ABS(KW-KS)
530 IF D=BW THEN RETURN
```



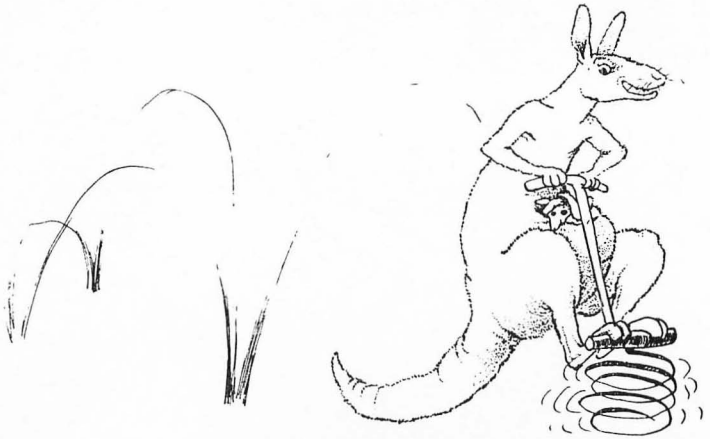
```
540 IF D=BW-2 THEN STAT=-200
550 IF D=0 THEN STAT=500:GOTO 640
560 IF D<=5 THEN STAT=STAT+35
570 IF RS=RW THEN STAT=STAT+40
580 IF KS=KW THEN STAT=STAT+40
590 IF ABS(P(I,0)-RW)+ABS(P(I,1)-KW)=BW
THEN STAT=STAT+80
600 FOR T=0 TO 4
610 IF RS<>P(T,0) THEN STAT=STAT+10
620 IF KS<>P(T,1) THEN STAT=STAT+10
630 NEXT T
640 IF STAT+RND(0)<MAX THEN RETURN
650 MAX=STAT:IMAX=I:RMAX=RS:KMAX=KS
660 RETURN
699 REM ### evaluate goats ###
700 MAX=-400
710 FOR I=0 TO 4
720 IF P(I,2)=0 THEN 810
730 KS=P(I,1):RS=P(I,0)+P(I,2)
```

```
740 GOSUB 500
750 RS=P(I,0)-P(I,2)
760 GOSUB 500
770 RS=P(I,0):KS=P(I,1)+P(I,2)
780 GOSUB 500
790 KS=P(I,1)-P(I,2)
800 GOSUB 500
810 NEXT I
820 RETURN
899 REM ### move goat ###
900 POSITION P(IMAX,1),P(IMAX,0)
910 ? #6;"."
920 P(IMAX,0)=RMAX:P(IMAX,1)=KMAX
930 P(IMAX,2)=INT(5*RND(0)+1)
940 POSITION KMAX,RMAX:? #6;P(IMAX,2)
950 IF MAX<400 THEN RETURN
960 POSITION 2,11
970 ? #6;"   @you lose [HJK  "
980 GET #2,X:END
999 REM ##### main program ###
1000 GOSUB 100:REM initialize      #
1010 GOSUB 300:REM input & move    #
1020 GOSUB 700:REM evaluate goats #
1030 GOSUB 900:REM move goat       #
1040 GOTO 1010
```

Spring

A number of well-known computer games are based on fairly straightforward physical equations. For instance all those 'lunar landing' games use an equation which gives a position as a function of time when the initial position and velocity and the acceleration due to gravity are known.

SPRING is also based on a physical idea, or to be more precise, a physical fantasy. Imagine a spring with one end fixed to the centre of your computer screen and a ball attached to the other. If this were a normal spring, when you stretched it and let it go the ball would simply bounce back and forth. This however is an unusual spring, subject to some very mysterious forces which will cause the ball to move in a most peculiar manner — as you will soon see!



```
10 REM ########## S p r i n g ##########
20 GRAPHICS 24
30 SETCOLOR 1,0,15:COLOR 1
40 SETCOLOR 2,RND(0)*16,6
50 SETCOLOR 4,RND(0)*16,6
60 REM starting pos. & spring constants
70 X=RND(0)*309.5+4.5
80 Y=RND(0)*185.5+2.5
90 AX=RND(0)*50+5:AY=RND(0)*50+5
100 PLOT X,Y
110 REM compute velocity & displacement
120 VX=VX+(159.5-X)/AX:X=X+VX
130 VY=VY+(95.5-Y)/AY:Y=Y+VY
140 DRAWTO X,Y
150 GOTO 120
```

Apollo 2000

Remember July 1969 . . . . that 'giant leap for mankind' when Neil Armstrong stepped onto the moon. When we recall the excitement felt throughout the world on that day it's hardly surprising that so many movies and computer games take space travel as their theme.

In Apollo 2000 you control a space module on its journey to a planet. After the RUN command you can choose your destination — any of the nine planets in our solar system or the mysterious planet named RANDOM. Having made your choice the planet's landscape will appear on the screen, revealing the plateau on which you are to land.

There is a choice of 2 levels of play. Level 1 is relatively simple. Level 2 is a bit more difficult because now your module is in orbit around the planet.

Vertical movement is controlled by the keys 0-9, where 0 gives no boost from your landing rockets, while horizontal movement is achieved by using the keys and . Once a key is pressed, your module will maintain its direction and velocity until you change their values. All relevant data are displayed on the screen.

```
10 REM ######### Appollo 2000 #########
20 DIM C$(20),S$(10),PL$(20)
30 OPEN #2,4,0,"K:"
40 GOTO 3200
99 REM ### input ###
100 IF FUEL<=0 THEN POKE 656,3:POKE 657,
21:? "TANK EMPTY !! <ESC>[<2>]";:GAS=0:H
G=0:GG=0:RETURN
110 IF PEEK(764)=255 THEN 160
120 GET #2,X:X=X-48
130 IF X>=0 AND X<=9 THEN HG=0:GAS=X:GOT
O 150
140 HG=10*((X=14)-(X=12))
150 GG=GAS+ABS(HG)
```

```
160 IF GG*INT<=FUEL THEN RETURN
170 TEK=FUEL-GG*INT:TEK=1+TEK/GG/INT
180 GAS=TEK*GAS:HG=TEK*HG:GG=GAS+ABS(HG)
190 RETURN
199 REM ### computation ###
200 GW=GG*SG:R=R0+HGT:PAR=INT
210 OHV=HV:GOSUB 600
220 ODIS=DIS:GOSUB 500
230 OVEL=VEL:GOSUB 400:VEL=FUN
240 OHGT=HGT:GOSUB 300:HGT=FUN
250 TIME=TIME+INT:FUEL=FUEL-GG*INT
260 M=WGT+FUEL*SG
270 RETURN
299 REM ### compute new heigth ###
300 FUN=OHGT+PAR*(OVEL+PAR*(((HV+OHV)^2/
4-JM/R)/2/R))
310 IF GAS=0 THEN RETURN
320 FUN=FUN+EFF*GAS/GW*((PAR-M/GW)*LOG(M
/(M-GW*PAR))+PAR)
330 RETURN
399 REM ### compute new velocity ###
400 FUN=OVEL+PAR/R*((HV+OHV)^2/4-JM/R)
410 IF GAS=0 THEN RETURN
420 FUN=FUN+EFF*GAS/GW*LOG(M/(M-GW*PAR))
430 RETURN
499 REM ### compute new distance ###
500 DIS=ODIS+(OHV+HV)/2*PAR:RETURN
599 REM ### compute new h. velocity ###
600 IF HG=0 THEN RETURN
610 HV=OHV+HEFF*HG*PAR/M:RETURN
699 REM ### move ship ###
700 X=INT(BASE+10+ODIS/HSC+0.5)
710 Y=INT(HB-5-OHGT/SC+0.5)
720 COLOR 0:GOSUB 800
730 X=INT(BASE+10+DIS/HSC+0.5)
740 IF X<5 OR X>154 THEN RETURN
750 Y=INT(HB-5-HGT/SC+0.5)
760 IF Y<5 OR Y>91 THEN RETURN
770 COLOR 2:GOSUB 800
780 RETURN
799 REM ### plot ship ###
```

