# GIANT BOOK OF GAMES FOR YOUR DRAGON

TIM HARTNELL
ALAN BLACKMAN
ROGER BUSH
ROBERT YOUNG

A FONTANA ORIGINAL

FONTANA

# Giant Book of Games for Your Dragon

Tim Hartnell
Alan Blackman
Roger Bush
Robert Young

# GIANT BOOK OF GAMES FOR YOUR DRAGON

Fontana Paperbacks

First published in 1984 by Fontana Paperbacks,
8 Grafton Street, London W1X 3LA



Made and printed in Great Britain by
Hazell, Watson and Viney Limited,
Member of the BBCC Group, Aylesbury, Bucks

To Dianne and all the family,
for their wholehearted support

# Contents

# Introduction

In this collection of games for the Dragon 32, we have tried to cater for as many different tastes as possible. We have chosen games of all types: arcade-style games, dice games, card games, games which use the high resolution graphics to great effect, simulation games, and even several adventure games. All the games in this book have been designed to provide maximum interest and excitement. We have included instructions and hints for playing each game, and, where appropriate, notes explaining the construction of the program.

No computer program, especially a game program, is ever truly finished. There is always some fine tuning or highlights which can be added. We present these games to you not as a finished product but as a starting point for your own ideas. Please feel free to adapt them to your own use. Add your own personal touches to them, more sound or graphic displays, whatever takes your fancy. Most important of all, make sure you enjoy playing them.

Tim Hartnell
Alan Blackman
Roger Bush
Robert Young

# Arcade Games

# Crazy Highway

Crazy Highway is an action game which makes good use of the high resolution graphics of your computer.

You play the part of a curious cat who is trying to cross a super highway. The highway is shown in red on the screen and the cat is displayed as an "H" shape. Whizzing up and down the highway are a number of green cars. Each time you are hit by a car you lose one of your nine lives. The game finishes when you have lost all nine lives or if you successfully cross the road twenty times. Once you start across the road you cannot turn back, you can only pause to let a car go past. You cannot go back the other way until you have reached the other side of the highway.

The game begins with only one car on the highway, then each time you are run over another car appears. A word of warning: watch out for stalled cars.

The game has two versions. The first is a keyboard version. The "A" key moves your cat to the left and the "L" key moves him to the right.

The second version is for joystick control. The joystick should be plugged into the right joystick port.

```
1 REM KEYBOARD VERSION
10 REM CRAZY HIGHWAY
20 PMODE 3,1:CLS:PCLS0
30 GOSUB 670
40 GOSUB 240:GOSUB 320:REM STORE CHARACT
ERS
50 PUT (0,0)-(15,15),BL,PSET
60 SCREEN 1,0:WP=8:FOR I=1 TO 10:FL=1
```

```
70 GOSUB 360:REM MOVE CAT
80 GOSUB 440:REM MOVE VEHICLE
90 IF WP*FL>240 OR (WP<9 AND FL=-1) THEN
 100 ELSE 70
100 Z=Z+1
110 IF FL=1 THEN FL=-1:GOTO 70
120 IF I>3 THEN Q=INT(I/2)+1
130 NEXT I
140 CLS:GOTO 160
150 CLS:PRINT:PRINT"YOU HAVE LOST YOUR N
INE LIVES!!":PRINT:PRINT
160 PRINT:PRINT:PRINT:PRINT:PRINT"  YOU
HAVE BEEN HIT";XX;"TIMES!!":PRINT:PRINT"
  YOU HAVE CROSSED THE ROAD";Z:PRINT TAB
(12) "TIMES!!"
170 PRINT:PRINT:PRINT "DO YOU WISH TO PL
AY AGAIN?(Y/N)"
180 A$=INKEY$
190 IF A$="" THEN 180
200 IF A$="Y" THEN RUN
210 IF A$="N" THEN CLS:PRINT@ 235, "GOOD
BYE":GOTO 230
220 GOTO 180
230 GOTO 230
240 COLOR 2,0:DIM WA(2):DIM BL(7):DIM BW
(2):DIM HE(9):Q=1
250 DIM SI(9),CM(9)
260 DRAW "BM0,2;D5U3R7U2D6"
270 GET(0,0)-(7,7),WA,G
280 GET(20,20)-(35,35),BL,G
290 GET(20,20)-(27,27),BW,G
300 PUT(0,0)-(7,7),BW
310 RETURN
```

```
320 COLOR 1,0:DIM CA(7)
330 DRAW"BM4,0;G2D10F2R4E3U8H3L4;BM3,2;D
3R8U3L8;BM3,10;D3R8U3L8":PAINT(7,7),1,1:
PAINT(7,1),1,1:PAINT(7,14),1,1
340 GET(0,0)-(15,15),CA,G
350 RETURN
360 A$=INKEY$
370 IF PPOINT(WP-4,88)=1 OR PPOINT(WP-4,
102)=1 OR PPOINT(WP+4,92)=1 OR PPOINT(WP
+4,102)=1 THEN PP=1:Q=Q+1:IF Q=10 THEN Q
=9
380 PUT(WP,92)-(WP+7,99),BW
390 IF A$="L" AND FL=1 OR A$="A" AND FL=
-1 THEN WP=WP+FL*16
400 PUT(WP, 92)-(WP+7, 99),WA
410 IF PP=1 THEN GOSUB 580
420 PP=0
430 RETURN
440 FOR K=1 TO Q
450 IF HE(K)=0 THEN GOSUB 520
460 PUT(HE(K),SI(K))-(HE(K)+15,SI(K)+15)
,BL,OR
470 SI(K)=SI(K)-CM(K)*20
480 IFSI(K)<26 AND CM(K)=1 OR SI(K)>170
AND CM(K)=-1 THEN HE(K)=0:PLAY"T20003AAD
CBCDFG":GOTO 500
490 PUT(HE(K),SI(K))-(HE(K)+15,SI(K)+15)
,CA,PSET
500 NEXT
510 RETURN
520 SI(K)=RND(2)-1
530 CM(K)=2*(SI(K)-.5):REM CAR MOVEMENT
540 SI(K)=160*SI(K)+20
```

```
550 HE(K)=ABS(WP-75+RND(150))
560 IF HE(K)>240 THEN HE(K)=480-HE(K)
570 RETURN
580 K=RND(5):HE(K)=0:J=2:XX=XX+1:PLAY "T
100V3003GGADFGEGG"
590 IF XX=9 THEN 150
600 FORD=1TO20
610 PLAY"O5BO1D"
620 IF D=5 THEN PLAY"T100DFGED"
630 NEXT:FOR D=1 TO 300:NEXT
640 PLAY"03C"
650 PCLS0
660 RETURN
670 CLS
680 PRINT@ 128, TAB(10) "crazy highway"
690 SCREEN 0,1
700 FOR DD=1 TO 1000:NEXT DD
710 FOR TT=1 TO 5
720 PLAY "T25004EEEDDCGFABBEDF"
730 NEXT TT
740 PRINT@ 192, " YOU MUST CROSS THE ROA
D TWENTY"
750 PRINT@ 256, TAB(14) "TIMES"
760 SCREEN 0,1
770 FOR DD=1 TO 500:NEXT DD
780 PRINT@ 352, TAB(8) "goodluck pussyca
t!!"
790 SCREEN 0,1
800 FOR DD=1 TO 1000:NEXT DD
810 PLAY "T50V3002L100AAAFFFGGGAAAFFFGGG
GG"
820 RETURN
```

```
1 REM JOYSTICK VERSION USE RIGHT JOYSTIC
K
10 REM CRAZY HIGHWAY
20 PMODE 3,1:CLS:PCLS0
30 GOSUB 670
40 GOSUB 240:GOSUB 320:REM STORE CHARACT
ERS
50 PUT (0,0)-(15,15),BL,PSET
60 SCREEN 1,0:WP=8:FOR I=1 TO 10:FL=1
70 GOSUB 360:REM MOVE CAT
80 GOSUB 440:REM MOVE VEHICLE
90 IF WP*FL>240 OR (WP<9 AND FL=-1) THEN
 100 ELSE 70
100 Z=Z+1
110 IF FL=1 THEN FL=-1:GOTO 70
120 IF I>3 THEN Q=INT(I/2)+1
130 NEXT I
140 CLS:GOTO 160
150 CLS:PRINT:PRINT"YOU HAVE LOST YOUR N
INE LIVES!!":PRINT:PRINT
160 PRINT:PRINT:PRINT:PRINT:PRINT"  YOU
HAVE BEEN HIT";XX;"TIMES!!":PRINT:PRINT"
  YOU HAVE CROSSED THE ROAD";Z:PRINT TAB
(12) "TIMES!!"
170 PRINT:PRINT:PRINT "DO YOU WISH TO PL
AY AGAIN?(Y/N)"
180 A$=INKEY$
190 IF A$="" THEN 180
200 IF A$="Y" THEN RUN
210 IF A$="N" THEN CLS:PRINT@ 235, "GOOD
BYE":GOTO 230
220 GOTO 180
230 GOTO 230
```

```
240 COLOR 2,0:DIM WA(2):DIM BL(7):DIM BW
(2):DIM HE(9):Q=1
250 DIM SI(9),CM(9)
260 DRAW "BM0,2;D5U3R7U2D6"
270 GET(0,0)-(7,7),WA,G
280 GET(20,20)-(35,35),BL,G
290 GET(20,20)-(27,27),BW,G
300 PUT(0,0)-(7,7),BW
310 RETURN
320 COLOR 1,0:DIM CA(7)
330 DRAW"BM4,0;G2D10F2R4E3U8H3L4;BM3,2;D
3R8U3L8;BM3,10;D3R8U3L8":PAINT(7,7),1,1:
PAINT(7,1),1,1:PAINT(7,14),1,1
340 GET(0,0)-(15,15),CA,G
350 RETURN
360 A=63-JOYSTK(0)
370 IF PPOINT(WP-4,88)=1 OR PPOINT(WP-4,
102)=1 OR PPOINT(WP+4,92)=1 OR PPOINT(WP
+4,102)=1 THEN PP=1:Q=Q+1:IF Q=10 THEN Q
=9
380 PUT(WP,92)-(WP+7,99),BW
390 IF A<20*FL OR (A>43 AND FL=-1) THEN
WP=WP+FL*16
400 PUT(WP, 92)-(WP+7, 99),WA
410 IF PP=1 THEN GOSUB 580
420 PP=0
430 RETURN
440 FOR K=1 TO Q
450 IF HE(K)=0 THEN GOSUB 520
460 PUT(HE(K),SI(K))-(HE(K)+15,SI(K)+15)
,BL,OR
470 SI(K)=SI(K)-CM(K)*20
```

```
480 IFSI(K)<26 AND CM(K)=1 OR SI(K)>170
AND CM(K)=-1 THEN HE(K)=0:PLAY"T20003AAD
CBCDFG":GOTO 500
490 PUT(HE(K),SI(K))-(HE(K)+15,SI(K)+15)
,CA,PSET
500 NEXT
510 RETURN
520 SI(K)=RND(2)-1
530 CM(K)=2*(SI(K)-.5):REM CAR MOVEMENT
540 SI(K)=160*SI(K)+20
550 HE(K)=ABS(WP-75+RND(150))
560 IF HE(K)>240 THEN HE(K)=480-HE(K)
570 RETURN
580 K=RND(5):HE(K)=0:J=2:XX=XX+1:PLAY "T
100V3003GGADFGEGG"
590 IF XX=9 THEN 150
600 FORD=1TO20
610 PLAY"O5BO1D"
620 IF D=5 THEN PLAY"T100DFGED"
630 NEXT:FOR D=1 TO 300:NEXT
640 PLAY"03C"
650 PCLS0
660 RETURN
670 CLS
680 PRINT@ 128, TAB(10) "crazy highway"
690 SCREEN 0,1
700 FOR DD=1 TO 1000:NEXT DD
710 FOR TT=1 TO 5
720 PLAY "T25004EEEDDCGFABBEDF"
730 NEXT TT
740 PRINT@ 192, " YOU MUST CROSS THE ROA
D TWENTY"
750 PRINT@ 256, TAB(14) "TIMES"
```

```
760 SCREEN 0,1
770 FOR DD=1 TO 500:NEXT DD
780 PRINT@ 352, TAB(8) "goodluck pussyca
t□□"
790 SCREEN 0,1
800 FOR DD=1 TO 1000:NEXT DD
810 PLAY "T50V30O2L100AAAFFFGGGAAAFFFGGG
GG"
820 RETURN
```

# Pro Golf

This program makes very good use of the Dragon's high resolution graphics facilities to draw a fairway complete with trees, a water hazard and several bunkers.

The fairway is a dogleg to the right, and you can either play it safe and hit down the fairway, or, if you are confident of your golfing skills, you can take the direct line over the trees and along the water hazard.

The computer will tell you which way the wind is blowing and how strong it is. North is to the top of the screen, south is at the bottom, east is to the right and west is to the left.

After you have studied the fairway and the layout of the various obstacles it is time to tee off. To change from the graphics screen to the text screen, just press any key. You will then be asked in which direction you wish to hit the ball. This direction is entered as a compass bearing anywhere from zero to 360 degrees. Just to make the game a little bit harder, we have put zero degrees to the left of the screen. This makes 90 degrees straight down, 180 degrees to the left, and straight up the screen is 270 degrees.

After you have selected the direction, the computer will ask you which club you are going to use. Club number one is your driver and club number ten would be used for a short-approach shot.

Next you will be asked how hard you want to hit the ball: hard, medium or soft. After you have answered this question, the ball will speed off towards the tee – provided you haven't hooked, sliced or mis-hit the ball, of course!

When you have reached the green the computer will tell you which way the green slopes and will then give you the chance to putt out the hole. To change from the graphics screen to the text screen, once again press any key.