```
800 PLOT X-1,Y-5:DRAWTO X+2,Y-5
810 PLOT X-2,Y-4:DRAWTO X+2,Y-4
820 PLOT X-3,Y-3:DRAWTO X+3,Y-3
830 PLOT X,Y-2:DRAWTO X+3,Y-2
840 PLOT X,Y-1:DRAWTO X+3,Y-1
850 PLOT X-3,Y-2:DRAWTO X-3,Y-1
860 PLOT X-3,Y:DRAWTO X+3,Y
870 PLOT X-3,Y+1:DRAWTO X+3,Y+1
880 PLOT X-4,Y+5:PLOT X-5,Y+5
890 DRAWTO X-2,Y+2:PLOT X+4,Y+5
900 PLOT X+5,Y+5:DRAWTO X+2,Y+2
910 RETURN
999 REM ### output values ###
1000 POKE 656,0:POKE 657,11
1010 ? C$;INT(VEL*100)/100;
1020 POKE 657,29:? C$;INT(FUEL*100)/100
1030 POKE 657,11:? C$;INT(HGT*100)/100;
1040 POKE 657,29:? C$;INT(TIME*100)/100
1050 POKE 657,11:? C$;INT(HV*100)/100;
1060 POKE 657,29:? C$;INT(M*100)/100
1070 POKE 657,11:? C$;INT(DIS*100)/100;
1080 POKE 657,29:? C$;GAS;
1090 IF HG<0 THEN ? "  <";
1100 IF HG>0 THEN ? "  >";
1110 RETURN
1199 REM ### check position of ship ###
1200 X=INT(BASE+10+DIS/HSC+0.5)
1210 Y=INT(HB-5-HGT/SC+0.5)
1220 REM # out of range #
1230 IF X<5 OR X>154 OR Y<5 THEN 1400
1240 REM # beneath platform #
1250 IF X<BASE OR X>BASE+20 THEN 1330
1260 IF HGT<=0 THEN S$="platform":GOTO 1
600
1270 IF VEL<0 OR OVEL>0 THEN 1330
1280 INT1=0:INT2=INT:EPS=1.0E-03
1290 IF HGT<=0 THEN INT2=NUL:GOTO 1610
1300 ADR=400:GOSUB 3100
1310 PAR=NUL:GOSUB 300:HGT=FUN
1320 IF HGT<=0 THEN INT2=NUL:GOTO 1610
1330 REM # too low
```

```
1340 IF Y>74 THEN S$="surface":GOTO 1800
1350 REM # touching planet
1360 LOCATE X-5,Y+5,C1:LOCATE X+5,Y+5,C2
1370 IF C1=1 OR C2=1 THEN S$="surface":G
OTO 1800
1380 RETURN
1399 REM ### out of range ###
1400 ? "<ESC>[<2>J<ESC>[<CLEAR>]
FLYING OUT OF RANGE"
1410 ? "       COMUNICATIONS LOST !"
1420 GOSUB 700
1430 SOUND 0,60,10,15:GOSUB 1500
1440 SOUND 0,120,10,15:GOSUB 1500
1450 SOUND 0,0,0,0
1460 ? "<ESC>[=]       Try again ?";
1470 POKE 764,255:GET #2,X
1480 IF X<>89 THEN 2230
1490 POP :POP :GOTO 3230
1499 REM ### wait ###
1500 FOR W=1 TO 70:NEXT W:RETURN
1599 REM ### crash or land ? ###
1600 INT1=0:INT2=INT
1610 EPS=1.0E-03:ADR=300:GOSUB 3100
1620 PAR=NUL:GOSUB 400:VEL=FUN
1630 TIME=TIME+NUL
1640 IF VEL<LANVEL THEN 1800
1699 REM ### land !! ###
1700 GRAPHICS 0:SETCOLOR 1,0,0:SETCOLOR
2,11,7
1710 ? "<ESC>[<CLEAR>]<ESC>[=]<ESC>[=]
You are landed savely on ";PL$
1720 SUR=100:GOSUB 1760
1730 ? "<ESC>[=]  No damage."
1740 ? "<ESC>[=]  Your mark for this lan
ding:";INT(5+VEL/2)+5
1750 GOSUB 2200:RETURN
1760 ? "<ESC>[=]  Speed: ";INT(VEL*100)/
100;:? " m/s."
1770 ? "<ESC>[=]  Time: ";INT(TIME*100)/
100;:?
1780 ? "<ESC>[=]  Survived crew: ";SUR;"
 %"
```

```
1790 RETURN
1799 REM ### crash !! ###
1800 GOSUB 700
1810 IF X<5 THEN X=5
1820 IF X>154 THEN X=154
1830 IF Y<5 THEN Y=5
1840 IF Y>74 THEN Y=74
1850 FOR L=15 TO 0 STEP -1
1860 SOUND 0,30,9,L:SOUND 2,120,0,L
1870 SOUND 2,120,0,L:SOUND 3,140,0,L
1880 COLOR 0:GOSUB 2100
1890 COLOR 2:GOSUB 2100
1900 NEXT L
1910 GRAPHICS 0
1920 SETCOLOR 1,0,0:SETCOLOR 2,3,7
1930 ? "<ESC>[<CLEAR><ESC>[=]<ESC>[=]
You are crashed on the ";S$
1940 ? "<ESC>[=]  on planet ";PL$
1950 SUR=0:IF ABS(VEL)<30 THEN SUR=100-I
NT(ABS(VEL)/0.3)
1960 GOSUB 1760
1970 ? "<ESC>[=]  Damage: ";
1980 IF VEL<-15 THEN ? "un";
1990 ? "reparable."
2000 ? "<ESC>[=]  You created a crater,
depth: ";INT(1.56*ABS(VEL))/10;" m."
2010 GOSUB 2200:RETURN
2100 FOR I=0 TO 5
2110 PLOT X-I,Y-I:PLOT X,Y-I
2120 PLOT X+I,Y-I:PLOT X+I,Y
2130 PLOT X+I,Y+I:PLOT X,Y+I
2140 PLOT X-I,Y+I:PLOT X-I,Y
2150 NEXT I:RETURN
2199 REM ### again ? ###
2200 ? "<ESC>[=]<ESC>[=]<ESC>[=]<ESC>[=]
 Do you want another time ?";
2210 POKE 764,255:GET #2,X
2220 IF X=89 THEN RUN
2230 POKE 82,2:GRAPHICS 0
2240 END
2299 REM ### choose planet ###
```

```
2300 GRAPHICS 2:SETCOLOR 2,0,0
2310 ? #6;"YOUR MISSION IS TO"
2320 ? #6;"LAND safely ON A    PLANET."
2330 ? #6;"YOU CAN FIRE FUEL TO";
2340 ? #6;"CONTROL THE LANDING."
2350 ? #6;"0-9 VERTICAL AND"
2360 ? #6;"< OR > HORIZONTAL."
2370 ? #6:? #6;"WHICH PLANET ?";
2380 ? "0:RANDOM     1:MERCURIUS  2:VENUS
"
2390 ? "3:EARTH      4:MOON       5:MARS"
2400 ? "6:JUPITER    7:SATURNUS   8:URANU
S"
2410 ? "9:NEPTUNUS                ???";
2420 GET #2,PL:PL=PL-48
2430 IF PL=0 THEN PL=INT(9*RND(0)+1)
2440 FOR I=1 TO PL
2450 READ PL$,COLOR,M1,R0,F0,H0,EFF,HEFF
2460 NEXT I
2470 RETURN
2480 REM ### planet data ###
2500 DATA MERCURIUS,1,3.29E23,2.24E6,160
,350,10000,5000000
2510 DATA VENUS,2,4.87E24,6.05E6,200,500
,10000,5000000
2520 DATA EARTH,9,5.97E24,6.37E6,200,600
,10000,5000000
2530 DATA MOON,5,7.35E22,3.8E5,200,500,3
00000,3000000
2540 DATA MARS,3,6.39E23,3.4E6,200,500,1
0000,5000000
2550 DATA JUPITER,11,1.9E27,7.14E7,200,2
500,20000,50000000
2560 DATA SATURNUS,4,5.69E26,6E7,200,200
0,10000,20000000
2570 DATA URANUS,0,8.72E25,2.6E7,160,400
,10000,10000000
2580 DATA NEPTUNUS,13,1E26,2.42E7,180,28
0,10000,10000000
2590 REM ### initialize I ###
2600 WGT=2500:SG=10:LANVEL=-10
2610 C$="          <ESC>[+]<ESC>[+]<ESC>
[+]<ESC>[+]<ESC>[+]<ESC>[+]<ESC>[+]<ESC>
```