You will be asked which direction you want to hit the ball. This is entered as a compass direction in exactly the same way as your tee shot. You select how hard you wish to hit the ball and then sit back and watch the result. Putting is very difficult on the computer course and it will take a lot of practice before you become an expert.

```
10 REM PRO GOLF
20 PCLEAR4
30 GOSUB 2310
40 GOSUB 160
50 GOSUB 230
60 GOSUB 350
70 FOR J=1 TO I:X(J)=60:Y(J)=170:NEXT
80 FOR J=1 TO I:IF G(J)=1 THEN IFJ<I THE
N 120 ELSE 80
90 GOSUB 560:CLS:GOSUB 620
100 GOSUB 860:IF G=I THEN 130
110 PSET(X(J),Y(J),PS(J))
120 NEXT:IF G<I THEN 80
130 FOR J=1 TO I:GOSUB 1440:NEXT
140 GOTO 2010
150 GOTO 150
160 DIM TR(10)
170 CLS:PRINT@264,"A HOLE OF GOLF"
180 PRINT:PRINT"HOW MANY PEOPLE ARE PLAY
ING";:INPUT I
190 DIM X(I),Y(I),SH(I),K(I),J(I),G(I),B
U(I)
200 SCREEN 0,1
210 FOR K=1 TO 6:PLAY"O1T100GGEDAAAFF":N
EXT
```

```
220 RETURN
230 PMODE3,1
240 PCLS
250 A$="S6BM40,40; U3D3E2D2E2G2R2G2R3L3F
2L2F2H2D2H2D3U3G2U2G2E2L2E2L3R3H2R2H2F2U
2F2"
260 DRAW A$
270 PAINT(40,42),4,4
280 LINE(40,46)-(40,55),PSET:LINE(36,55)
-(45,55),PSET
290 GET(30,36)-(49,55),TR,G
300 PCLS
310 RETURN
320 GOSUB 2080:RETURN
330 X$=INKEY$:IFX$=""THEN 330
340 RETURN
350 FOR K=1 TO 30
360 SCREEN 1,0
370 READ X,Y
380 PUT(X,Y)-(X+19,Y+19),TR,OR
390 NEXT
400 DATA 0,0,10,10,0,10,20,5,0,30,5,60,4
,150,20,170,150,175,232,0,210,5,220,20,2
00,40,210,32,230,50,220,80,95,60,100,75,
110,80,100,90,20,100,80,140,110,0,120,55
,140,100
410 DATA 180,100,190,130,200,170,150,110
,85,170
420 COLOR 3,3
430 DRAW"BM180,60;R4F2R4F2R4F2D4F3D2G2D2
L3D3G5L3H2L4H3U4L4G3U2G2U3L3G4L3G3U4E2U3
R4E4R2G3E4L2H4R5E4U3E3F2D2"
440 PAINT(185,80),3,3
```

```
450 DRAW"BM0,145;R4E3R5E2R3E4R3F2R4F3R4E
2R3E2F3R4E4U3E4R4F4D2F3R4F4R3E4R4E3U4E4U
4H4U3E3R4E4"
460 DRAW"BM0,149;R5E2R5E2R3E4R3F2R4F3R4E
2R3E2F3R6E4U3E4R1F3D2F4R4F4R6E4R4E5U4E4U
5H4U2E3R2E2"
470 PAINT(135,130),3,3
480 COLOR 2,1
490 DRAW"BM40,20;R4F4D2G3L4G2R1G3D2F3D2G
2L3H3U4H2U3E4U2E4":PAINT(40,25),2,2
500 DRAW"BM185,0;R5F1R3F2R2F1D2F2G2L2G1L
1H1U1H2L3U1H1L2H2L2L2L3G4L2G2L2H3U2E3R2H
1E4R13":PAINT(185,4),2,2
510 COLOR 2,0
520 LINE(181,34)-(181,24),PRESET:LINE(18
1,24)-(184,26),PRESET
530 LINE(76,170)-(79,173),PRESET,BF:LINE
(42,170)-(45,173),PRESET,BF
540 SCREEN 1,0
550 RETURN
560 PSET(XO(J),YO(J),PS(J))
570 PS(J)=PPOINT(X(J),Y(J))
580 PSET(X(J),Y(J),2)
590 XO(J)=X(J):YO(J)=Y(J)
600 PLAY"O3T100CDA"
610 RETURN
620 PRINT@32,"   THE WIND IS BLOWING FRO
M THE ";
630 PRINT@10,"PLAYER NO.";J
640 IF FL>0 THEN FH=1:AW=FL:GOTO 670
650 AW=RND(8)
660 FL=AW
```

```
670 ON AW GOTO 680,690,700,710,720,730,7
40,750
680 AW$="NORTH EAST":GOTO 760
690 AW$="EAST":GOTO 760
700 AW$="SOUTH EAST":GOTO 760
710 AW$="SOUTH":GOTO 760
720 AW$="SOUTH WEST":GOTO 760
730 AW$="WEST":GOTO 760
740 AW$="NORTH WEST":GOTO 760
750 AW$="NORTH":GOTO 760
760 AW=(3.14147/4)*(AW-6)
770 PRINT@67,AW$;
780 IF FH=1 THEN WS=WW:GOTO 800
790 WS=RND(3):WW=WS
800 ON WS GOTO 830,820,810
810 WS$="STRONG":GOTO 840
820 WS$="MEDIUM":GOTO 840
830 WS$="LIGHT"
840 PRINT" AND IS ";WS$:WS=7*WS:FH=0
850 RETURN
860 GOSUB 2070:GOSUB 2100:PRINT@128,"WHA
T DIRECTION DO YOU WISH TO  HIT THE BAL
L IN (0-360DEGREES)";:INPUT AS:AS=AS*3.1
4148/180
870 SH(J)=SH(J)+1:GOSUB 2100
880 PRINT@224,"WHICH CLUB DO YOU WISH TO
 USE?  (1-10)";:INPUTCC:CC=INT(CC):IFCC<
1 OR CC>10 THEN 880
890 PRINT@290,"DO YOU WISH TO HIT THE BA
LL     HARD(H),MEDIUM(M) OR SOFT(S)?"
900 INPUT H$
910 IF H$="H" THEN S=1.1:GOTO 950
920 IF H$="M" THEN S=.8:GOTO 950
```

```
930 IF H$="S" THEN S=.6:GOTO 950
940 GOTO 900
950 IFRND(3)=1 THEN HO=RND(21)-11 ELSE H
O=0
960 IF RND(15)=1 THEN HO=(RND(3)-2)*30
970 IF RND(20)=1 THEN S=.2:AS=AS+(RND(3)
-2)*40*3.14148/180:H$="DUFFED":GOTO 1000
980 AS=AS+HO*3.14148/180:IFHO=0 THEN 101
0
990 IF HO<0 THEN H$="HOOKED" ELSEH$="SLI
CED"
1000 PRINT@ 416,"YOU HAVE ";H$;" THE BAL
L":GOSUB 2080
1010 D=S*(110-10*CC)+D*RND(20)/100
1020 XD=D*COS(AS):YD=D*SIN(AS)
1030 WA=WS*COS(AW):WB=WS*SIN(AW)
1040 R=.02*D*(11-CC)
1050 IF BU(J)=1 THEN GOSUB 2290
1060 IF XD=0 THEN 1270
1070 OO=0:FOR LC=1 TO 20
1080 IF LC>1 THEN GOSUB 2110
1090 PX=INT(LC*XD/20)
1100 PY=INT(LC*YD/20)
1110 IF D<30 THEN D=30
1120 WX=INT((D-30)*LC*WA/2000)
1130 WY=INT((D-30)*LC*WB/2000)
1140 XX=LC:H=(3*CC+20)*.01*S*XX*(20-XX)
1150 IFRND(30)=1 THEN WG=WG+RND(2)
1160 GX=INT(LC*WG*COS(AW)/50)
1170 GY=INT(LC*WG*SIN(AW)/50)
1180 TX=PX+WX+GX
1190 TY=PY+WY+GY
```

```
1200 X(J)=X(J)+TX-OX:IFX(J)>255 OR X(J)<
1 THEN GOSUB 2050
1210 Y(J)=Y(J)+TY-OY:IFY(J)>191 OR Y(J)<
1 THEN GOSUB 2050
1220 SCREEN 1,0:GOSUB 560
1230 OX=TX:OY=TY
1240 IF LC=20 THEN OX=0:OY=0
1250 NEXT:GOSUB 1290
1260 IF X(J)>161 AND X(J)<201 AND Y(J)>1
4 AND Y(J)<54 THEN GOSUB 1410
1270 GOSUB 2090:RETURN
1280 A$=INKEY$:IF A$=""THEN 1280 ELSE RE
TURN
1290 IF OO=1 THEN RETURN
1300 FOR RC=1 TO 10
1310 GOSUB 2180
1320 RX=INT(RC*R*COS(AS)/10)
1330 RY=INT(RC*R*SIN(AS)/10)
1340 X(J)=X(J)+RX-AX:Y(J)=Y(J)+RY-AY
1350 IF X(J)>255 OR X(J)<1 THEN GOSUB 20
50
1360 IF Y(J)>191 OR Y(J)<1 THEN GOSUB 20
50
1370 GOSUB 560:AX=RX:AY=RY
1380 IF RC=10 THEN AX=0:AY=0
1390 PLAY"O3T90GFDCEA"
1400 NEXT:RETURN
1410 G=G+1:G(J)=1
1420 PLAY"O3T50GFDBCDA"
1430 J(J)=256*(X(J)-161)/40:K(J)=192*(Y(
J)-14)/40:RETURN
1440 PMODE 1,2:PCLS
1450 PSET(J(J),K(J),2):AO=J(J):BO=K(J)
```

```
1460 LINE(128,96)-(128,56),PSET
1470 LINE-(142,60),PSET:LINE-(128,64),PS
ET:PAINT(136,60),4,4
1480 GOSUB 1520
1490 GG=0:GOSUB 1750:IF GG=1 THEN RETURN
1500 GOTO 1480
1510 RETURN
1520 CLS:PRINT@35,"THE GREEN SLOPES IN A
     ";
1530 PRINT@10,"PLAYER NO.";J
1540 IF FG>0 THEN FI=1:AL=FG:GOTO 1560
1550 AL=RND(8):FG=AL
1560 ON AL GOTO 1570,1580,1590,1600,1610
,1620,1630,1640
1570 AS$="NORTH EASTERLY":GOTO 1650
1580 AS$="EASTERLY":GOTO 1650
1590 AS$="SOUTH EASTERLY":GOTO 1650
1600 AS$="SOUTHERLY":GOTO 1650
1610 AS$="SOUTH WESTERLY":GOTO 1650
1620 AS$="WESTERLY":GOTO 1650
1630 AS$="NORTH WESTERLY":GOTO 1650
1640 AS$="NORTHERLY"
1650 AL=(3.14147/4)*(AL-2)
1660 PRINT@67,AS$;" DIRECTION"
1670 IF FI=1 THEN SS=ST:GOTO 1690
1680 SS=RND(3):ST=SS
1690 ON SS GOTO 1700,1710,1720
1700 SS$="SLIGHT":GOTO 1730
1710 SS$="MEDIUM":GOTO 1730
1720 SS$="STEEP"
1730 PRINT@99,"GREEN SLOPE IS ";SS$:SS=2
0*SS:FI=0
1740 RETURN
```

```
1750 GOSUB 2080:GOSUB 2100:PRINT@128,"WH
AT DIRECTION DO YOU WISH TO  HIT THE BA
LL (0-360 DEGREES)",:INPUT AP:AP=AP*3.14
147/180
1760 SH(J)=SH(J)+1:GOSUB 2100
1770 PRINT@224,"HOW HARD DO YOU WISH TO
HIT THE BALL (1-20)";:INPUT HP:HP=7*(HP+
INT(RND(HP/2)))
1780 XD=HP*COS(AP):YD=HP*SIN(AP)
1790 SA=HP*SS*COS(AL)/100:SB=HP*SS*SIN(A
L)/100
1800 FOR LC=1 TO 20
1810 DX=INT(LC*XD/20):DY=INT(LC*YD/20)
1820 SX=INT(LC*SA/20):SY=INT(LC*SB/20)
1830 TJ=DX+SX:TK=DY+SY
1840 J(J)=J(J)+TJ-OA:K(J)=K(J)+TK-OB
1850 IF J(J)>255 THEN J(J)=250
1860 IF J(J)<1 THEN J(J)=1
1870 IF K(J)>191 THEN K(J)=191
1880 IF K(J)<1 THEN K(J)=1
1890 GOSUB 1920
1900 OA=TJ:OB=TK:IF LC=20 THEN OB=0:OA=0
1910 NEXT:RETURN
1920 SCREEN 1,0:PSET(AO,BO,1)
1930 PSET(J(J),K(J),2)
1940 AO=J(J):BO=K(J)
1950 PLAY"O2T40FGF"
1960 IF J(J)>126 AND J(J)<130 AND K(J)>9
4 AND K(J)<98 THEN GOSUB 1980
1970 RETURN
1980 FOR K=1 TO 20 STEP 2
1930 PLAY"O3T100AACDCGFEE":SOUND12*K,20/
K
```

```
2000 NEXT:LC=20:GG=1:RETURN
2010 CLS:FOR J=1 TO I:PRINT"PLAYER NO.";
J;" TOOK";SH(J);"STROKES":NEXT
2020 PRINT:PRINT"DO YOU WISH TO PLAY AGA
IN?"
2030 A$=INKEY$:IF A$=""THEN 2030ELSE IF
A$="Y" THEN RUN
2040 END
2050 CLS:PRINT@192,"YOU HAVE HIT THE BAL
L OUT OF    BOUNDS!!!!  START THE HOLE A
GAIN AND LOSE TWO STOKES":GOSUB 2070
2060 RC=10:OO=1:SH(J)=2:LC=20:X(J)=60:Y(
J)=170:RETURN
2070 FOR K=1 TO 500:NEXT
2080 FOR K=1 TO 500:NEXT
2090 FOR K=1 TO 500:NEXT:RETURN
2100 SCREEN 1,0:GOSUB 1280:SCREEN 0,1:RE
TURN
2110 IF PS(J)=4 AND H<16 THEN 2120 ELSE
RETURN
2120 CLS:PRINT@238,"CLUNK":PLAY"O4T50ADC
GDEEDCAEC"
2130 AS=RND(360*3.14148/180)
2140 XD=D*(1-LC/20)*COS(AS)/2
2150 YD=D*(1-LC/20)*SIN(AS)/2
2160 OX=0:OY=0
2170 GOSUB 2090:RETURN
2180 IF PS(J)=3 THEN 2210
2190 IF PS(J)=2 THEN 2270
2200 RETURN
2210 CLS:PRINT@236,"SPLOOSH":PLAY"O1T20A
O2T40BCO3T80DEFO4G":GOSUB 2090
2220 RC=10:AM=AS+3.14148
```

```
2230 X(J)=X(J)+COS(AM)
2240 Y(J)=Y(J)+SIN(AM)
2250 IF PPOINT(X(J),Y(J))<>3 THEN 2260 E
LSE 2230
2260 SH(J)=SH(J)+1:GOSUB 2080::PRINT:PRI
NT"YOU MUST TAKE A DROP AND LOSE A STROK
E":GOSUB 2080:GOSUB 560:RETURN
2270 CLS:PRINT@237,"PLOP":PLAY"O2T5FBO1T
2FCA":GOSUB 2090
2280 RC=10:BU(J)=1:RETURN
2290 BU(J)=0:IFRND(3)=1 THENXD=.3*XD:YD=
.3*YD:RETURN
2300 CLS:PRINT@225,"THE BALL HAS ROLLED
BACK INTO   THE BUNKER":GOSUB 2080:XD=0:
RETURN
2310 CLS 3
2320 FOR ZZ=1 TO 50
2330 POKE 1023+RND(512),112
2340 SOUND RND(75)+50,1
2350 NEXT ZZ
2360 FOR ZZ=1 TO 10
2370 PRINT@ 256, TAB(12) "pro golf"
2380 PLAY "O4T4L50AAAACCCCEEEE"
2390 PRINT@ 256, ""
2400 NEXT ZZ
2410 RETURN
```

# Escape

Escape is a fast-moving graphics game designed to test your coordination.