```
[+]<ESC>[+]<ESC>[+]"
2620 J=6.67E-11:JM=J*M1
2630 RETURN
2699 REM ### plot planet ###
2700 GRAPHICS 7:SETCOLOR 0,COLOR,6:SETCO
LOR 1,10,7:SETCOLOR 2,1,8
2710 BASE=INT(RND(0)*130)+5
2720 H=INT(RND(0)*20)+59
2730 COLOR 1:X1=0:X2=BASE-1:GOSUB 2830
2740 COL=2:HB=H:I=H:K=H+2
2750 COLOR COL:PLOT BASE+20,K
2760 DRAWTO BASE+20,I:DRAWTO BASE,I
2770 POSITION BASE,K
2780 POKE 765,COL:XIO 18,#6,0,0,"S:"
2790 IF COL=2 AND H<77 THEN COL=1:I=H+3:
K=79:GOTO 2750
2800 X1=BASE+21:X2=159
2810 COLOR 1:GOSUB 2830
2820 RETURN
2830 FOR I=X1 TO X2
2840 PLOT I,H:DRAWTO I,79
2850 IF H>79 THEN H=78:GOTO 2880
2860 IF H<59 THEN H=60:GOTO 2880
2870 H=H+INT(RND(0)*3)-1
2880 NEXT I
2890 RETURN
2899 REM ### initialize II ###
2900 TIME=0:INT=0.5:GAS=0:HG=0:FUEL=F0
2910 VEL=0:OVEL=0:HGT=H0:OHGT=H0
2920 R=R0+HGT:M=WGT+FUEL*SG
2930 ? "<ESC>[<CLEAR>]    *****   ";PL$;
"   *****"
2940 ? "<ESC>[=]    LEVEL (1 OR 2): ?";
2950 GET #2,LEV:IF LEV<49 OR LEV>51 THEN
 2930
2960 HV=SQR(JM/R)*(LEV=50):OHV=HV
2970 HSC=5*SQR(JM/R)/160
2980 DIS=-HSC*(BASE+5):ODIS=DIS
2990 SC=HGT/(HB-10)
3000 POKE 82,0:POKE 752,1
3010 ? "<ESC>[<CLEAR>]Speed (V.):
   Fuel    :"
```

```
3020 ? "Height    :           Time   :"
3030 ? "Speed (H.):           Weight :"
3040 ? "Distance  :           Trottle:";
3050 RETURN
3099 REM ### root finder ###
3100 NUL=1E+88
3110 PAR=INT1:GOSUB ADR:FUN1=FUN
3120 PAR=INT2:GOSUB ADR:FUN2=FUN
3130 NULO=NUL
3140 NUL=INT1-(INT1-INT2)*FUN1/(FUN1-FUN
2)
3150 PAR=NUL:GOSUB ADR
3160 IF ABS(NUL-NULO)<EPS THEN RETURN
3170 IF FUN*FUN1<0 THEN INT2=NUL:FUN2=FU
N:GOTO 3130
3180 INT1=NUL:FUN1=FUN:GOTO 3130
3199 REM ##### main program #####
3200 GOSUB 2300:REM choose planet #
3210 GOSUB 2600:REM initialize I  #
3220 GOSUB 2700:REM plot planet   #
3230 GOSUB 2900:REM initialize II #
3240 GOSUB 700:REM move ship      #
3250 GOSUB 100:REM input          #
3260 GOSUB 200:REM computation    #
3270 GOSUB 1200:REM check pos.    #
3280 GOSUB 1000:REM output values #
3290 GOTO 3240
```

At the Market

Have you ever wandered through a market and been amazed at the speed with which the salesmen can add up a list of prices? Play this game with your family and find out how good a market trader you would make.

You will see pairs of numbers of increasing length which you have to add up. You will soon find out that this isn't as easy as it sounds. To see why, consider the sum

```
  75856
+ 37637
```

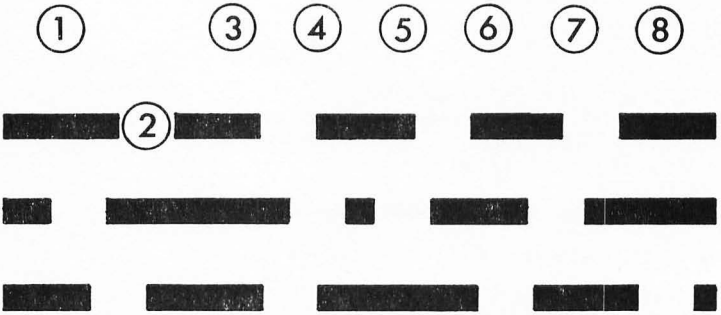
Normally you would add the numbers in the right hand column first, then the column next to it, and so on. With the computer, however, you must enter the answer starting with the left hand column. It is this difference that makes the game so tricky.

How many numbers can you add correctly within the time limit of about one minute? Can you beat our record of eight?

```
10 REM ########## At The Market ##########
20 DIM A$(15)
30 A=1:POKE 19,0:POKE 20,0
40 T1=INT(RND(0)*10^A)
50 IF T1<10^(A-1) THEN 40
60 T2=INT(RND(0)*10^A)
70 IF T2<10^(A-1) THEN 50
80 ? :? T1:? T2;" +"
90 FOR B=0 TO A:PRINT "-";:NEXT B
100 ? :? "Answer :";:INPUT ANS
110 IF ANS=T1+T2 THEN 140
120 ? "Wrong, it is : ";T1+T2
130 A=A-1:GOTO 140
140 IF PEEK(20)+PEEK(19)*256<3600 THEN A
=A+1:GOTO 40
150 T1=INT(RND(0)*10^A):IF T1<10^(A-1) T
HEN 40
160 ? :? "Your score is : ";A
165 ? "Job specification: ";
170 FOR I=1 TO A:READ A$:NEXT I:? A$
180 ? "Time : ";INT((PEEK(20)+PEEK(19)*2
56)/0.6)/100;" seconds"
190 ? "Do you want to play again ";:INPU
T A$
200 IF A$(1,1)="Y" THEN RUN
210 END
220 DATA Try again,Schoolkid,Paper-boy,C
lerk,Auto dealer
230 DATA Broker,Director,Oil magnate,Mil
lionaire,World champion
```

Fallout

At the start of this simple but absorbing game you will see eight horizontal bars with gaps in them. Above the bars are eight checkers which can fall through the gaps. In this example



checker 2 has already fallen into a gap. The object of the game is to get all eight checkers through the bars. You can do this by moving the bars to line up the gaps for the checkers to fall through.

To move a particular bar enter a command of the form

BDS

where

B is the number of the bar (the bars are numbered from 1 at the top to 8 at the bottom)

D is the direction you wish to move it (L for left, R for right) and

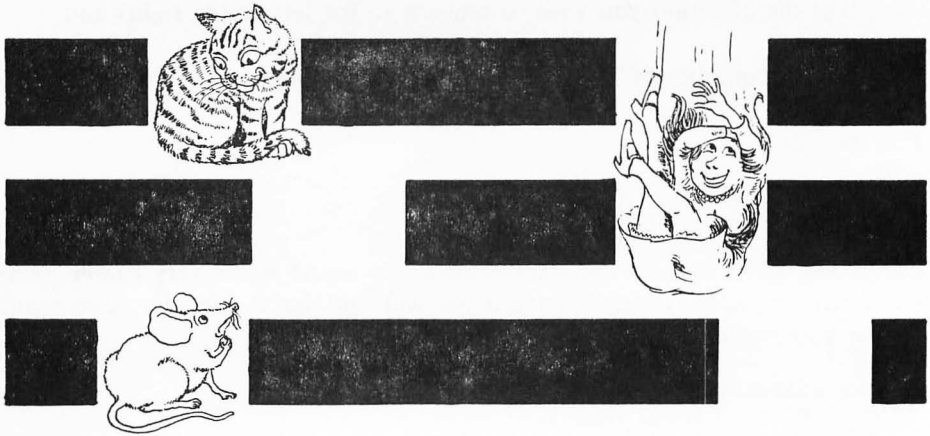
S is the number of steps the bar has to be shifted.

For instance

```
3R12
```

moves bar 3, 12 steps to the right. This might sound like a very simple game but when you actually start playing it you will find that it provides quite a stiff test of your ability to think logically.

```
10 REM ######### F a l l o u t #########
20 DIM G$(153),H$(153),RU$(17)
30 DIM BL$(1),BA$(1),I(8)
40 POKE 752,1
50 GOTO 800
99 REM ### initialize ###
100 BL$=" ":BA$="@ @"
110 G$=BL$:G$(17)=BL$:G$(2)=G$
120 G$(18)=BA$:G$(153)=BA$:G$(19)=G$(18)
130 FOR Y=1 TO 8
140 FOR B=1 TO 4
150 X=INT(RND(0)*17+1)
160 G$(Y*17+X,Y*17+X)=" "
170 NEXT B
180 I(Y)=2*Y:G$(I(Y),I(Y))=STR$(Y)
190 NEXT Y
200 RETURN
299 REM ### draw screen ###
300 H$=G$
310 ? "<ESC>[<CLEAR>]<ESC>[=]SCORE:
    TURN:"
320 POSITION 10,1:? S0;
330 POSITION 26,1:? TU;
340 POSITION 2,5
350 FOR B=0 TO 8
```



```
360 ? B;" ";H$(B*17+1,(B+1)*17)
370 NEXT B
380 RETURN
399 REM ### input ###
400 POSITION 10,22:? "(ESC)(SHIFT)(BACK
S)ENTER MOVE ";
410 INPUT RO$
420 RC=ASC(RO$(1,1))-48
430 IF RC<1 OR RC>8 THEN 400
440 DI=(RO$(2,2)="R")-(RO$(2,2)="L")
450 IF DI=0 THEN 400
460 TI=VAL(RO$(3))
470 IF TI<1 THEN 400
480 RETURN
499 REM ### drop checkers ###
500 FOR B=1 TO 8
510 I=I(B):J=I+17:IF I<0 THEN 630
520 IF H$(I,I)=BL$ OR H$(I,I)=BA$ OR H$(
J,J)<>BL$ THEN 630
530 I(B)=J:H$(J,J)=H$(I,I):H$(I,I)=BL$
535 MI=INT((I-1)/17):MJ=INT((J-1)/17)
540 MI=INT((I-1)/17):MJ=INT((J-1)/17)
```

```
550 POSITION 3+J-17*MJ,MJ+5:? H$(J,J);
560 POSITION 3+I-17*MI,MI+5:? BL$;
570 IF J<137 THEN 620
580 SC=SC+VAL(H$(J,J))
590 I(B)=-1:H$(J,J)=BL$
600 POSITION 10,1:? SC;
610 POSITION 3+J-17*MJ,MJ+5:? BL$
620 B=0
630 NEXT B
640 RETURN
699 REM ### shift bar ###
700 RO$=H$(17*RC+1,17*(RC+1))
710 IF DI=1 THEN H$(17*RC+1,17*RC+1)=RO$
(17):H$(17*RC+2,17*(RC+1))=RO$(1,16):GOT
O 730
720 H$(17*(RC+1),17*(RC+1))=RO$(1,1):H$(
17*RC+1,17*(RC+1)-1)=RO$(2)
730 POSITION 2,5+RC:? RC;" ";H$(17*RC+1,
17*(RC+1))
740 FOR B=1 TO 8:R=INT((I(B)-1)/17)
750 IF RC<>R THEN 780
760 IF I(B)=RC*17+8*DI+9 THEN I(B)=RC*17
-9*DI+9
770 I(B)=I(B)+DI
780 NEXT B
790 RETURN
799 REM ##### main program #####
800 GOSUB 100:REM initialize          #
810 GOSUB 300:REM draw screen         #
820 RC=1:GOSUB 500:REM drop checkers#
830 GOSUB 400:REM input               #
840 FOR DU=1 TO TI
850 GOSUB 700:REM shift bar           #
860 GOSUB 500:REM drop checkers       #
870 NT=NT+1:POSITION 26,1:? NT;
880 NEXT DU
890 IF SC<36 THEN 830
900 SC=0:NT=0
910 POSITION 2,22:? "<ESC><SHIFT><BACK S
>Once more ";
920 INPUT RO$:IF RO$(1,1)<>"Y" THEN GRAP
HICS 0:END
```

```
930 ? "<ESC><SHIFT><BACK S>ame situatio
n ";:INPUT RO$
940 IF RO$(1,1)<>"Y" THEN RUN
950 H$=G$:GOTO 810
1499 REM ##### main program #####
1500 GOSUB 100:REM initialize #
1510 GOSUB 200:REM draw screen #
1515 RC=1:GOSUB 300:REM
1520 GOSUB 600:REM input #
1530 FOR DU=1 TO TI
1540 GOSUB 400:REM shift
1542 GOSUB 300:REM drop
1550 NT=NT+1:POSITION 26,1:? NT;
1560 NEXT DU
1570 IF SC<36 THEN 520
1580 END
```

Ship's Attack

First, let's describe this game as realistically as possible. In the lower half of the screen is a shape like this:

Little squares fall down from the top of the screen. You must move the shape using the joystick or the cursor controls .

to stop the squares hitting it. If that sounds dull what about this . . .

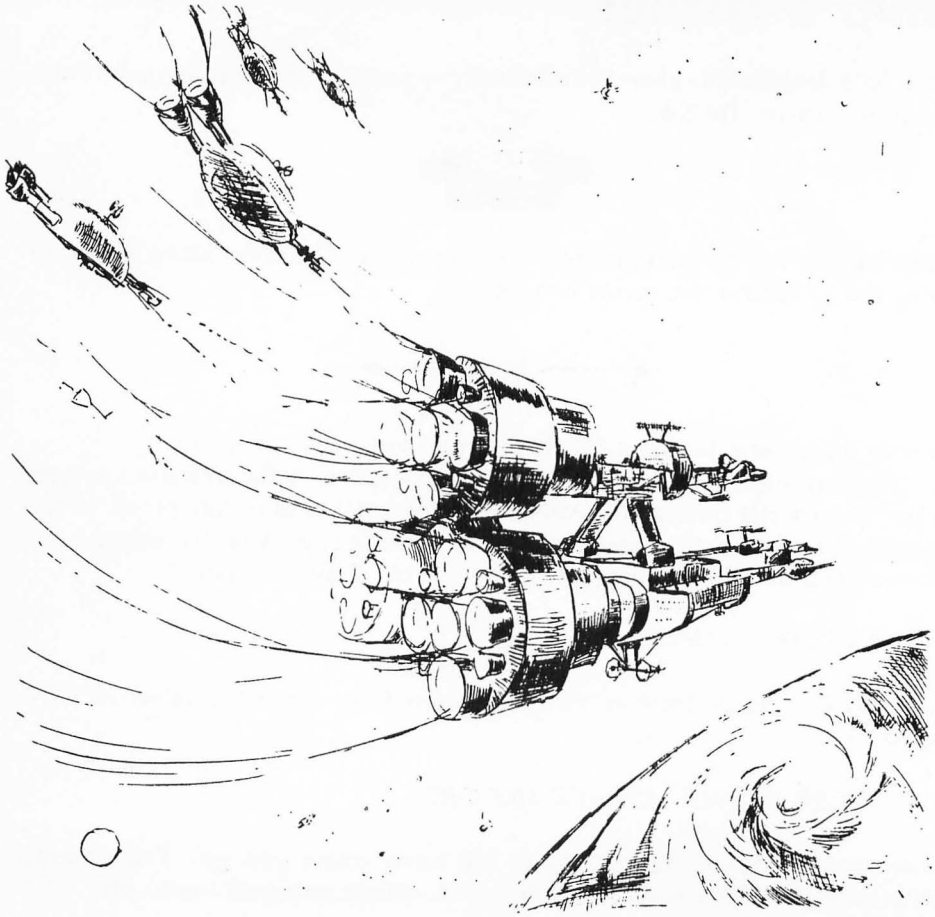
You are captain of one of the finest spaceships in the universe. As you cruise majestically through the Milky Way you can't help feeling proud of the magnificent vessel under your command. Then just as you are nearing the harbour and the end of your voyage is in sight, disaster strikes —

A FLYING SAUCER ATTACK!

How can you escape these unfriendly invaders from across the universe? Why not try your

SUPER ATOMIC ESCAPE MOTOR?

The more flying saucers you avoid, the more points you get. That sounds better, doesn't it? Anyway, the game is fast, simple and great fun to play, and that is all that matters.

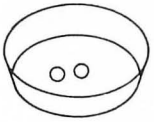