Lines 50 to 100 print a series of characters across the top of the screen. The idea is to use a movable bat to bounce a ball up into these characters until you have knocked enough of them out of the way. This allows the ball to escape to the top of the screen. If you miss three balls with the bat, the game ends and you are given a score.

The movement of the bat is controlled by the "Z" and the "/" keys. The "Z" key moves the bat to the left and the "/" key moves the bat to the right.

Line 40 sets the variable SC to zero. SC holds your score for each game and has to be reset each time the game is played. Variable CX in line 110 sets the number of balls available in the game. You can increase or decrease the difficulty of the game by altering CX.

Line 120 uses BA (the variable which remembers the position of the ball across the screen) and a randomly generated number to give a different starting position for the ball each time it is served. Line 130 sets the starting position of the ball down the screen. Variable F in line 140 is the distance across the screen of the player's bat.

The variables X and Y in line 150 are used to alter the position of the ball on the screen. They are added to the screen location in lines 260 and 270. To change the direction in which the ball is moving, the values of X and Y are converted to negative numbers and back again as required. Adding a negative number to another number actually subtracts it from that number, so by adding minus one to BD in line 270, the direction of the ball is changed and it goes back up the screen.

Variables EA and ED are used to hold the previous screen location of the ball and, just before the ball is about to be

POKEd into its new location by line 180, line 170 POKEs a blank space into the old location.

Line 200 prevents the ball from going off either side of the screen. The subroutine which is called up changes the direction of the ball by altering the value of X (line 370).

Line 120 checks to see if the ball is 12 lines down from the top of the screen. If it is, the program then goes to the routine in lines 390 to 420. This checks to see if the ball has hit the bat. If contact has been made, the game goes on, with the ball going back up the screen. If the ball has missed the bat, we then go to line 430 where one is subtracted from CX, the score is printed and new starting positions are chosen for the bat and the next ball. If CX equals zero then the game is over.

Lines 220, 230 and 250 use PEEK to check the line above the ball to see if the space which the ball is about to move into is occupied by an X. The routines which are then called up POKE a space to blank out the X which has been hit, increment the score and then change the direction of the ball by altering the value of Y.

In lines 260 and 270 the location of the ball is changed by adding the current values of X and Y. Your input from the keyboard is accepted in lines 280 to 300. The value of F is altered and lines 310 and 320 make sure your bat stops when it reaches the edge of the screen.

Lines 330 and 340 are used to POKE the bat on to the screen. The extra POKE statements in line 340 are used to POKE spaces either side of your bat so that only one bat appears on the screen at any one time.

Line 350 sends the program back to line 170 so that it can all happen again.

```
10 REM ESCAPE
20 GOSUB 1000
30 CLS
```

```
40 SC=0
50 PRINT "##############################
##";
60 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXX
X"
70 PRINT " X X X X X X X X X X X X X X
 "
80 PRINT "X . X . X . X . X . X . X . X
."
90 PRINT " X . X . X . X . X . X . X . X
 "
100 PRINT "X X X X X X X X X X X X X X
 X"
110 CX=3
120 BA=RND(5)+6
130 BD=7
140 F=5+RND(10)
150 X=1:Y=1
160 EA=BA:ED=BD
170 POKE 1024+(32*ED+EA),96
180 POKE 1024+(32*BD+BA),79
190 EA=BA:ED=BD
200 IF BA>30 OR BA<1 THEN GOSUB 360
210 IF BD>12 THEN GOTO 390
220 IF PEEK(992+32*BD+BA)=88 THEN GOSUB
530:GOTO 260
230 IF PEEK(992+32*BD+BA+1)=88 THEN GOSU
B 530:GOTO 260
240 IF BD<1 THEN GOTO 250
250 IF PEEK(992+32*BD+BA-1)=88 THEN GOSU
B 530
260 BA=BA+X
270 BD=BD+Y
```

```
280 A$=INKEY$
290 IF A$="Z" THEN F=F-2
300 IF A$="/" THEN F=F+2
310 IF F<1 THEN F=1:POKE 1440+F,96:POKE
1440+F+1,96
320 IF F>30 THEN F=30:POKE 1440+F,96:POK
E 1440+F-1,96
330 S=1440+F
340 POKE S-2,96:POKE S+2,96:POKE S,140
350 GOTO 170
360 GOSUB 980
370 X=-X
380 RETURN
390 GOSUB 980
400 IF ABS(F-BA)>1 THEN 430
410 Y=-Y
420 GOTO 220
430 GOSUB 980
440 CX=CX-1
450 IF CX=0 THEN 610
460 PRINT@ 448, "SCORE:";SC
470 PRINT@ 416,
       ":REM 31 SPACES
480 FOR T=100 TO 200 STEP 10:SOUND T,1:N
EXT T
490 BA=RND(5)+6
500 BD=7
510 F=5+RND(10)
520 GOTO 160
530 GOSUB 980
540 POKE (992+32*BD+BA),96
550 IF RND(10)>8 THEN POKE (992+32*BD+BA
+1),96
```

```
560 IF RND(10)<2 THEN POKE (992+32*BD+BA
-1),96
570 SC=SC+437
580 PRINT@ 448, "SCORE:";SC
590 Y=-Y
600 RETURN
610 FOR DD=1 TO 1000:NEXT DD
620 FOR TT=1 TO 5
630 PLAY "T250V3004AAAGGGCCCFFF"
640 FOR DD=1 TO 10:NEXT DD
650 NEXT TT
660 CLS
670 PMODE 1,1
680 PRINT@ 160, TAB(8) "THE GAME IS OVER
"
690 SCREEN 0,1
700 FOR DD=1 TO 500:NEXT DD
710 PRINT@ 224, TAB(6) "YOUR SCORE WAS";
SC
720 SCREEN 0,1
730 FOR DD=1 TO 1500:NEXT DD
740 GOTO 30
750 FOR DD=1 TO 1000:NEXT DD
760 CLS 0
770 FOR SS=10 TO 200 STEP 20:SOUND SS,1:
NEXT SS
780 GOSUB 940
790 GOSUB 920
800 FOR TT=1 TO 5
810 CLS 3
820 GOSUB 940
830 PLAY "T250V3002AAAGGGFFF"
840 GOSUB 920
```

```
850 CLS 4
860 GOSUB 940
870 PLAY "T250V30O2AAAGGGFFF"
880 GOSUB 920
890 NEXT TT
900 GOSUB 920
910 GOTO 30
920 FOR DD=1 TO 200:NEXT DD
930 RETURN
940 PRINT@ 160, TAB(8) "YOU HAVE DONE IT
"
950 GOSUB 920
960 PRINT@ 224, TAB(6) "YOUR SCORE WAS";
SC
970 RETURN
980 SOUND 5,1
990 RETURN
1000 CLS 0
1010 FOR TT=1 TO 250
1020 XX=RND(7)*16+143
1030 YY=RND(511)+1024
1040 IF PEEK(YY)<>128 THEN GOTO 1030
1050 POKE YY,XX
1060 NEXT TT
1070 PLAY "T250V30L2O3AAAAGGGGFFFFAAAA"
1080 FOR GG=1 TO 5
1090 PRINT@ 192, TAB(12) "ESCAPE"
1100 FOR DD=1 TO 100:NEXT DD
1110 PLAY "T250V30L2O2FFFFAAAAFFFFGGGG"
1120 PRINT@ 192, "
          ":REM 31 SPACES
1130 FOR DD=1 TO 100:NEXT DD
1140 NEXT GG
1150 RETURN
```

# Dog Catcher

Here we have a clever little program which will cause a great deal of excitement. You are a little dog (shown on the screen as a reversed "plus" sign) and you are trying to get away from the dog catcher (shown as a blue square). The part of the dog catcher is, of course, played by the computer.

By moving your dog around the screen – using the "A" key to move up the screen, the "Z" key to move down screen, and the "," and "." keys to move left and right – you must avoid the electronic, fully computerized dog catcher for as long as possible. After the dog catcher has caught you (and you will be surprised just how quickly he can do it) you will be told how long you avoided capture.

Line 50 sets the starting position of your dog, and line 60 gives the initial location of the dog catcher. The variables EA and ED are used to blank out the dog as it moves, and EH and EV do the same for the dog catcher. These variables are used in lines 100 and 110.

The routine from line 120 to line 180 accepts your input, and line 190 POKEs the dog on to the screen.

Lines 200 to 260 control the movements of the dog catcher, and line 270 POKEs him on to the screen. Line 290 checks to see if the dog catcher has caught up with you. If you are still free, line 310 loops the program back to line 70.

```
10 CLS
20 REM DOG CATCHER
30 Z=1024
40 GOSUB 420
50 BA=20:BD=5
60 TH=31:TV=15
```

```
70 EA=BA:ED=BD
80 EH=TH:EV=TV
90 CLS 2
100 POKE Z+(32*ED+EA),159
110 POKE Z+(32*TV+TH),159
120 A$=INKEY$
130 IF A$<>"A" AND A$<>"Z" AND A$<>"," A
ND A$<>"." THEN A$=B$
140 B$=A$
150 IF A$="A" AND BD>0 THEN BD=BD-1
160 IF A$="Z" AND BD<15 THEN BD=BD+1
170 IF A$= "." AND BA<31 THEN BA= BA+1
180 IF A$= "," AND BA>0 THEN BA= BA-1
190 POKE Z+(32*BD+BA),43
200 IF RND(10)>3 THEN 230
210 IF BD>TV THEN TV=TV+1
220 IF RND(10)>3 THEN 250
230 IF BD<TV THEN TV=TV-1
240 IF RND(10)>3 THEN 260
250 IF BA<TH THEN TH=TH-1
260 IF BA>TH THEN TH=TH+1
270 POKE Z+(32*TV+TH),175
280 T=T+1
290 IF BD=TV AND BA=TH THEN 320
300 SOUND 20,1
310 GOTO 70
320 FOR DD=1 TO 500:NEXT DD
330 FOR A=1 TO 12
340 CLS RND(9)-1
350 SOUND RND(20)+10,1
360 FOR DD=1 TO 25:NEXT DD
370 NEXT A
380 PRINT@ 128, TAB(11) "I GOT YOU!"
```

```
390 FOR DD=1 TO 500:NEXT DD
400 PRINT@ 288, TAB(7) "YOUR SCORE WAS";
T
410 GOTO 410
420 CLS 2
430 POKE Z+128+31,43
440 POKE Z+128+X,175
450 X=X+1
460 IF X=31 THEN 500
470 FOR DD=1 TO 25:NEXT DD
480 POKE Z+127+X,159
490 GOTO440
500 SOUND 25,1
510 FOR A=1 TO 4
520 CLS 0
530 FOR DD=1 TO 150:NEXT DD
540 FOR A=1 TO 4
550 PRINT@ 128, TAB(8) "** DOG CATCHER *
*"
560 SOUND 100,A+2
570 NEXT A
580 FOR DD=1 TO 1000:NEXT DD
590 CLS 2
600 RETURN
```

# Maze Runner

Maze Runner is an arcade-type action game which will require quick reflexes and a good deal of luck.

The computer will draw a blue border around the screen and place blue blocks in the playing area. Down in the right-hand bottom corner there is an exit gate, indicated by a black and white bar.

The computer will place you, the maze runner, in the top left-hand corner of the field. You must run across the screen and out of the exit gate without bumping into the maze blocks or the surrounding wall. You start the game with 2000 points, but each time you hit a block or a wall points are deducted from your score. The idea is to complete the task losing as few points as possible.

To make life a little more interesting, the computer keeps adding maze blocks to the screen as the game runs.

To control the maze runner, a reversed "plus" sign, use the "A", "Z", "," and "." keys. The "A" key moves the runner up the screen and the "Z" key moves you down screen. The "," and "." keys move you left and right respectively.

Line 40 sets your score to 2000. This is reduced by lines 330, 370, 410 and 450 each time you bump into a block or a wall. Line 50 sets B$ equal to the "." key. This is done to start the maze runner moving at the beginning of the game.

Lines 60 to 110 print up the playing field. Line 120 gives the variable SL the value of 1090; this is the place where the maze runner starts on the screen. The part of the computer's memory which controls the screen begins at memory location 1024 and runs to 1534. By using POKE we can place a value into any of these locations, thereby putting something directly on to the screen without using PRINT. This is how we will create the movement in this game.

Lines 140 to 160 choose screen locations at random and

POKE the maze blocks into them. The variable W is then set to equal the value of the screen location.

Lines 180 to 240 then accept the input from the keyboard and move the maze runner around the screen by adding or subtracting numbers from the variable SL. By adding or subtracting 32 (the number of spaces across the screen) from SL, we can make the character move up or down the screen. It moves into the corresponding location in the line above or below the one it is already in. The PEEK statements check to see if the new screen location is already occupied by a blue block (code 175). If it is, the subroutines are called up. Each of these four routines does virtually the same job. First they call up the sound routine to provide the noise of hitting a block at full speed. The next line moves the maze runner back to his previous position (this prevents him from moving on to the block and erasing it). The following line alters your score, then the routine RETURNs to the main program.

Lines 250 and 260 are safeguards to prevent the maze runner from accidentally leaving the screen and running around other memory locations. Many unusual disasters can occur if a stray POKE value is allowed to wander through the computer's ROM. Always put safeguards into your moving graphics programs to prevent this from happening.