```
10 REM ######## Ships's Attack #########
20 DIM SH$(3),NS$(3),BU$(1),NB$(1)
30 OPEN #2,4,0,"K:"
40 GOTO 500
99 REM ### initialization ###
100 SH$="@[KUL]@":NS$="   ":BU$="[T]":NB
$=" "
110 PO=0:OPL=9:PL=10:BY=23
120 ? "<ESC>[<CLEAR>]<ESC>[=]Enter level
 : ";:INPUT LE
130 GRAPHICS 17:POKE 756,226
140 SETCOLOR 0,0,0
150 ? #6;"    @SHIPS@ @ATTACK@"
160 POSITION PL-1,23:? #6;SH$;
170 RETURN
199 REM ### move ship ########
200 OPL=PL:K=0
210 IF PEEK(764)<>255 THEN GET #2,K
220 PL=PL+(K=42)-(K=43)+(STICK(0)=7)-(ST
ICK(0)=11)
230 IF PL=OPL THEN RETURN
240 POSITION OPL-1,23:? #6;NS$;
250 IF PL<1 THEN PL=1
260 IF PL>18 THEN PL=18
270 POSITION PL-1,23:? #6;SH$;
280 RETURN
299 REM ### drop bullet ###
300 POSITION BX,BY:? #6;NB$;
310 BY=BY+1+LE
320 IF BY<23.5 THEN 390
330 IF ABS(BX-PL)>1 THEN 340
335 FOR W=50 TO 250:SOUND 0,W,0,15:NEXT
W:SOUND 0,0,0,0:? "<ESC>[<CLEAR>]<ESC>[=
]You have ";PO-1;" points.":END
340 BX=PL+INT((3-LE)*(RND(0)*3-1))
350 IF BX<1 THEN BX=1
360 IF BX>18 THEN BX=18
370 BY=LE:PO=PO+1
380 FOR W=100 TO 50 STEP -3:SOUND 0,W,10
,15:NEXT W:SOUND 0,0,0,0
390 POSITION BX,BY:? #6;BU$;
400 RETURN
```

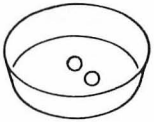
```
499 REM ##### main program #####
500 GOSUB 100:REM initialization #
510 GOSUB 200:REM move ship      #
520 GOSUB 300:REM drop bullet    #
530 GOTO 510
```

Mini Mancala

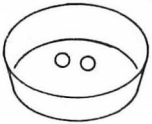
MINI MANCALA is based on an old Arabian game. It is played by moving stones between cups. There are four cups: A and B are the computer's, and C and D are yours. At the start of the game there are two stones in each cup.



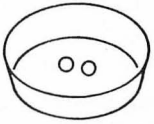
A (computer)



B (computer)

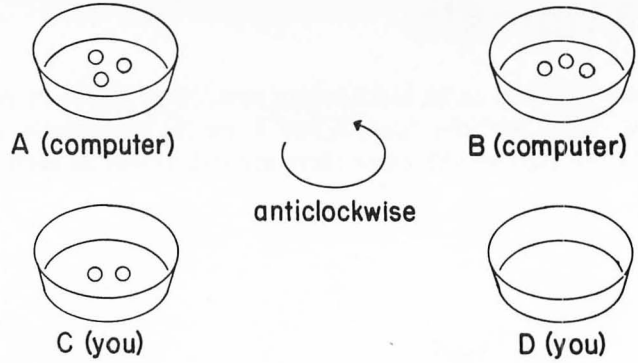


C (you)



D (you)

In turn, the players take the stones from one of their own cups and distribute them counterclockwise to the other three cups, one stone per cup. For instance you might choose to move the stones from cup D like this:



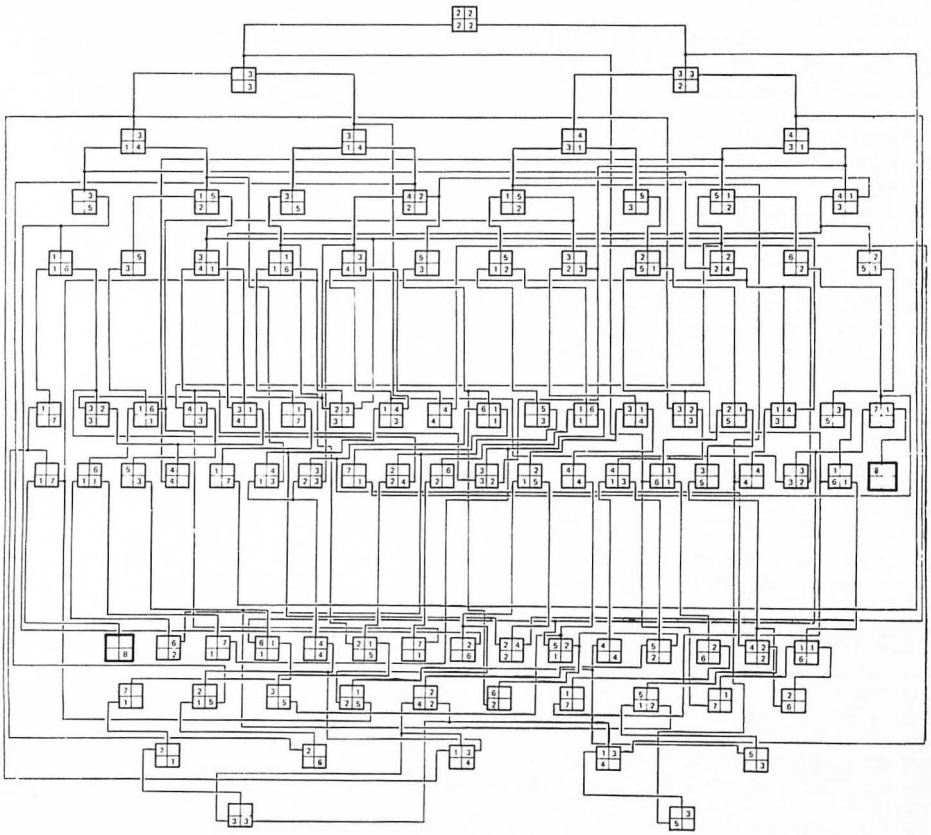
After this move cup D is empty. In fact, there will always be at least one empty cup, because during each move stones cannot be put back into the cup from which they were taken.

To win the game you must get all the stones into your own cups.

On the computer, the cups are represented by squares with numbers on them indicating the number of stones they contain. You can choose the level of difficulty you prefer, 1, 2, or 3 (1 being the easiest), and who has the first move. The computer will ask which of your squares you wish to move the stones from and will tell you what its own move is. The position of the stones on the board is displayed after each move.

You may be surprised that this complicated game can be described in such a short program. The diagram shows all the possible moves, and is an excellent example of how a strategic game can be represented schematically.

This game was created by C. Freeling.

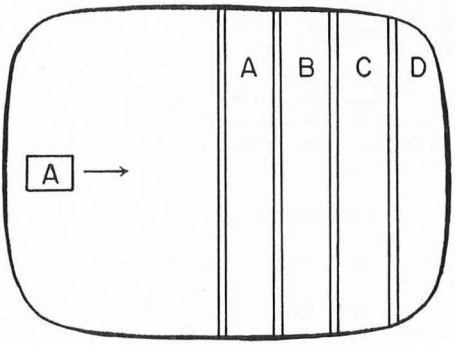