Line 270 checks to see if you have arrived at the exit gate. If you have, the end routine is called up.

Line 280 POKEs a blank space into location W, which is the previous position of the runner. Lines 210 to 240 updated SL, so the character is then POKEd into the current value of SL. This means that only one maze runner appears on the screen at a time to give the appearance of continuous movement. The maze runner is POKEd into a location, a space is then POKEd into the same location and the maze runner character is immediately POKEd into a new location.

Line 290 POKEs another maze block on to the screen and then line 300 sends the program back to do it all over again.

This program shows the basic techniques of moving graphics. Study it carefully and adapt it to your own programs.

```
10 REM MAZE RUNNER
20 GOSUB 750
30 CLS
40 SC=2000
50 B$="."
60 FOR XX=0 TO 31:PRINT@ XX, CHR$(175);:
NEXT XX
70 FOR XX=32 TO 415 STEP 32
80 PRINT@ XX, CHR$(175);"
                ";CHR$(175):REM 30 SPACES
90 NEXT XX
100 FOR XX=416 TO 444:PRINT@ XX, CHR$(17
5);:NEXT XX
110 FOR XX=1469 TO 1471:POKE XX,156:NEXT
 XX
120 SL=1030
130 FOR T=1 TO 30
140 Z=1056+RND(401)
150 POKE Z,175
160 NEXT T
170 W=SL
180 A$=INKEY$
190 IF A$<>"A" AND A$<>"Z" AND A$<>"," A
ND A$<>"." THEN A$=B$
200 B$=A$
210 IF A$="." THEN SL=SL+1:IF PEEK(SL)=1
75 THEN GOSUB 310
220 IF A$="," THEN SL=SL-1:IF PEEK(SL)=1
75 THEN GOSUB 350
230 IF A$="Z" THEN SL=SL+32:IF PEEK(SL)=
175 THEN GOSUB 390
240 IF A$="A" THEN SL=SL-32:IF PEEK(SL)=
175 THEN GOSUB 430
```

```
250 IF SL<1024 THEN SL=1024
260 IF SL>1535 THEN SL=1024
270 IF PEEK(SL)=156 THEN 620
280 POKE W,96:POKE SL,43
290 POKE 1024+RND(401),175
300 GOTO 170
310 SOUND 10,1
320 SL=SL-1
330 SC=SC-(RND(50)+50)
340 GOTO 460
350 SOUND 10,1
360 SL=SL+1
370 SC=SC-(RND(50)+50)
380 GOTO 460
390 SOUND 10,1
400 SL=SL-32
410 SC=SC-(RND(50)+50)
420 GOTO 460
430 SOUND 10,1
440 SL=SL+32
450 SC=SC-(RND(50)+50)
460 PRINT@ 448, "SCORE: ";SC
470 IF SC=<0 THEN GOTO 490
480 RETURN
490 FOR DD=1 TO 1000:NEXT DD
500 CLS 0
510 PRINT@ 224, TAB(3) "YOU HAVE RUN OUT
 OF POINTS"
520 PLAY "V30L100AAAAAAAAAA"
530 FOR DD=1 TO 100:NEXT DD
540 CLS 0
550 FOR DD=1 TO 200:NEXT DD
560 PRINT@ 224, TAB(9) "YOU HAVE LOST"
```

```
570 PLAY "V30L100FFFFFFFFFF"
580 FOR DD=1 TO 300:NEXT DD
590 TT=TT+1
600 IF TT=3 THEN TT=0:FOR DD=1 TO 1000:N
EXT DD:GOTO 30
610 GOTO 500
620 FOR DD=1 TO 1000:NEXT DD
630 FOR X=0 TO 8
640 CLS X
650 SOUND RND(50)+125,1
660 FOR DD=1 TO 10:NEXT DD
670 NEXT X
680 PRINT@ 128, TAB(8) "THE GAME IS OVER
"
690 FOR DD=1 TO 500:NEXT DD
700 PRINT@ 224, TAB(6) "YOUR SCORE WAS";
SC
710 FOR DD=1 TO 500:NEXT DD
720 IF HS<SC THEN HS=SC:PRINT@ 320, TAB(
5) "THIS IS THE BEST SCORE"
730 FOR DD=1 TO 1500:NEXT DD
740 GOTO 30
750 CLS 3
760 FOR DD=1 TO 500:NEXT DD
770 FOR NN=1 TO 50
780 POKE1024+RND(468),191
790 SS=RND(25)+200
800 SOUND SS,1
810 NEXT NN
820 PRINT@ 224, TAB(9) "*MAZE RUNNER*"
830 FOR DD=1 TO 1000:NEXT DD
840 FOR TT=1 TO 5
850 PLAY "V30T250L2000O4AAAAGGGGFFFFFFFF"
```

```
860 FOR DD=1 TO 10:NEXT DD
870 NEXT TT
880 RETURN
```

# Paper Chase

You are fortunate to be right on the spot when a large sum of money blows out of the window of a bank. Bundles of notes are lying on the ground and you (the V shape) must rush around picking up the bundles (indicated by the numbers which appear on the screen) before the wind blows them away. Each time you pick up a bundle of money, the amount of money in the bundle is added to the total amount you have. This is displayed at the top of the screen.

You control your man on the screen by using four keys on the keyboard. The "A" key moves you up the screen; the "Z" key moves you down screen; the "," key and the "." key move you left and right respectively.

Paper Chase is only a short program but it will provide you with lots of fun.

Line 40 and line 50 set the score and the starting location of your player. Line 60 sets the variable M, which is used to POKE your character on to the screen. Lines 130 to 160 change the value of M depending on which direction you are travelling. In this way the appearance of the character on the screen remains roughly the same no matter which way you are going.

Line 70 is the start of the FOR/NEXT loop which controls the main part of the game. Line 80 selects a number and a screen location to POKE the number into.

In line 90, variable W is set equal to the screen location (SL). W is used later to place a blank space into the old location of your player. This happens in line 190.

Lines 100 to 160 accept your directions from the keyboard and alter the variable SL accordingly. B$ is used to create continuous movement. In line 120 B$ is set equal to A$, your input. The next time through the loop, if a key isn't pressed then line 110 uses B$ as an input. This is, of course, the direction you last chose.

Line 170 sets the variable Q to equal the PEEK value of SL. Line 180 then tests Q to see if you have 'picked up' a bundle of money. If you have, it works out how much money was in the bundle and then goes to line 230 where your score is printed on the screen. The next line gives the sound effect. Line 200 POKEs your player on to the screen. Line 210 is used to remove the bundle of money from the screen. If the random number generator is less than 100, the money stays on the screen. If it equals 100 then the screen is cleared and the next Z is called up. This also happens after you have picked up a bundle of notes.

```
10 REM PAPER CHASE
20 GOSUB 370
30 CLS
40 SC=0
50 SL=1056
60 M=86
70 FOR Z=1 TO 20
80 POKE 1088+RND(468),RND(9)+112
90 W=SL
100 A$=INKEY$
110 IF A$<>"." AND A$<>"," AND A$<>"A" A
ND A$<>"Z" THEN A$=B$
120 B$=A$
130 IF A$="." THEN SL=SL+1:M=126
140 IF A$="," THEN SL=SL-1:M=124
150 IF A$="A" AND SL>1088 THEN SL=SL-32:
M=65
160 IF A$="Z" AND SL<1503 THEN SL=SL+32:
M=86
170 Q=PEEK(SL)
```

```
180 IF Q>111 AND Q<122 THEN SC=SC+(Q*RND
(10)+5):GOTO 230
190 POKE W,96
200 POKE SL,M
210 IF RND(100)<100 THEN 90
220 FOR T=1 TO 20:NEXT T
230 CLS:PRINT@ 0, "MONEY $";SC;" ";20-Z;
"NOTES LEFT"
240 SOUND RND(25)+175,1
250 NEXT Z
260 FOR DD=1 TO 1000:NEXT DD
270 CLS 3
280 FOR TT=1 TO 5
290 PLAY "V30T250L1001GGGGGGGG02CCCCCC
C"
300 FOR DD=1 TO 10:NEXT DD
310 NEXT TT
320 PRINT@ 160, TAB(4) "ITS ALL OVER MON
EYMAKER"
330 FOR DD=1 TO 500:NEXT DD
340 PRINT@ 288, TAB(6) "YOU MADE";SC;"DO
LLARS"
350 FOR DD=1 TO 2000:NEXT DD
360 GOTO 30
370 CLS 0
380 FOR XX=1024 TO 1535
390 POKE XX,RND(126)+128
400 NEXT XX
410 GG=64
420 FOR TT=1 TO 5
430 PLAY "T200V3002GGAAGGFFGGAA"
440 PRINT@ GG, TAB(10) "PAPER CHASE"
450 FOR DD=1 TO 50:NEXT DD
```

```
460 GG=GG+96
470 NEXT TT
480 FOR DD=1 TO 250:NEXT DD
490 RETURN
```

# Meteor

In this game, written by Mark Farrell, your job is to blast large meteors as they bounce about the sky.

The movement of your laser cannon is controlled by the left and right arrow keys on your keyboard. The "F" key fires the cannon. At the beginning of the game you will be asked which level of difficulty you require. Number one is the hardest, number ten is the easiest. This level of difficulty relates to how long each meteor stays in the sky before it disappears. Each time you let a meteor get away you lose a life.

You will also be asked how many lives you require. You are allowed a minimum of two and a maximum of seven. You lose a life each time you let a meteor get away or if you fire at a meteor and miss. After each hit you will be given a points score. The game ends when all your lives are used up.

Line 120 accepts your required level of difficulty and line 130 makes sure your input is legal. Line 140 multiplies the level of difficulty by three, and this then becomes the length of the loop used to keep the meteor on the screen. Each time the program runs through the loop, from line 190 to line 430, line 250 increments the variable S by one and then tests it to see if it equals the level of difficulty. If it doesn't, we keep on going; but if it does, the program goes to the routine from line 490 which tells us that the time has run out.

Line 170 sets up the starting position for your laser cannon, and line 180 is the starting coordinates for the meteor. Lines 200 and 210 control the random movements of the meteor. Variable A is the vertical position and variable B the horizontal location.

Lines 220 to 240 provide the background of stars in the night sky. Lines 250 to 340 are the routines which draw first your laser cannon and then the meteor.

Your input from the keyboard is taken and processed by lines 350 to 410. Lines 360 and 370 accept the direction changes from the arrow keys and increment variable X accordingly. Lines 390 to 410 then redraw the cannon in its new position. Line 380 checks to see if your input was "F", and if it was we then go to the routine which fires the cannon.

Lines 440 to 470 are the firing routine. The loop in lines 440 to 460 draws the shot from the laser, and the first statement in line 460 checks to see if you have hit the meteor. If you have, we then go to the routine from line 520 which congratulates you and gives you your score. If you haven't hit the meteor then the loop continues. When the loop has finished running the computer knows that you have missed the meteor and so we go to the routine from line 540 which tells you that you have missed.

```
10 REM METEOR
20 CLS 0
30 POKE 65495,0
40 PCLS
50 CLEAR 1000
60 FOR TL=1 TO 5
70 PRINT@ 172,"**********";
80 PLAY "T20002AGFD"
90 FOR DD=1 TO 100:NEXT DD:PRINT@172,"**
METEOR**";
100 PLAY "T20003AFAFDE"
110 FOR DD=1 TO 100:NEXT DD:NEXT TL
120 PRINT@ 295,"";:INPUT"DIFFICULTY (1-1
0)";DF
130 IF DF<1 OR DF>10 THEN GOTO 120
140 DI=DF*3
150 PRINT@ 362,"";:INPUT "LIVES (2-7)";L
```

```
160 IF L<2 OR L>7 THEN 150
170 X=100:Y=193
180 A=100:B=100
190 PMODE 4,1:SCREEN 1,1
200 Q=RND(2):IF Q=1 THEN A=A+RND(25) ELS
E A=A-RND(25):IF A>255 OR A<27 THEN A=RN
D(50)+50
210 W=RND(2):IF W=1 THEN B=B+RND(25) ELS
E B=B-RND(25):IF B>193 OR B<27 THEN B=RN
D(50)+50
220 FOR P=1 TO 5
230 PSET(RND(255),RND(193))
240 NEXT P
250 S=S+1:IF S=DI THEN GOTO 490
260 A$="U5E4F4D5L2U5L4D5L2BU5R8L2H2G2"
270 M$="R3U3L3D3U3E3R3G3D3E3U3D3L3U3D3G3
"
280 MM$=M$+"R3E1R4E2G2L3R3G1"+M$
290 DRAW"S8;A0;BM"+STR$(X)+","+STR$(Y)+A
$
300 DRAW"A3;BM"+STR$(A)+","+STR$(B)+MM$
310 DRAW"A1;BM"+STR$(A)+","+STR$(B-20)+M
M$
320 PAINT(X+2,Y-3),1,1
330 PAINT(X+13,Y-3),1,1
340 PAINT(X+8,Y-13),1,1
350 E$=INKEY$
360 IF E$=CHR$(8) THEN X=X-8
370 IF E$=CHR$(9) THEN X=X+8
380 IF E$="F" THEN GOTO 440
390 LINE(X-8,Y)-(X+28,Y-20),PRESET,BF
400 LINE(A-12,B+10)-(A+12,B-30),PRESET,B
F
```

```
410 PSET(X+8,Y-20)
420 IF PPOINT(X+8,Y-15)=1 THEN GOTO 490
430 GOTO 190
440 FOR T=173 TO 1 STEP -1
450 PSET(X+8,T)
460 IF PPOINT(X+8,T-1)=5 THEN 520 ELSE N
EXT
470 PCLS:GOTO 540
480 FOR T=1 TO 600:NEXT T
490 FOR T=1 TO 10:LINE(RND(255),RND(193)
)-(RND(255),RND(193)),PSET,B:NEXT T
500 CLS 0:PRINT@ 198, "TIME UP - 1 SHIP
LOST!!";
510 PRINT@ 297,"LIVES LEFT =";:L=L-1:PRI
NT L;:PCLS:FOR U=1 TO 1500:NEXT U:S=0:IF
 L=0 THEN 580 ELSE GOTO 170
520 FOR TL=1 TO 3:PLAY "T2500IFGFGO2ABAB
O3DDDP1":NEXT TL
530 CLS 4:PRINT@ 202,"GOOD SHOT!!";:PRIN
T@ 260,"YOU EARN BONUS POINTS!!!";:SC=SC
+RND(200)+500:PRINT@ 330,"SCORE=";SC;:S=
0:FOR V=1 TO 1000:NEXT V:PCLS:GOTO 170
540 FOR TL=1 TO 2:PLAY "T2500IAAFFGGAA":
NEXT TL
550 CLS 0
560 PRINT@ 196, "YOU MISSED - 1 SHIP LOS
T!!";
570 PRINT@ 297, "LIVES LEFT =";:L=L-1:PR
INT L;:PCLS:FOR U=1 TO 1500:NEXT U:S=0.I
F L=0 THEN 580 ELSE GOTO 170
580 CLS 0
590 PLAY "T1000IAABBCCDDCCBBAABBAA"
```

```
600 PRINT@ 192, TAB(8) "THE GAME IS OVER
":FOR DD=1 TO 1000:NEXT DD
610 PRINT@ 256, TAB(10) "SCORE =";SC
620 PRINT@ 320, TAB(8) "PLAY AGAIN (Y/N)
"
630 A$=INKEY$:IF A$="" THEN GOTO 630
640 IF A$="Y" THEN RUN
650 CLS0:PRINT@ 192, TAB(12) "GOODBYE"
660 GOTO 660
```

# Adventure Games

# Vadrand's Treasure

Here we have an adventure-type program, written by Doug Rolfe, which will generate a great deal of suspense and excitement.

You are travelling through an underground fortress built by the warrior king, Vadrand. Vadrand is long dead and the fortress has fallen into ruin but there are still treasures to be found in the many levels and rooms. A magical creature roams through the levels, left behind to protect the fortress from treasure hunters. This monster may only be killed by a sword which you may be fortunate enough to find in the rooms about you. However, finding the sword may not be the answer to your problems. Before you can defeat the creature you must have sufficient experience points to survive the encounter. You gain experience points by finding and examining boxes and chests. In this way you will also find much gold, but watch out for booby traps.

When you get within five rooms of the monster you will probably hear him moving about. If you do not have the sword or the necessary experience, stay away from him.

The program has a teleport feature but you may only use this when you have gained the required magical skills.

After each move you will be told which directions are open to you.

The commands for the game are:

"U" = up
"D" = down
"R" = right
"L" = left
"F" = forwards
"B" = back
"T" = teleport

"G" = get
"P" = put down
"E" = examine
"I" = inventory
"Q" = quit the game

Lines 50 and 60 set up the starting positions for the adventurer and the monster.

Line 70 sets up your stamina and the amount of gold and magic available.

Line 100 calls up the routine which displays the options you have open to you (lines 350 to 430).

Line 110 calls up the routine which moves the monster.

Lines 140 to 330 accept your keyboard input and work out your move (line 170 checks to see if you have been killed).

Line 340 is called into play if you make an illegal move.

Lines 440 to 550 change your position in the fortress.

Lines 560 to 590 provide the inventory routine.

Lines 600 to 610 control the locations of the walls.

Lines 630 to 680 provide the routine which looks after the boxes, the chests and the sword.

Lines 690 to 750 are the GET routine.

Lines 760 to 820 are the PUT DOWN routine.

Lines 820 to 980 are the EXAMINE routine.

Lines 990 to 1100 reset the order of possessions after the GET, PUT DOWN or EXAMINE routines have been used.

Lines 1200 to 1280 move the monster and check his position in relation to your own.

Lines 1290 to 1370 tell you the direction of the monster.

Lines 1380 to 1470 are the monster encounter routine.

```
10 REM VADRAND'S TREASURE
20 GOSUB 1530
30 NP=0
```

```
40 CLS
50 X=RND(100)+20:Y=RND(100)+20:Z=RND(70)
+2
60 MX=RND(20)+X-10:MY=RND(20)+X-Y:MZ=RND
(10)-5+Z
70 T=0.3:S=200+RND(400):M=100+RND(400):G
=500+RND(100)
80 FOR XX=1 TO Z:SOUND 200,1:NEXT XX
90 PRINT:PRINT "YOU HAVE BEEN TRANSPORTE
D TO":PRINT "LEVEL";Z
100 GOSUB 350
110 GOSUB 630
120 GOSUB 1200
130 IF MX=X AND Y=MY AND Z=MZ THEN GOSUB
 1380
140 PRINT "YOUR MOVE:";:S=S-0.1:M=M-0.05
:EX=XE+0.05
150 A$=INKEY$
160 IF A$<>"" THEN PRINT A$
170 IF S<5 THEN GOTO 580
180 IF A$="Q" THEN GOTO 1140
190 IF A$="U" THEN GOTO 540
200 IF A$="E" THEN GOTO 820
210 IF A$="D" THEN GOTO 520
220 IF A$="F" THEN GOTO 440
230 IF A$="I" THEN GOTO 560
240 IF A$="B" THEN GOTO 460
250 IF A$="R" THEN GOTO 500
260 IF A$="G" THEN GOTO 690
270 IF A$="P" THEN GOTO 760
280 IF A$="L" THEN GOTO 480
290 IF A$="T" AND M>600 THEN X=X-10+RND(
21)-1:Y=Y-10+RND(21)-1:GOTO 100
```

```
300 IF A$="T" AND M<=600 THEN GOSUB 1660
:PRINT:PRINT "YOU DO NOT HAVE THE MAGICA
L":PRINT "SKILLS TO TELEPORT YOURSELF YE
T":IF NC=0 THEN PRINT "LOOKS LIKE YOU'RE
 TRAPPED FOR":PRINT "GOOD":GOTO 140
310 IF A$="T" THEN GOTO 140
320 IF A$<>"" THEN PRINT:PRINT "I DON'T
UNDERSTAND THAT...":GOTO 140
330 GOTO 150
340 PRINT "thump!!!":SOUND 1,2:PRINT "(I
 GUESS THAT MEANS YOU CAN'T":PRINT "GO T
HAT WAY)":GOTO 140
350 NC=0:PRINT "YOU CAN GO : ";
360 X=X+1:GOSUB 600:IF W<T THEN NC=1:GOS
UB 1700:PRINT "LEFT, ";
370 X=X-2:GOSUB 600:X=X+1:IF W<T THEN NC
=1:GOSUB 1700:PRINT "RIGHT, ";
380 Y=Y+1:GOSUB 600:IF W<T THEN NC=1:GOS
UB 1700:PRINT "FORWARDS, ";
390 Y=Y-2:GOSUB 600:Y=Y+1:IF W<T THEN NC
=1:GOSUB 1700:PRINT "BACKWARDS, ";
400 Z=Z+1:GOSUB 600:IF W<T THEN NC=1:GOS
UB 1700:PRINT "DOWN, ";
410 Z=Z-2:GOSUB 600:Z=Z+1:IF W<T AND Z>1
 THEN NC=1:GOSUB 1700:PRINT "UP, ";
420 IF NC=0 THEN PRINT "NOWHERE, YOU ARE
 TRAPPED!!":GOSUB 1720:ELSE PRINT "..."
430 RETURN
440 Y=Y+1:GOSUB 600:IF W<T THEN GOTO 100
450 Y=Y-1:GOTO 340
460 Y=Y-1:GOSUB 600:IF W<T THEN GOTO 100
470 Y=Y+1:GOTO 340
480 X=X+1:GOSUB 600:IF W<T THEN GOTO 100
```

```
490 X=X-1:GOTO 340
500 X=X-1:GOSUB 600:IF W<T THEN GOTO 100
510 X=X+1:GOTO 340
520 Z=Z+1:GOSUB 600:IF W<T THEN GOTO 100
530 Z=Z-1:GOTO 340
540 Z=Z-1:GOSUB 600:IF W<T THEN GOTO 100
550 Z=Z+1:GOTO 340
560 PRINT "GOLD PIECES:";G:PRINT "MAGIC
POINTS:";INT(M):PRINT "EXPERIENCE POINTS
:";INT(EX):PRINT "STAMINA:";INT(S)
570 FOR XX=1 TO NP:PRINT PO$(XX):NEXT XX
580 IF S<5 THEN PRINT "YOU HAVE DIED!!":
GOTO 1480
590 GOTO 140
600 U=100*SQR((X^2)+(Y^2)*ABS(Z))
610 W=U-INT(U)
620 RETURN
630 RN=RND(20)-1
640 N=0
650 IF RN=1 OR RN=0 THEN PRINT "YOU HAVE
 FOUND A BOX!!":N=N+1:OB$(N)="BOX"
660 IF RN=2 OR RN=3 THEN PRINT "YOU HAVE
 FOUND A SMALL CHEST!!":N=N+1:OB$(N)="CH
EST"
670 IF RN=19 AND WE=0 THEN PRINT "YOU HA
VE FOUND THE SWORD YOU":PRINT "NEED TO K
ILL THE MONSTER!!":WE=1
680 RETURN
690 IF NP=3 THEN PRINT "YOU CAN'T CARRY
ANY MORE":GOTO 140
700 IF N=0 THEN PRINT "THERE IS NOTHING
TO GET!!":GOTO 140
710 PRINT "THERE IS:":PRINT
```

```
720 FOR XX=1 TO N:PRINT XX;")  ";OB$(XX)
:NEXT XX
730 PRINT:INPUT "WHICH:";GE:IF GE<1 OR G
E>N THEN PRINT "IMPOSSIBLE!!":GOTO 730
740 PRINT "O.K. YOU HAVE A ";OB$(GE):NP=
NP+1:PO$(NP)=OB$(GE):N=N-1
750 GOTO 140
760 IF NP=0 THEN PRINT "YOU HAVE NOTHING
 TO DROP":GOTO 140
770 PRINT "YOU HAVE:":PRINT
780 FOR XX=1 TO NP:PRINT XX;")  ";PO$(XX
):NEXT XX
790 PRINT:INPUT "WHICH:";DR:IF DR<1 OR D
R>NP THEN PRINT "IMPOSSIBLE!!":GOTO 790
800 PRINT "O.K. YOU HAVE DROPPED A";PO$(
GE):OB$(N)=PO$(GE):N=N+1:GOSUB 1070:NP=N
P-1
810 GOTO 140
820 IF NP=0 AND N=0 THEN PRINT "THERE IS
 NOTHING TO EXAMINE":GOTO 140
830 IF NP=0 THEN GOTO 850
840 PRINT "YOU HAVE:":PRINT:FOR XX=1 TO
NP:PRINT XX;")  ";PO$(XX):NEXT XX
850 IF N=0 THEN GOTO 870
860 PRINT:PRINT "YOU SEE:":PRINT:FOR XX=
NP+1 TO NP+N:PRINT XX;")  ";OB$(XX-NP):N
EXT XX
870 PRINT:INPUT "WHICH:";IN:IF IN<0 OR I
N>N+NP THEN PRINT "IMPOSSIBLE!!":GOTO 87
0
880 IF N=0 OR IN=<NP THEN I$=PO$(IN):GOS
UB 990
```

```
890 IF NP=0 OR IN>NP THEN I$=OB$(IN-NP):
GOSUB 1030
900 PRINT "THE ";I$;
910 K=RND(10)-1
920 IF K=0 THEN GOSUB 1660:PRINT " HELD
POISONOUS GAS!!!":S=S-30
930 IF K=1 THEN GOSUB 1660:PRINT " HELD
A SNAKE!!":PRINT "YOU WERE BITTEN!!":S=S
-20:EX=EX+5
940 IF K>1 AND K<5 THEN GOSUB 1620:A=RND
(20)+5:PRINT " CONTAINED";A;"GOLD":PRINT
 " PIECES!!":G=G+A:A$="I":GOTO 230
950 IF K>4 AND K<8 THEN GOSUB 1720:PRINT
 " WAS EMPTY"
960 IF K=8 THEN GOSUB 1660:PRINT " EXPLO
DED.  YOU WERE KILLED!!":S=0:GOTO 580
970 IF K=9 THEN GOSUB 1620:PRINT " CONTA
INED A SPELL THAT": PRINT " YOU":PRINT "
RELEASED.  IT HAS BOOSTED":PRINT "YOUR S
TAMINA AND EXPERIENCE":PRINT "POINTS!!":
S=S+RND(50):EX=EX+RND(100):A$="I":GOTO 2
30
980 EX=EX+5:GOTO 140
990 IF IN=1 THEN PO$(1)=PO$(2):PO$(2)=PO
$(3):PO$(3)="":NP=NP-1
1000 IF IN=2 THEN PO$(2)=PO$(3):PO$(3)="
":NP=NP-1
1010 IF IN=3 THEN PO$(3)="":NP=NP-1
1020 RETURN
1030 IF IN-NP=1 THEN OB$(1)=OB$(2):OB$(2
)=OB$(3):OB$(3)="":N=N-1
1040 IF IN-NP=2 THEN OB$(2)=OB$(3):OB$(3
)="":N=N-1
```

```
1050 IF IN-NP=3 THEN OB$(3)="":N=N-1
1060 RETURN
1070 IF DR=1 THEN PO$(1)=PO$(2):PO$(2)=P
O$(3):PO$(3)=""
1080 IF DR=2 THEN PO$(2)=PO$(3):PO$(3)="
"
1090 IF DR=3 THEN PO$(3)="":NP=NP-1
1100 RETURN
1110 U=100*SQR((MX^2)+(MY^2)*ABS(MZ))
1120 W=U-INT(U)
1130 RETURN
1140 PRINT:PRINT "ARE YOU SURE (Y/N)?"
1150 A$=INKEY$:IF A$="" THEN GOTO 1150
1160 IF A$="Y" THEN CLS:PRINT "O.K. BYE.
. ":END
1170 IF A$="N" THEN PRINT "GREAT!!!":GOS
UB 350
1180 GOTO 110
1190 GOTO 1150
1200 CX=SGN(X-MX):CY=SGN(Y-MY):CZ=SGN(Z-
MZ)
1210 MY=MY+CY:GOSUB 1110:IF W<10 THEN GO
TO 1230
1220 MY=MY-CY
1230 MX=MX+CX:GOSUB 1110:IF W<T THEN GOT
O 1250
1240 MX=MX-CX
1250 MZ=MZ+CZ:GOSUB 1110:IF W<T THEN GOT
O 1270
1260 MZ=MZ-CZ
1270 IF ABS(X-MX)<5 AND ABS(Y-MY)<5 AND
ABS(Z-MZ)<5 THEN GOTO 1290
1280 RETURN
```

```
1290 PRINT "YOU HEAR A SCRATCHING NOISE
";
1300 IF SGN(X-MX)=-1 THEN PRINT "RIGHT,
";
1310 IF SGN(X-MX)=1 THEN PRINT "LEFT, ";
1320 IF SGN(Y-MY)=-1 THEN PRINT "BACKWAR
D, ";
1330 IF SGN(Y-MY)=1 THEN PRINT "FORWARDS
, ";
1340 IF SGN(Z-MZ)=-1 THEN PRINT "UP, ";
1350 IF SGN(Z-MZ)=1 THEN PRINT "DOWN, ";
1360 PRINT "OF YOU"
1370 RETURN
1380 PRINT "THE MONSTER HAS CAUGHT UP TO
 YOU"
1390 FOR DD=1 TO 500:NEXT DD:GOSUB 1700
1400 IF EX<500 OR WE<>1 THEN PRINT "YOU
HAVE NO HOPE OF BATTLING HIM":PRINT "HE
HAS MUNCHED YOU!!":S=0:GOTO 580
1410 FOR DD=1 TO 1000:NEXT DD:GOSUB 1700
1420 BO=(EX*S*G)/200
1430 IF RND(10)>6 AND BO>RND(200)+20 THE
N PRINT "YOU WERE VICTORIOUS!!":GOTO 148
0
1440 FOR DD=1 TO 750:NEXT DD:GOSUB 1700
1450 PRINT "YOU WERE KILLED":PRINT "YOU
SHOULD HAVE KEPT AWAY FROM":PRINT "HIM":
GOTO 1480
1460 FOR DD=1 TO 750:NEXT DD:GOSUB 1700
1470 RETURN
1480 PRINT:INPUT "DO YOU WISH TO PLAY AG
AIN?";A$
1490 A$=LEFT$(A$,1)
```

```
1500 IF A$="Y" THEN RUN
1510 IF A$="N" THEN CLS:PRINT "O.K. BYE.
  ":END
1520 GOTO 1480
1530 CLS 8
1540 FOR XX=1 TO 15
1550 PRINT@ XY, TAB(7) "Vadrand's treasu
re"
1560 PLAY "O4T4L 75EEEEEEFFFFFFGGGGGG"
1570 XY=XY+32
1580 CLS 8
1590 NEXT XX
1600 CLS
1610 RETURN
1620 FOR XX=1 TO 5
1630 SOUND 200,1
1640 NEXT XX
1650 RETURN
1660 FOR XX=1 TO 5
1670 SOUND 10,1
1680 NEXT XX
1690 RETURN
1700 SOUND 150,2
1710 RETURN
1720 PLAY "T4L2002AAACCCO1GGGGGO2AAA"
1730 RETURN
```