```
10 REM ######### Mini-Mancala #########
20 DIM MK(1,1),L(3,1)
30 OPEN #2,4,0,"K:"
40 GOTO 700
99 REM ### initialize ###
100 FOR I=0 TO 1
110 MK(0,I)=2:MK(1,I)=2
120 FOR L=0 TO 3
130 READ X:L(L,I)=X
140 NEXT L
150 NEXT I
160 DATA 0,1,1,0,0,0,1,1
170 RETURN
199 REM ### display board ###
200 GRAPHICS 18
210 POSITION 4,0:? #6;"mini[M]mancala"
220 ? #6:? #6:? #6;"      @A-------B@"
230 ? #6;"       @!@2 @!@ 2@!@"
235 ? #6;"       @!-----!@"
240 ? #6;"       @!@2 @!@ 2@!@"
250 ? #6;"      @C-------D@"
260 RETURN
299 REM ### input ###
300 POSITION 0,10
310 ? #6;"@MOVE STONES FROM CUP (C,D) ?@
";
320 POKE 764,255:GET #2,X
330 IF X=67 AND MK(1,0)>0 THEN L=1:? #6;C
HR$(X):RETURN
340 IF X=68 AND MK(1,1)>0 THEN L=2:? #6;C
HR$(X):RETURN
350 SOUND 0,20,10,15
360 FOR T=1 TO 100:NEXT T
370 SOUND 0,0,0,0
380 GOTO 300
399 REM ### computer's turn ###
400 POSITION 0,10:? #6;"now it[6]s my tu
rn[A]                  "
410 FOR W=0 TO 500:NEXT W
420 Q=MK(0,0)*1000+MK(0,1)*100+MK(1,0)*10+M
(1,1)
430 L=3
```

```
440 IF MK(0,0)>0 AND (LEV<3 AND RND(0)*LE
V<0.4 OR MK(0,1)=0 OR Q=1430 OR Q=1340 OR
 Q=6110 OR Q=1160) THEN L=0
450 POSITION 0,10:? #6;"@I MOVE STONES F
ROM  CUP @";CHR$(65+(L=3));
460 RETURN
499 REM ### spread stones ###
500 G=L(L,0):H=L(L,1)
510 IF MK(G,H)=0 THEN RETURN
520 L=L+1:IF L>3 THEN L=L-4
530 I=L(L,0):J=L(L,1)
540 IF I=G AND J=H THEN 520
550 GOSUB 600:GOTO 510
599 REM ### move one stone ###
600 MK(G,H)=MK(G,H)-1:MK(I,J)=MK(I,J)+1
610 POSITION 7+4*H,4+2*G:? #6;MK(G,H)
620 POSITION 7+4*J,4+2*I:? #6;MK(I,J)
630 FOR W=1 TO 200:NEXT W
640 RETURN
699 REM ##### main program #####
700 GOSUB 100:GOSUB 200
710 ? #6:? #6:? #6;"ENTER LEVEL (1-3)?"
720 GET #2,X:LEV=X-48
730 IF LEV<1 OR LEV>3 THEN 720
740 POSITION 0,10:? #6;"WHO STARTS? YOU
OR ME ?"
750 GET #2,X:IF X=89 THEN 790
760 GOSUB 300:REM input              #
770 GOSUB 500:REM spread stones      #
780 IF MK(1,1)=8 THEN POSITION 0,9:? #6;"
you win":GOTO 830
790 GOSUB 400:REM computer's turn #
800 GOSUB 500:REM spread stones      #
810 IF MK(0,0)=8 THEN POSITION 0,9:? #6;"
I win":GOTO 830
820 GOTO 760
830 ? #6;"DO YOU WANT TO PLAY AGAIN ?
    "
840 POKE 764,255:GET #2,X:IF X=89 THEN R
UN
850 END
```

Stop It!

Although this is only a short program it gives rise to a fast and exciting game. The screen looks like this



The square on the left has a letter on it. When it moves across the screen you must try and stop it in the region bearing the same letter by pressing any key.

At the start of the game the computer requests a level of difficulty (1 is the easiest, 2 is harder and 3 is the most difficult) and asks you how many times you want to play. You'll be shown your score at the end, for instance

```
Your score is 0 out of 10 (0%)
```

Never mind, better luck next time!

```
10 REM ######### S t o p   I t #########
20 DIM BU$(1)
30 ? "(ESC)[(CLEAR)](ESC)[(=)]Enter leve
l of difficulty (1-3):";:INPUT DI:DT=25-
6*DI
40 ? "(ESC)[(=)]Enter number of turns :"
;:INPUT NT
50 GRAPHICS 18:GOSUB 600
60 SETCOLOR 0,4,6
70 SETCOLOR 1,0,15
80 SETCOLOR 2,12,10
90 POSITION 11,0:? #6;"@[a[b[c[d@"
100 FOR RO=1 TO 11
110 POSITION 11,RO:? #6;"@[ [ [ [@"
120 NEXT RO
130 TU=TU+1
140 CO=INT(RND(0)*4):BU$=CHR$(CO+65+32)
150 RO=1+2*INT(RND(0)*6)
155 POKE 764,255
160 FOR C=0 TO 18
170 POSITION C,RO:? #6;" ";BU$;
180 FOR DE=1 TO DT:NEXT DE
190 IF PEEK(764)<>255 OR PTRIG(1)=0 OR S
TRIG(0)=0 THEN C=C+1:GOTO 210
200 NEXT C
210 POSITION C,RO:? #6;" ";
215 POSITION 11,RO:? #6;"@[ [ [ [@"
220 IF C<>12+2*CO THEN T=200:GOSUB 500:G
OTO 250
230 T=20:GOSUB 500:HI=HI+1
240 POSITION 0,0:? #6;HI;
250 IF TU<NT THEN 130
260 GRAPHICS 0:POSITION 5,5
270 ? "Your score is ";HI;" out of ";NT;
" (";INT(HI/NT*100);"%)"
280 END
500 SOUND 0,T,10,15
510 FOR DE=1 TO 5*DT:NEXT DE
520 SOUND 0,0,0,0
530 RETURN
599 REM ### change characterset ###
600 TOP=PEEK(106)-8
```

```
610 POKE 204,TOP:POKE 206,224
620 FOR X=1536 TO 1555
630 READ V:POKE X,V
640 NEXT X
650 Q=USR(1536)
660 DATA 104,162,4,160,0,177,205,145,203
,200,208,249,230,206,230,204,202,208,242
,96
670 RAMSET=TOP*256
680 FOR X=RAMSET+59*8 TO RAMSET+59*8+7
690 POKE X,255
700 NEXT X
710 POKE 756,TOP
720 RETURN
```

The Swedish Popsong

A *theme con variatone* is a tune that, although it is based on one that has been heard before, has its own mood and identity. This game could be considered as a *theme con variatone* as it has some similarities with another game in this book, but presents its own unique challenge to the solver.

You will see 16 fields filled with a random arrangement of As and Bs e.g.

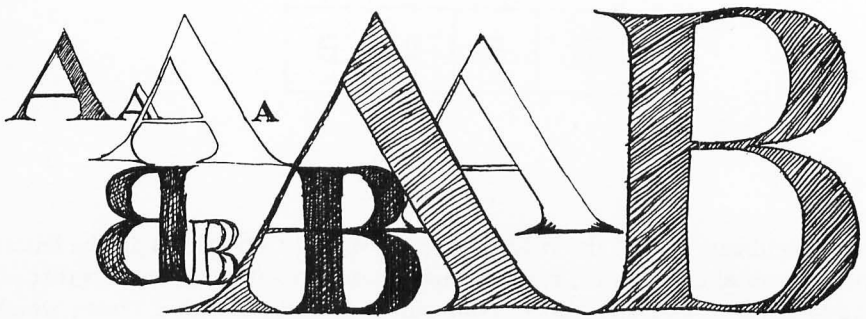
B	A	B	A
A	A	A	B
A	B	A	B
B	B	A	B

When you indicate one of the fields (enter ROW and COLUMN) all the letters on the horizontal and vertical rows through that field will be altered so that all the As become Bs and vice versa. Your aim is to end up with a screen which

looks like this:

A	B	B	A
A	B	B	A
A	B	B	A
A	B	B	A

```
10 REM ###### BaBa ######
20 DIM B(4,4)
30 GRAPHICS 18
40 OPEN #2,4,0,"K:"
50 GOTO 500
99 REM ### initialize ###
100 FOR R=1 TO 4
110 FOR C=1 TO 4
```



```
120 B(R,C)=INT(2*RND(0))
130 NEXT C
140 NEXT R
150 ? #6;" @the swedish popsong@"
160 POSITION 8,2:? #6;"1234"
170 POSITION 9,9:? #6;CHR$(159)
180 RETURN
199 REM ### draw screen ###
200 FOR R=1 TO 4
210 POSITION 6,3+R:? #6;R;" ";
220 FOR C=1 TO 4
230 IF B(R,C) THEN ? #6;"b";:GOTO 250
240 ? #6;"a";
250 NEXT C
260 NEXT R
270 RETURN
299 REM ### get input and flip ###
300 POSITION 2,9:? #6;"@row   @"
310 GET #2,R:R=R-48
320 IF R<1 OR R>4 THEN 300
330 FOR C=1 TO 4
340 B(R,C)= NOT B(R,C)
350 NEXT C
360 POSITION 2,9:? #6;"@column@"
370 GET #2,C:C=C-48
380 IF C<1 OR C>4 THEN 360
390 B(R,C)= NOT B(R,C)
400 FOR R=1 TO 4
410 B(R,C)= NOT B(R,C)
420 NEXT R
430 RETURN
499 REM ##### main program #####
500 GOSUB 100:REM initialize   #
510 GOSUB 200:REM draw screen  #
520 GOSUB 300:REM input & turn #
530 GOTO 510
```

Vowels and Consonants

This competition game can be played by the whole family. Unlike most other computer games, it does not involve numbers or arithmetic. At the start of the game enter the number of players, and then take turns to play. The computer will give you seven letters: when it asks you

Vowel or Consonant (V,C)?

enter V for a vowel or C for a consonant. When you have your seven letters the computer challenges you to make as long a word as possible out of them, using each letter once only. There is a time limit, shown on a clock on the screen. It is amazing how addictive a simple game like this can become. Try it and see!

```
10 REM ##### Vowels And Consonants #####
20 OPEN #2,4,0,"K:"
30 ? "<ESC>[<CLEAR>]<ESC>[<=>]Number of
Players :";:INPUT NP
40 DIM PO(NP),CHS$(21),CH$(1),LE$(7)
50 FOR P=1 TO NP:PO(P)=0:NEXT P
60 GOTO 900
99 REM ### Initialize screen ###
100 GRAPHICS 18
110 ? #6;"PLR.:":? #6;"PTS.:"
120 FOR P=1 TO NP
130 POSITION 4+3*P,0:? #6;P
140 POSITION 3+3*P,1
150 IF PO(P)<10 THEN ? #6;" ";
160 ? #6;PO(P)
170 NEXT P
180 POSITION 0,3:? #6;"turn of player[Z]
 ";CHR$(PL+16)
190 POSITION 0,5:? #6;"@vowel or consona
nt  [H]V[L]C[I]@";CHR$(159);
200 POSITION 0,11
210 ? #6;"@time left[Z]      sec[N]@";
```

```
220 POKE 20,0:POKE 19,0:POKE 18,0
230 RETURN
299 REM ### Clock ###
300 TI=(PEEK(18)*65536+PEEK(19)*256+PEEK
(20))/60
310 POSITION 11,11
320 ? #6;60-INT(TI*10)/10
330 IF TI>60 THEN POSITION 0,11:? #6;"ti
me is up! 0 pts.  ";:SOUND 0,250,10,15:P
OP :RETURN
340 RETURN
399 REM ### choose characters ###
400 FOR DU=1 TO 7
410 GOSUB 300
420 IF PEEK(764)=255 THEN 410
430 GET #2,IN
440 IF IN=86 THEN CHS$="AEIOU":GOTO 470
450 IF IN=67 THEN CHS$="BCDFGHJKLMNPQRST
VWXYZ":GOTO 470
460 GOTO 410
470 IN=INT(RND(0)*LEN(CHS$)+1)
480 LE$(DU)=CHS$(IN,IN)
490 POSITION 5+2*DU,6
500 ? #6;CHR$(ASC(LE$(DU))+32)
510 NEXT DU
520 RETURN
599 REM ### input word ###
600 LE=0
```