# Cavern of the Shadow Thieves

Next, we have an adventure program which creates its own stable map. That is, the map does not change within a game but is totally different from game to game. The action takes place underground in a maze of twenty caves connected by tunnels.

Although the number of possible combinations is not infinite, it is so huge that you are unlikely ever to strike the same cavern system twice in your lifetime. However, the lack of true randomness in the computer's random number generator diminishes the total number of caverns you may visit, although you are unlikely to stumble across the same cavern more than once.

Cavern of the Shadow Thieves takes place in a very odd environment. You are within a maze of twenty caves trying to get to cave number twenty. Each cave has four, and only four, tunnels leading from it to other caves. Some of the tunnels are one-way only, while others allow you to travel back and forth.

Several of the tunnels will contain rare treasures such as platinum shields and sparkling amulets. You can carry up to four items at a time. Other caverns are the homes of unusual inhabitants, such as gruesome gnomes and zany zombies. Each inhabitant will demand a toll of some sort before it will let you pass. If you are carrying the item, the inhabitant will take it off you and allow you to pass; if you do not have it, you will be sent back to the cave you have just left.

You should make a map as you make your way through the system of caves. This way, you will know where the inhabitants are and what bribes they require to let you past. This can save you from a lot of aimless wandering. As we have said, the interconnecting tunnels do not change within the course of a game, so you can plot your map with a great deal of confi-

dence. Note that some caves contain a magic teleportation spell which will move you at random to another cave.

The only goal of this game is to get to cave number twenty, and you will find that the output from the program will assist you in this task, reporting your position and possessions after each move.

```
10 REM CAVERN OF THE SHADOW THIEVES
20 GOSUB 1580
30 DIM A(20,4),R$(4),B$(20),C$(20),D$(20
),E$(20),F$(20)
40 CLS
50 GOSUB 1050:REM BUILD CAVERNS
60 GOSUB 960:REM FILL ARRAYS
70 GOSUB 860:REM HIDE INHABITANTS AND TR
EASURES
80 Q=20:REM TIMER
90 Y=1:X=0
100 CLS
110 REM MAJOR GAME CYCLE
120 GOTO 240
130 CLS:PRINT "**********************
*******"
140 PLAY"T20L4V3003GA"
150 Q=Q-1:IF Q<1 THEN 1410
160 PRINT:PRINT "TIME REMAINING:";Q
170 PRINT:PRINT TAB(6) "YOU ARE IN CAVER
N";Y
180 IF X>0 THEN PRINT "YOU HAVE COME FRO
M CAVERN";X
190 IF R$(1)>"" OR R$(2)>"" OR R$(3)>""
OR R$(4)>"" THEN PRINT "YOU ARE CARRYING
:"
```

```
200 Z=1
210 IF R$(Z)<>"" THEN PRINT Z;" - ";R$(Z)
220 IF Z<4 THEN Z=Z+1:GOTO 210
230 RETURN
240 GOSUB 130
250 IF LEN(F$(Y))=0 THEN 440
260 PRINT:PRINT "THE CAVERN CONTAINS A "
:PRINT F$(Y)
270 IF INKEY$<>"" THEN 270
280 PRINT:PRINT "DO YOU WANT IT (Y/N)"
290 Q$=INKEY$
300 IF Q$<>"Y" AND Q$<>"N" THEN 290
310 IF Q$="N" THEN 440
320 IF R$(1)="" OR R$(2)="" OR R$(3)=""
OR R$(4)="" THEN 400
330 PRINT "YOU ARE CARRYING TOO MUCH"
340 INPUT "WHICH ITEM DO YOU WISH TO DRO
P";S
350 IF S<1 OR S>4 THEN 340
360 T$=R$(S)
370 R$(S)=F$(Y)
380 F$(Y)=""
390 GOTO 430
400 G=1
410 IF R$(G)="" THEN R$(G)=F$(Y):F$(Y)="
":GOTO 430
420 IF G<4 THEN G=G+1:GOTO 410
430 GOSUB 130
440 IF LEN(D$(Y))=0 THEN 650
450 IF ASC(D$(Y))=42 THEN 810
460 PRINT:PRINT "THERE IS A ";D$(Y);" HE
RE,":PRINT "WHO ";
470 FOR TT=1 TO 5
```

```
480 PLAY "T100V3003AFGAFGAFG"
490 FOR DD=1 TO 10:NEXT DD
500 NEXT TT
510 PRINT "WANTS A ";E$(Y):PRINT "TO LET
 YOU PASS"
520 FOR DD=1 TO 2000:NEXT DD
530 G=1
540 IF R$(G)=E$(Y) AND E$(Y)<>"" THEN 60
0
550 IF G<4 THEN G=G+1:GOTO 540
560 PRINT:PRINT "YOU MUST RETURN TO";X
570 FOR DD=1 TO 3000:NEXT DD
580 P=X:X=Y:Y=P
590 GOTO 240
600 PRINT "AND YOU HAVE IT!"
610 FOR DD=1 TO 2000:NEXT DD
620 PRINT "THE ";D$(Y);" VANISHES!":D$(Y
)=""
630 F$(Y)=R$(G)
640 R$(G)=""
650 PRINT:PRINT "TUNNELS LEAD TO";A(Y,1)
;A(Y,2);A(Y,3);A(Y,4)
660 PRINT:INPUT "WHERE DO YOU WANT TO GO
";M
670 IF M=0 THEN Q=Q-5:M=1+RND(16):GOTO 7
20
680 G=1
690 IF A(Y,G)=M THEN 720
700 IF G<4 THEN G=G+1:GOTO 690
710 GOTO 660
720 X=Y
730 Y=M
740 IF Y=20 THEN 760
```

```
750 GOTO 240
760 REM SUCCESS
770 CLS
780 FOR DD=1 TO 30:PRINT TAB(DD) "*":PLA
Y "V30T20003AFG04CED03GFA":NEXT DD
790 PRINT "YOU HAVE MADE IT!!"
800 GOTO 1280
810 REM TELEPORTATION
820 FOR DD=1 TO 30:PRINT TAB(DD) "#":PLA
Y "V30T25003AFG04AFG03AFG":NEXT DD
830 X=Y
840 Y=RND(9)+8
850 GOTO 240
860 REM DISTRIBUTE INHABITANTS TREASURES
870 FOR E=1 TO 16
880 F=RND(18)+2
890 D$(F)=B$(RND(20)):REM INHABITANTS
900 E$(F)=C$(RND(20)):REM BRIBES
910 F=RND(20)
920 F$(F)=C$(RND(20)):REM CAVERN CONTENT
S
930 IF RND(10)>9 THEN D$(F)="*":REM TELE
PORTATION
940 NEXT E
950 RETURN
960 REM CREATE CAVERNS
970 PRINT:PRINT:PRINT "STANBY AS I HIDE
THE TREASURES.."
980 FOR D=1 TO 20
990 READ B$(D)
1000 NEXT D
1010 FOR D=1 TO 20
1020 READ C$(D)
```

```
1030 NEXT D
1040 RETURN
1050 REM CONSTRUCT CAVERN
1060 PRINT:PRINT "STANDBY AS I CONSTRUCT
 THE CAVE"
1070 FOR B=1 TO 20
1080 PRINT TAB(B) "*"
1090 FOR C=1 TO 4
1100 A(B,C)=B+RND(7)-RND(6)
1110 IF A(B,C)=B OR A(B,C)<1 OR A(B,C)>2
0 THEN 1100
1120 NEXT C
1130 IF A(B,1)=A(B,2) OR A(B,1)=A(B,3) O
R A(B,1)=A(B,4) OR A(B,2)=A(B,3) OR A(B,
3)=A(B,4) OR A(B,2)=A(B,4) THEN 1090
1140 IF RND(10)<2 AND B>12 THEN A(B,RND(
4))=20
1150 NEXT B
1160 CLS
1170 RETURN
1180 REM INHABITANTS
1190 DATA HAIRY HOBGOBLIN,BALD BERSERKER
,SKINNY SKELETON,GRUESOME GNOME,CUNNING
CONJURER
1200 DATA CRAZY CENTIPEDE,DEMENTED DWARF
,SAVAGE SHRIEKER,CREEPY CRAWLIE,ROTTEN R
ODENT
1210 DATA TERRIBLE TOAD,STICKY STURGE,GH
ASTLY GHOUL,WICKED WEASEL,LUMPY LEGEND
1220 DATA ZANY ZOMBIE,CROOKED CRAB,WRATH
FUL WRAITH,WEIRD WEREWOLF,GIANT GARGOYLE
1230 REM THE LOOT
```

```
1240 DATA PIECE OF COPPER,MITHRIL COIN,P
LATINUM SHIELD,COPPER HEADBAND,MAGIC SCR
OLL
1250 DATA FABULOUS POTION,WAND OF HEALIN
G,SWORD OF FIRE,SPARKLING AMULET,WAR HAM
MER
1260 DATA PIECE OF COPPER,MITHRIL COIN,P
LATINUM SHIELD,COPPER HEADBAND,MAGIC SCR
OLL
1270 DATA FABULOUS POTION,WAND OF HEALIN
G,SWORD OF FIRE,SPARKLING AMULET,WAR HAM
MER
1280 FOR DD=1 TO 1500:NEXT DD
1290 CLS RND(8)
1300 PRINT@ 192, TAB(2) "YOUR CAVERN MAS
TER RATING IS"
1310 PRINT TAB(12) 100*(100-Q)+2*Q
1320 FOR TT=1 TO 5
1330 PLAY "T250V30O2GAFGAFO4CEDO2GAF"
1340 FOR DD=1 TO 15:NEXT DD
1350 NEXT TT
1360 IF R$(1)>"" OR R$(2)>"" OR R$(3)>""
 OR R$(4)>"" THEN PRINT "YOU LEAVE WITH:
"
1370 FOR T=1 TO 4
1380 PRINT TAB(6) R$(T)
1390 NEXT T
1400 GOTO 1400
1410 FOR DD=1 TO 1500:NEXT DD
1420 PLAY "T50V30O1FGFGACACO2DEDECDCDO1A
FAFGFGFAAA"
1430 CLS0
1440 FOR DD=1 TO 500:NEXT DD
```

```
1450 FOR TT=1 TO 10
1460 PLAY "T250V3002AFG01GFA02AFG"
1470 FOR DD=1 TO 10:NEXT DD
1480 NEXT TT
1490 FOR TT=1 TO 10
1500 PRINT@ 192, TAB(2) "SORRY, FRIEND B
UT TIME IS UP"
1510 FOR DD=1 TO 200:NEXT DD
1520 PRINT@ 192, " "
1530 PLAY "T250V3003ACEG02GFACDE"
1540 NEXT TT
1550 PRINT@ 192, TAB(6) "YOU ARE FINISHE
D"
1560 PLAY "T20V3002AFGDEC01GFAFAFAF"
1570 GOTO 1570
1580 CLS RND(8)
1590 GG=128
1600 PRINT@ GG, "*"
1610 PLAY "T200V3004ACDF05GFA03GFA0"
1620 GG=GG+2
1630 IF GG<159 THEN GOTO 1600
1640 PRINT@ 192, TAB(2) "CAVERNS OF THE
SHADOW THIEVES"
1650 GG=256
1660 PRINT@ GG, "*"
1670 PLAY "T200V3005ACDFG3GFA04GFAD"
1680 GG=GG+2
1690 IF GG<287 THEN GOTO 1660
1700 RETURN
```

# Valley of the Undead

You and a companion have ventured into the Valley of the Undead, intent on searching the many caves and tunnels for the many treasures which are supposed to be lying around waiting to be picked up. Unfortunately, the two of you have become separated in the darkness. Suddenly you hear a scream, your companion has become trapped in a cave and you must get to him before the zombie does. Because of the twisting, turning passages, you are not always able to tell which direction the screams for help are coming from. As an added danger, there is always the chance that the zombie will ignore your friend and come after you!