```
610 GOSUB 300
620 IF PEEK(764)=255 THEN 610
630 GET #2,IN
640 IF IN=155 THEN RETURN
650 IF IN=126 AND LE>0 THEN POSITION LE-
1,8:? #6;" ";:LE=LE-1:GOTO 610
660 IF IN<65 OR IN>90 THEN 610
670 LE=LE+1:CHS$(LE)=CHR$(IN)
680 POSITION LE-1,8:? #6;CHR$(IN+160)
690 GOTO 610
699 REM ### compute points ###
700 IF LE=0 OR TI>60 THEN RETURN
710 FOR CO=1 TO LE
720 FOR C=1 TO 7
730 IF CHS$(CO,CO)=LE$(C,C) THEN LE$(C,C
)="<ESC><BACK S>GOTO 780"
740 NEXT C
750 POSITION 0,9:SOUND 0,200,10,15
760 ? #6;"wrong letters used [A]"
770 RETURN
780 NEXT CO
790 PO(PL)=PO(PL)+LE
800 POSITION 0,9
810 IF LE<7 THEN ? #6;"good[K] ";:SOUND
0,30,10,15:GOTO 830
820 ? #6;"eccelent[K] ";:SOUND 0,10,10,1
5
830 ? #6;CHR$(LE+16);" points[N]"
840 RETURN
899 REM ##### main program #####
900 FOR PL=1 TO NP
910 GOSUB 100:REM initialize screen #
920 GOSUB 400:REM choose characters #
930 POSITION 0,5
940 ? #6;"@MAKE A WORD OUT OF  THESE:@";
950 GOSUB 600:REM input word        #
960 GOSUB 700:REM compute points    #
970 FOR W=1 TO 500:NEXT W
980 SOUND 0,0,0,0
990 NEXT PL
1000 GOTO 900
```

Astrology

This program is based on a study made by the Dutch physicist and astrologer Dr. Ir. J. Van Slooten. He was a research worker at Philips laboratories who spent all his free time on astrology. He developed a theory that the phase of the moon at the time of birth was a very important astrological influence on a person's character. After studying the lives of hundreds of people he concluded, " . . . that the moon phase forecasts the extent to which the spiritual and emotional life, especially with respect to a person's social environment, will develop and furthermore the role he or she will play in our society . . . "

The diagram shows the cycle of the moon's phases. As there are three phases, waxing, full, and waning, so three types of person can be defined.

Individualists: born in the waxing moon phase, they have waxing energy. They like to work on their own, have strong wills, and are not discouraged by physical discomfort

Socialists: born in the full moon phase, they like to live in communities. Key words for these people are 'compare', 'choose' and 'combine'. Cooperation is a dominant factor but there is also rivalry and envy

Conservatives: born in the waning moon phase, they know that the light decreases but will return again. They are careful and sure of themselves, and like to have everything under control

These three types can be further subdivided to give twelve categories in all:

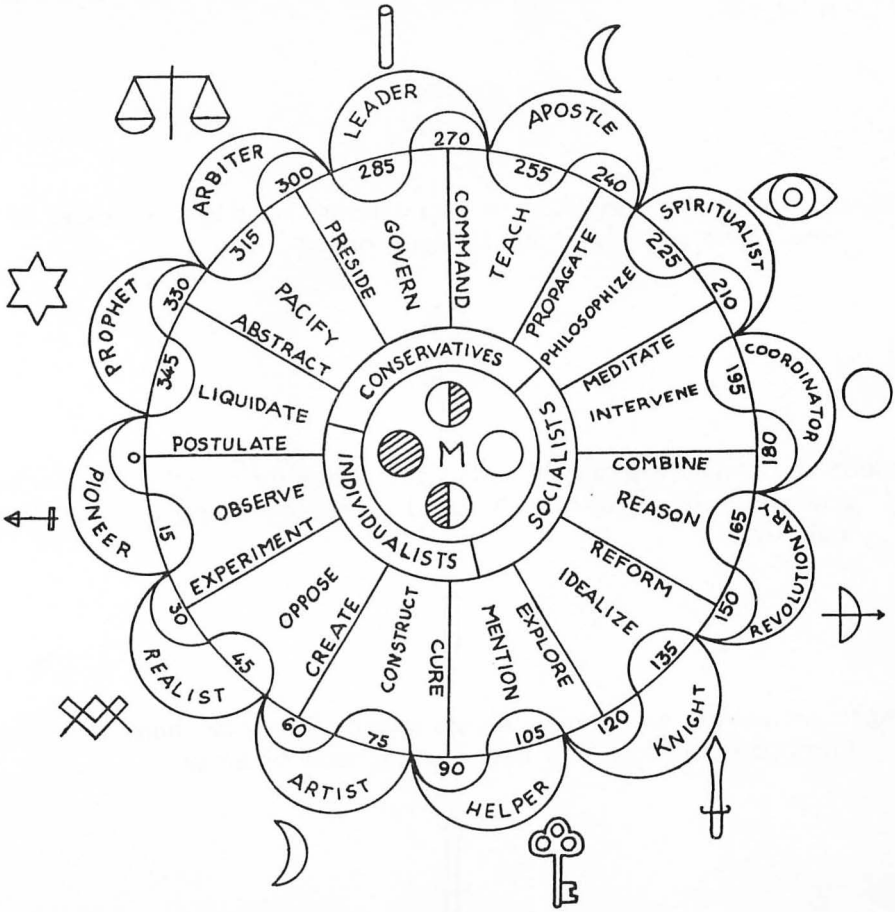
Pioneers are searchers, always ahead of the crowd. Their strength of purpose can sometimes make them appear stubborn

Coordinators like to be with other people, but feel that they are'more equal' than their companions. They are romantic and strongly attracted to family life

Realists look at the world in a very down-to-earth manner. They like to think that everything can be explained in purely physical terms

Spiritualists are rather 'other worldly' characters. Philosophical in outlook, they are seekers after the unknown

Artists transform material in a creative manner. They can produce works of art, new products, or anything that did not exist before


LEADER
APOSTLE
ARBITER
SPIRITUALIST
PROPHET
COORDINATOR
PIONEER
REVOLUTIONARY
REALIST
KNIGHT
ARTIST
HELPER
CONSERVATIVES
SOCIALISTS
INDIVIDUALISTS
M
270
285
255
240
300
225
315
210
330
195
345
180
0
165
15
150
30
135
45
120
60
105
75
90
COMMAND
TEACH
PRESIDE
GOVERN
PROPAGATE
PHILOSOPHIZE
PACIFY
ABSTRACT
MEDITATE
INTERVENE
LIQUIDATE
POSTULATE
COMBINE
REASON
OBSERVE
EXPERIMENT
REFORM
IDEALIZE
OPPOSE
CREATE
EXPLORE
MENTION
CONSTRUCT
CURE

Apostles are not themselves creative but spread the ideas of others to the world at large

Helpers take on responsibilities to their fellows that might otherwise be ignored. They are helpful and caring to others

Leaders also assume responsibilities but in this case for directing the lives and actions of others. They look forward where most people would prefer to look back

Knights are fearless adventurers, always searching for new things to explore. Idealistic in outlook, they have a strong sense of honor

Arbiters like to preside over the actions of others. they do not prejudge issues, but when they reach a decision they expect it to be obeyed

Revolutionaries are dissatisfied with the world as it is. They tend to look on the bad side of things, and have a desire to change the world

Prophets consider the world 'from above' bringing to bear the wisdom of the past. They may not always be listened to, of course

The program will tell you the phase of the moon on the day you were born. From this you can discover the secrets of your character!

```
10 REM ###### Moonphase Astrology ######
20 GRAPHICS 18
30 OPEN #2,4,0,"K:"
40 GOTO 500
99 REM ### read data ###
100 NUM=0
110 GET #2,X:? #6;CHR$(X);
120 IF X=155 THEN RETURN
130 NUM=NUM*10+(X-48)
140 GOTO 110
199 REM ### INPUT DATA ###
200 ? #6;"  @*% moonphase *%@";
210 POSITION 5,2:? #6;"ENTER DATE "
220 POSITION 0,4:? #6;"DAY      DD? ";
230 GOSUB 100:DD=NUM
240 ? #6;"MONTH    MM? ";
250 GOSUB 100:M=NUM
260 ? #6;"YEAR   YYYY? ";
270 GOSUB 100:Y=NUM
280 IF M<3 THEN M=M+12:Y=Y-1
290 RETURN
299 REM ### calculation ###
300 T=INT(365.25*Y)+INT(30.6*(M+1))
310 T=(T+DD-694038)/36525
320 LA=350.737486+1236*T*360+307*T
330 LA=LA+6*T/60+51.18*T/3600
340 LA=LA-5.17*T*T/3600
350 LA=LA-INT(LA/360)*360
360 LA=INT(LA+0.5)
370 RETURN
399 REM ### display result ###
400 POSITION 0,8
410 ? #6;"MOONPHASE = ";LA;" DEG.  "
420 RETURN
430 NUM=NUM*10+(X-48)
440 GOTO 410
499 REM ##### main program #####
500 GOSUB 200:REM input data      #
510 GOSUB 300:REM calculation     #
520 GOSUB 400:REM display result  #
530 GET #2,X:RUN
```

> > $5.95 FPT
USA

Does your brain get enough exercise? Are your wits as sharp, or your reactions as quick, as they should be? Tone up the muscles of your mind with this enthralling collection of arcade and action games, puzzles, brainteasers and competitions for your microcomputer.

Battle with GALACTIC MONSTERS and ZOMBIES IN THE SWAMP, defend yourself against a SHIP'S ATTACK, and set out on a nerve-jangling SHARK HUNT.

If you survive all that you can gamble at LAS VEGAS A GO GO and the KENTUCKY DERBY, plumb the mysteries of the BLACK BOX, and stretch your mind with intriguing board games like ONE TO FIVE, SHAKESPEARIAN SHUFFLE, and MINI MANCALA. Then,test your intellectual capabilities with VOWELS AND CONSONANTS, KEYBOARD MEMORY, and AT THE MARKET. Will you turn out to be a GENIUS AT WORK?

By now you'll be razor sharp and ready for anything, which is just as well because you are not even halfway through yet! There's even more excitement, skill and fun for you, your family, and friends. So test your reflexes and mental prowess on this extraordinary obstacle course.

What are you waiting for . . . on your marks, get set . . . GO!

ADDISON-WESLEY PUBLISHING COMPANY

ISBN 0-201-16477-9