After each turn you will be asked which direction you wish to go. The code for your input is:

"L" = left
"R" = right
"U" = up
"D" = down
"F" = forward
"B" = back

Line 20 sets up the starting position of your lost friend and line 30 gives you your starting position.

Lines 400 to 490 check your position against the position of your companion after each turn and give you the clues you need.

Variable CC keeps a count of the number of turns left in the game. CC is set in line 50, decremented in line 500 and then tested in line 620. When CC equals zero, we then go to the end routine which gives you the bad news that you were too late to save your friend.

Line 390 provides the one-in-a-hundred chance that the zombies will find you before you find your companion.

This program uses the system of a three-dimensional grid (actually an array), selecting a point inside it and then providing some form of feedback so that the point can be located. In this way many different games can be created. All you need to do is to create an exciting scenario to provide the interest and a method for providing some form of directions to the searcher. Try using this method to create games of your own.

```
10 REM VALLEY OF THE UNDEAD
20 A=RND(10):B=RND(10):C=RND(10)
30 X=RND(10):Y=RND(10):Z=RND(10)
40 CLS
50 CC=16
60 PRINT "YOU'VE GOT LITTLE TIME IN WHIC
H"
70 PRINT "TO FIND AN EXPLORER LOST WITHI
N"
80 PRINT "THE VALLEY OF THE UNDEAD"
90 PRINT:PRINT "()()()()()()()()()()()()
()()()()"
100 FOR DD=1 TO 1000:NEXT DD
110 PRINT@ 160, "HE IS TRAPPED IN THE ZO
MBIE'S"
120 PRINT "CAVE..AND HAS ONLY A SHORT TI
ME"
130 PRINT "BEFORE THE ZOMBIE KILLS HIM!!
!"
140 PRINT "YOU CAN TELL APPROXIMATELY WH
ERE"
150 PRINT@ 288, "HE IS BECAUSE HIS SCREA
MS LEAD"
```

```
160 PRINT "YOU ONWARD AND UPWARD AND LEF
T  AND...."
170 FOR DD=1 TO 1000:NEXT DD
180 PLAY "O5T3L50EEEFFFGGG"
190 FOR DD=1 TO 500:NEXT DD
200 PRINT "IS THAT A SCREAM I HEAR?"
210 PRINT:PRINT TAB(1) "()()()()()()()()
()()()()()()()"
220 PRINT@ 448, "hit any key to continue
"
230 HI$=INKEY$
240 IF HI$="" THEN GOTO 230
250 FOR XX=1 TO 16
260 SOUND 100,1
270 PRINT
280 NEXT XX
290 CLS
300 PRINT TAB(9) "yes it was!!"
310 FOR DD=1 TO 100:NEXT DD
320 PRINT:PRINT "HOWEVER THE SOUND IS MU
FFLED AND"
330 PRINT@ 96, "YOU CANNOT ALWAYS TELL E
XACTLY"
340 PRINT "WHERE IT IS COMING FROM...."
350 FOR DD=1 TO 1000:NEXT DD
360 PRINT " ()()()()()()()()()()()()()()
()"
370 PRINT:PRINT "THE SCREAM APPEARS TO B
E COMING FROM....."
380 FOR DD=1 TO 200:NEXT DD
390 IF RND(100)=1 THEN GOTO 860
400 IF RND(10)>6 THEN LH=LH+1:GOTO 430
```

```
410 IF X>B THEN PRINT "BELOW YOU":FOR DD
=1 TO 50:NEXT DD:SOUND 100,1
420 IF X<B THEN PRINT "ABOVE YOU":FOR DD
=1 TO 50:NEXT DD:SOUND100,1
430 IF RND(10)>6 THEN LH=LH+1:GOTO 460
440 IF Z>C THEN PRINT "TO THE LEFT":FOR
DD=1 TO 50:NEXT DD:SOUND 100,1
450 IF Z<C THEN PRINT "TO THE RIGHT":FOR
 DD=1 TO 50:NEXT DD:SOUND 100,1
460 IF RND(10)>6 AND LH=2 THEN GOTO 490
470 IF Y<A THEN PRINT "AHEAD OF YOU":FOR
 DD=1 TO 50:NEXT DD:SOUND 100,1:GOTO 500
480 IF Y>A THEN PRINT "BEHIND YOU":FOR D
D=1 TO 50:NEXT DD:SOUND 100,1:GOTO 500
490 IF LH=2 THEN PRINT "YOU CANT TELL WH
ERE HE IS!!"
500 CC=CC-1:LH=0
510 PRINT:PRINT "YOU HAVE";CC;"MINUTES L
EFT"
520 PRINT:PRINT "YOU ARE SEARCHING THE C
AVE"
530 PRINT "WHICH WAY DO YOU WANT TO GO"
540 INPUT "(U,D,R,L,F,B)";A$
550 IF A$="U" THEN X=X+1:IF X>10 THEN X=
10
560 IF A$="D" THEN X=X-1:IF X<1 THEN X=1
570 IF A$="R" THEN Z=Z+1:IF Z>10 THEN Z=
10
580 IF A$="L" THEN Z=Z-1:IF Z<1 THEN Z=1
590 IF A$="F" THEN Y=Y+1:IF Y>10 THEN Y=
10
600 IF A$="B" THEN Y=Y-1:IF Y<1 THEN Y=1
610 IF X=B AND C=Z AND A=Y THEN GOTO 750
```

```
620 IF CC>1 THEN CLS:GOTO 360
630 CLS 3
640 PRINT@ 128, TAB(6) "your friend is d
ead"
650 FOR XX=1 TO 5
660 SOUND 250-RR,SS+1
670 RR=RR+30
680 SS=SS+1
690 NEXT XX
700 PRINT@ 256, TAB(8) "YOU HAVE FAILED"
710 FOR XX=1 TO 5
720 SOUND RR,2
730 NEXT XX
740 GOTO 740
750 CLS 2
760 FOR XX=1 TO 3
770 SOUND 100,1
780 SOUND 150,1
790 SOUND 200,1
800 SOUND 250,1
810 NEXT XX
820 PRINT@ 160, TAB(7) "you have found h
im"
830 FOR DD=1 TO 200:NEXT DD
840 PRINT@ 320, TAB(3) "WELL DONE,FAITHF
UL FRIEND"
850 GOTO 850
860 CLS 0
870 PRINT@ 96, TAB(3) "THE ZOMBIES HAVE
FOUND YOU"
880 FOR XX=1 TO 10
890 SOUND 200,2
900 NEXT XX
```

```
910 PRINT@ 192, TAB(5) "THEY HAVE CAUGHT
 YOU!"
920 FOR XX=1 TO 10
930 SOUND 200,2
940 NEXT XX
950 PRINT@ 288, TAB(1) "YOU HAVE DIED,BR
AVE ADVENTURER"
960 PLAY "O1T1L50GGGGGGFFFFFFFFFFEEEEEEE
EEEEEEEE"
970 GOTO 970
```

# Just for Fun

# Gridiron

Here we have a computer version of the American football game, Gridiron. The program gives you the opportunity to test your athletic skills against the Super Rams, a computer Gridiron team that is very hard to beat.

The object of Gridiron is to gain ground against your opponent. You have four turns, or 'downs', to carry the ball and to try and make a gain of ten yards. If at the end of four downs you have not made ten yards, or if at any time you have lost possession of the ball, it is then your opponent's turn to carry the ball and to try to make ten yards. If you, at any time during your turn, gain ten yards, then you have made 'first down' and begin your turn again.

In the real (non-computer version!) game of Gridiron, if a team hasn't gained ten yards after three downs, it is usual, though not compulsory, to kick the ball. This puts the ball as far down field as possible for when your opponent takes possession. In the computer version, the computer will automatically kick the ball, both for you and itself, if ten yards haven't been gained by the end of the third down. It is possible to score points at this stage by kicking a field goal if you are close enough to the goal line to do so. The computer will also take care of this for you.

It is possible for the opposing team to take possession of the ball at any time by intercepting one of your plays.

All the information you require to play Gridiron is displayed on the screen along with prompts from the computer when it is waiting for you to have your turn. You will also be provided with a graphic display of the playing field showing the position of the ball.

When it is your turn to carry the ball, you have a choice of three different plays.

1. This is a relatively conservative play. It will gain you the least amount of ground but with the lowest risk of an interception.
2. This play will give you good gains but with a higher risk of losing possession of the ball.
3. This is a spectacular play which will gain you a lot of ground but with a very high probability that your opponent will intercept the ball.

Please note that it is possible to lose ground on a play. This happens when your opponents push your player back from the line of scrimmage.

To score points, the ball must be carried over the goal line for a touchdown. This results in a score of six points. Then follows the kick for goal or conversion. If successful, you gain an extra point. A field goal from a punt is worth three points.

A real game of Gridiron is played for four fifteen-minute quarters. Our computer version finishes after 100 plays have been made.

```
10 REM GRIDIRON
20 GOSUB 1470
30 GOSUB 70
40 GOSUB 230
50 GOSUB 460
60 GOTO 480
70 CLS 7
80 GOSUB 1680
90 FOR EE=1 TO 10
100 FF=RND(492)
110 POKE 1023+FF,42
120 SOUND 100,2
130 GOSUB 1680
```

```
140 NEXT EE
150 PRINT
160 PRINT@ 32+ND, TAB(9) "***gridiron***
"
170 GOSUB 1530:GOSUB1660
180 PRINT
190 ND=ND+64
200 GG=GG+1
210 IF GG=8 THEN RETURN
220 GOTO 160
230 CLS
240 FOR AA=0 TO 159
250 POKE 1344+AA,175
260 NEXT AA
270 POKE 1381+BB,159
280 IF BB<19 THEN 320
290 IF BB<63 THEN 330
300 IF BB<83 THEN 340
310 BB=0:GOTO 350
320 BB=20:GOTO 270
330 BB=64:GOTO 270
340 BB=84:GOTO 270
350 FOR AA=1 TO 11
360 POKE 1369+AA+BB,96
370 NEXT AA
380 BB=BB+32
390 IF BB=160 THEN 410
400 GOTO 350
410 FOR AA=1 TO 5
420 POKE 1343+AA,96
430 NEXT AA
440 GOSUB 1510
450 RETURN
```

```
460 IF RND(10)<6 THEN PL=1
470 RETURN
480 GOSUB 1200
490 T=T-1:IF T=0 THEN 970
500 GOSUB 1200
510 PRINT@ 64, "SUPER RAMS";S2;,"TIME=";
T:PRINT
520 PRINT "HUMAN TEAM";S1;,"DOWN="DN
530 GOSUB 1600:PRINT
540 PRINT "YARDS TO MAKE FOR FIRST DOWN"
;DG*5:PRINT
550 IF PL=1 THEN PRINT "ITS MY BALL":GOT
O 570
560 PRINT "ITS YOUR BALL":GOTO 620
570 B=RND(3)
580 IF DG>1 THEN B=B+1
590 IF DN=3 THEN B=3
600 IF B>3 THEN B=3
610 GOTO 660
620 PRINT@ 0, "CHOOSE YOUR PLAY";
630 INPUT B
640 IF B<1 OR B>3 THEN 620
650 GOSUB 1560
660 LC=RND(16)
670 IF LC-B<2 THEN GOTO 930
680 IF LC+B<10 THEN DG=DG:YG=0:DN=DN+1:B
$="PLAY":GOTO 720
690 IF LC+B<14 THEN DG=DG-1:YG=1:DN=DN+1
:B$="GAIN":GOTO 720
700 IF LC+B<16 THEN DG=DG-(B+1):YG=B+1:D
N=DN+1:B$="GAIN":GOTO 720
710 IF LC+B>15 THEN DG=DG+1:YG=-1:DN=DN+
1:B$="LOSS"
720 GOSUB 1560
```

```
730 PRINT@ 0, "ITS A";ABS(YG*5);"YARD "B
$:GOSUB 1530:GOSUB 1640
740 GOSUB 1490
750 IF PL=0 THEN DI=DI+YG
760 IF PL=1 THEN DI=DI-YG
770 IF DI<7 OR DI>25 THEN GOTO 1240
780 GOSUB 1510:GOSUB 1560
790 IF DG=<0 THEN DG=2:DN=1:FOR X=1 TO 4
:GOSUB 1530:NEXT X:GOSUB 1650:PRINT@ 0,"
FIRST DOWN":GOSUB 1650:GOSUB 1560:GOTO 4
80
800 IF DN<4 THEN GOSUB 1640:GOTO 480
810 PRINT@ 0, "FORCED TO PUNT":GOSUB 165
0:GOSUB 1530
820 GOSUB 1560
830 PU=RND(5)+5
840 PRINT@ 0, "ITS A";PU*5;"YARD PUNT":G
OSUB 1650
850 GOSUB 1490:GOSUB 1560
860 IF PL=1 THEN DI=DI-PU
870 IF PL=0 THEN DI=DI+PU
880 IF DI<7 OR DI>25 THEN GOTO 1360
890 GOSUB 1510
900 DN=1:DG=2:PL=ABS(PL-1)
910 GOSUB 1650
920 GOTO 480
930 PRINT@ 0, "interception":X=0:FOR X=1
 TO 5:GOSUB 1530:NEXT X
940 GOSUB 1640:GOSUB 1560
950 PL=ABS(PL-1):DN=1:DG=2
960 GOTO 480
970 CLS 0
980 PLAY"T103L75EEEEEEEEEEGGGGGGGGBBBBBB
"
```

```
990 PRINT TAB(8) "THE GAME IS OVER"
1000 PRINT
1010 PRINT TAB(8) "SUPER RAMS-";S2
1020 PRINT
1030 PRINT TAB(8) "HUMAN TEAM-";S1
1040 PRINT
1050 IF S2>S1 THEN GOTO 1150
1060 IF S1>S2 THEN GOTO 1100
1070 PRINT
1080 PRINT TAB(11) "ITS A DRAW"
1090 GOTO 1090
1100 FOR AA=1 TO 9
1110 PRINT TAB(12) "YOU WON!"
1120 SOUND 200,1
1130 NEXT AA
1140 GOTO 1140
1150 FOR AA=1 TO 9
1160 PRINT TAB(13) "I WON!"
1170 SOUND 50,1
1180 NEXT AA
1190 GOTO 1190
1200 FOR X=1 TO 3
1210 POKE 1304,96
1220 NEXT X
1230 RETURN
1240 GOSUB 1560:PRINT@ 0, "TOUCHDOWN!!!"
:X=0:GOSUB 1660
1250 GOSUB 1530
1260 X=X+1:IF X=10 THEN GOTO 1280
1270 GOTO 1250
1280 GOSUB 1650:GOSUB 1560
1290 IF RND(10)>1 THEN 1310
```

```
1300 PRINT@ 0,"MISSED THE CONVERSION":GO
SUB 1640:AD=6:GOTO 1320
1310 PRINT@ 0, "ITS CONVERTED":AD=7:GOSU
B 1640
1320 IF DI<7 THEN S2=S2+AD
1330 IF DI>25 THEN S1=S1+AD
1340 PL=ABS(PL-1):DG=2:DN=1:DI=16
1350 GOSUB 1510:GOTO 480
1360 PRINT@ 0, "FIELD GOAL ATTEMPT":GOSU
B 1640
1370 GOSUB 1560
1380 IF RND(10)>5 THEN GOTO 1400
1390 PRINT@ 0, "MISSED":GOSUB 1640:AD=0:
GOTO 1410
1400 PRINT@ 0, "SUCCESS":GOSUB 1640:AD=3
1410 IF DI>25 THEN S1=S1+AD
1420 IF DI<7 THEN S2=S2+AD
1430 GOSUB 1560
1440 PL=ABS(PL-1):DG=2:DI=16:DN=1
1450 GOSUB 1510
1460 GOTO 480
1470 T=100:DN=1:DG=2
1480 DI=16:RETURN
1490 POKE 1407+DI,175
1500 RETURN
1510 POKE 1407+DI,106
1520 RETURN
1530 SOUND 200,2
1540 GOSUB 1680
1550 RETURN
1560 FOR JJ=1 TO 31
1570 POKE 1023+JJ,96
1580 NEXT JJ
1590 RETURN
```

```
1600 FOR JJ=1 TO 2
1610 POKE 1246+JJ,96
1620 NEXT JJ
1630 RETURN
1640 FOR DD=1 TO 500:NEXT DD
1650 FOR DD=1 TO 250:NEXT DD
1660 FOR DD=1 TO 150:NEXT DD
1670 FOR DD=1 TO 50:NEXT DD
1680 FOR DD=1 TO 10:NEXT DD
1690 RETURN
```

# Mini Golf

We are now moving on to the nine-hole golf course cleverly hidden inside your computer.

You will be able to drive down the fairway, play a chip shot on to the green and then putt accurately to come home under par. Be sure to avoid the sand trap, and don't overshoot the green. Both of these mishaps will result in a one-stroke penalty.

Lines 320 to 420 PRINT up all the relevant information you will need to play each hole.

Line 340 tells you which hole you are currently playing and the number of shots you have taken for the hole so far.

Line 360 tells you the par for the hole and the total number of shots you have taken up to that point.

Line 380 tells you the distance to the hole. This is most important because you will need this information to choose the club you need for the next shot. You have a choice of six clubs: by pressing the keys from "1" to "5" you can hit a 250-yard drive (key number "1") or a 50-yard shot to the green (key number "5") or anything in between. By using the "0" you can play a 20-yard chip shot on to the green when a "5" would hit too far.

Your club selection is registered by lines 430 to 510.

Lines 530 to 640 then work out the distance you have hit the ball, depending on which club you have used.

Line 670 adds one to the number of shots you have taken. The GOSUB 1110 provides the sound of the ball being hit.

Line 680 checks to see if the distance you have hit (DH) equals the distance to the hole (HD). If it does, you are then congratulated for hitting the ball into the hole.

Line 950 checks to see if you are on the green. If so, we then go to the putting routine beginning at line 1130. Once again you are given all the relevant information you will need,

including the distance to the hole. You are now using your putter and have to decide how hard you are going to hit the ball ("1" to "5"). Number "1" is a gentle tap in less than one yard from the cup and number "5" is a 20-foot putt from out near the edge of the green.

After nine holes of golf you will be told how well your score compares with the par for the course.

We are sure that Mini Golf will provide you with many hours of fun and healthy outdoor exercise without ever leaving your keyboard!

```
10 REM MINI GOLF
20 GOSUB 1820
30 GOSUB 100
40 GOSUB 130
50 GOSUB 240
60 GOSUB 320
70 GOTO 420
80 END
90 CLS
100 PRINT@ 192, TAB(2) "WHAT IS YOUR NAM
E";:INPUT B$
110 CLS
120 RETURN
130 FOR AA=1 TO 160
140 POKE 1375+AA,32
150 NEXT AA
160 FOR AA=1 TO 5
170 POKE 1380+A+B,255
180 B=B+32
190 NEXT  AA
200 IF A=1 THEN GOTO 220
210 A=1:B=0:GOTO 160
```

```
220 POKE 1441,48
230 RETURN
240 HN=HN+1
250 HD=RND(450)+100
260 G=INT(HD/30)
270 IF HD<175 THEN PA=3:GOTO 300
280 IF HD<425 THEN PA=4:GOTO 300
290 PA=5
300 PC=PC+PA
310 RETURN
320 PRINT@ 0, "PLAYER: ";B$
330 PRINT
340 PRINT "HOLE ";HN,"SHOTS ";NS
350 PRINT
360 PRINT "PAR  ";PA,"TOTAL ";SU
370 PRINT
380 PRINT "DISTANCE TO HOLE=";HD
390 PRINT
400 PRINT "PAR FOR COURSE=   ";PC
410 RETURN
420 PRINT@ 320, "WHICH CLUB WILL YOU USE
"
430 E$=INKEY$
440 IF E$="" THEN GOTO 420
450 GOSUB 650
460 IF E$="5" THEN GOTO 530
470 IF E$="4" THEN GOTO 550
480 IF E$="3" THEN GOTO 570
490 IF E$="2" THEN GOTO 590
500 IF E$="1" THEN GOTO 610
510 IF E$="0" THEN GOTO 630
520 GOTO 420
530 DH=RND(20)+40
```

```
540 GOTO 670
550 DH=RND(40)+80
560 GOTO 670
570 DH=RND(60)+120
580 GOTO 670
590 DH=RND(80)+160
600 GOTO 670
610 DH=RND(100)+200
620 GOTO 670
630 DH=RND(10)+10
640 GOTO 670
650 PRINT@ 320,"                              ":REM 1
7 SPACES
660 RETURN
670 NS=NS+1:GOSUB 1110
680 IF DH=HD THEN 740
690 HS=HD+20
700 HD=HD-DH
710 IF DH>HS THEN 810
720 GOSUB 1090
730 GOTO 900
740 CLS
750 PRINT@ 192, TAB(5) "## ITS IN THE HO
LE ##"
760 FOR H=1 TO 50:NEXT H
770 SOUND 10,1
780 I=I+1
790 IF I=5 THEN I=0:CLS:GOTO 1530
800 GOTO 740
810 CLS 0
820 PRINT@ 192, TAB(3) "YOU'VE OVERSHOT
THE GREEN"
830 GOSUB 1070
```

```
840 PRINT@ 288, TAB(9) "PENALTY STROKE"
850 GOSUB 1070
860 NS=NS+1
870 CLS
880 HD=-20
890 GOTO 1130
900 IF Y=27 OR Y=26 THEN POKE 1471-Y,255
:GOTO 920
910 POKE 1471-Y,32
920 K=INT(DH/G)
930 Y=Y+K
940 POKE 1471-Y,42
950 IF HD<20 THEN 1130
960 IF Y=27 OR Y=26 THEN GOTO 990
970 GOSUB 320
980 GOTO 420
990 PRINT@ 320, "YOU ARE IN THE BUNKER"
1000 GOSUB 1070
1010 GOSUB 650
1020 PRINT@ 320, "PENALTY STROKE"
1030 GOSUB 1070
1040 GOSUB 650
1050 NS=NS+1
1060 GOTO 60
1070 FOR DD=1 TO 1000:NEXT DD
1080 RETURN
1090 PRINT@ 178, "    ":REM FOUR SPACES
1100 RETURN
1110 SOUND 5,1
1120 RETURN
1130 CLS 0
1140 PRINT@ 192, TAB(6) "YOU ARE ON THE
GREEN"
```

```
1150 PLAY "T200V30O3GGAAFFCCDEFG"
1160 GOSUB 1070
1170 CLS
1180 LL=1216:BB=1236
1190 FOR XX=LL TO BB
1200 POKE XX,32
1210 NEXT XX
1220 LL=LL+32:BB=BB+32
1230 IF LL>1504 THEN GOTO 1250
1240 GOTO 1190
1250 POKE 1386,48
1260 PRINT@ 0, "PLAYER: ";B$
1270 POKE 1386+INT(HD/2),42
1280 PRINT
1290 PRINT "DISTANCE TO HOLE: ";ABS(HD)
1300 PRINT:PRINT "NUMBER OF SHOTS: ";NS
1310 PRINT@ 160, "FORCE OF PUTT (1 TO 5)
";
1320 G$=INKEY$
1330 IF G$="" THEN 1320
1340 SOUND 10,1
1350 IF G$="1" THEN PP=1:GOTO 1410
1360 IF G$="2" THEN PP=RND(2)+1:GOTO 141
0
1370 IF G$="3" THEN PP=RND(4)+3:GOTO 141
0
1380 IF G$="4" THEN PP=RND(6)+5:GOTO 141
0
1390 IF G$="5" THEN PP=RND(10)+10:GOTO 1
410
1400 GOTO 1310
1410 POKE 1386+INT(HD/2),32
1420 IF HD>0 THEN HD=HD-PP:GOTO 1440
```

```
1430 IF HD<0 THEN HD=HD+PP
1440 IF HD=0 THEN GOTO 1470
1450 NS=NS+1
1460 GOTO 1260
1470 SOUND 10,1
1480 NS=NS+1
1490 CLS
1500 FOR DD=1 TO 100:NEXT DD
1510 PRINT "YOU HAVE HOLED YOUR PUTT"
1520 PRINT:FOR DD=1TO 200:NEXT DD
1530 PRINT "SHOTS FOR THIS HOLE=";NS
1540 PRINT:FOR DD=1TO 200:NEXT DD
1550 PRINT "PAR FOR THIS HOLE WAS";PA
1560 PRINT:FOR DD=1TO 200:NEXT DD
1570 FOR DD=1 TO 200:NEXT DD:PRINT "PAR
FOR THE COURSE SO FAR";PC
1580 SU=SU+NS:NS=0
1590 PRINT:PRINT "SHOTS SO FAR";SU
1600 PRINT:PRINT " THIS WAS HOLE NUMBER"
;HN
1610 PRINT:FOR DD=1TO 200:NEXT DD
1620 PRINT:PRINT TAB(6) "any key to cont
inue"
1630 S$=INKEY$
1640 IF S$="" THEN 1630
1650 A=0:B=0:Y=0
1660 IF HN=>9 THEN 1680
1670 CLS:GOTO 40
1680 CLS:PRINT@ 192, "YOUR SCORE WAS";
1690 FS=PC-SU
1700 IF FS>0 THEN PRINT FS;"STROKES UNDE
R PAR":GOTO 1730
1710 IF FS<0 THEN FS=ABS(FS):PRINT FS;"S
HOTS OVER PAR":GOTO 1730
```

```
1720 IF FS=0 THEN PRINT " EVEN WITH PAR"
1730 FOR DD=1 TO 2000:NEXT DD
1740 CLS
1750 SOUND 25,2
1760 PRINT@ 192, TAB(6) "ANOTHER ROUND (
Y/N)";
1770 SCREEN 0,1
1780 A$=INKEY$
1790 IF A$="" THEN 1780
1800 IF A$<>"Y" THEN PRINT@ 256, TAB(12)
 "good bye":GOTO 1800
1810 RUN
1820 CLS
1830 FOR T=1 TO 3
1840 PLAY "T200L100V30O5AGDFEACGFAEGG"
1850 NEXT T
1860 GG=2:FF=30
1870 FOR TT=1 TO 15
1880 PRINT@ GG, "*":PRINT@ FF, "*";
1890 PLAY "T250V30O2GAFGO3AGAO2FFG"
1900 GG=GG+34:FF=FF+30
1910 NEXT  TT
1920 CLS 7
1930 PRINT@ 192, TAB(8) "** MINI GOLF **
"
1940 FOR TT=1 TO 3
1950 PLAY "T200L50V30O3GFGFAFO4CDCFEO3FA
F"
1960 FOR DD=1 TO 10
1970 NEXT TT
1980 CLS
1990 RETURN
```

# Fireman

Now for some fast arcade action with Fireman. You are attempting to save a burning building with your trusty firehose. As each window of the building bursts into flames, you must aim your hose and direct a jet of water into the window before the building catches fire.

To aim your hose at the billowing flames, press the number key (1 to 9) which corresponds to the burning window. The windows are numbered like this:

1 4 7

2 5 8

3 6 9

If you are successful and you save the window, your score will be incremented. If you miss the window, you will lose one of your three lives, so if you miss three times the game is over. At the end of the game, if you wish to play again just press any key, or if you wish to stop playing press the BREAK key to escape from the program.

The program has a difficulty factor: level 1 is the easiest and level 10 is the hardest.

Lines 90 and 100 draw the building and line 110 draws the fireman and the hose. Lines 140 to 240 work out which window has caught fire. Lines 280 to 300 accept the INPUT from the keys using INKEY$, and then check the number you have pressed with the number of the burning window. Line 400 squirts the jet of water into the window if you have aimed accurately.

When you have missed three times, line 330 gives you the bad news. The routine in lines 340 to 380 draws the little scene which follows a miss. This shows the building on fire.

The routine in lines 430 to 450 draws the flames leaping out of the burning window.

This clever little program was written by Mark Farrell.

```
10 REM FIREMAN
20 F=300:CLS2:PRINT@162,"bby programming
 presents :";:FOR T=360 TO 200 STEP- 32:
PRINT@T,"***FIREMAN***";:NEXTT
30 PRINT@388," ";:INPUT"DIFFICULTY(1 - 1
0)";DIF:D=61-6*DIF:IFDIF<1 OR DIF>10 THE
N30
40 L=3
50 CLS 3
60 F=0:T=0
70 PCLS3
80 PMODE 3,1
90 DRAW"C2;BM 10,190;R30U40R40D40L40R20U
40BD40R50U150L100D150"
100 DRAW"C1;BM 20,140U20BU10U20BU10U20R2
0BR10R20BR10R20D20BD10D20BD10D20L20BL10L
20BL10L20U20R20BR10R20BR10R20BU10L20BL10
L20BL10L20U20R20BR10R20BR10R20BU10L20BL1
0L20BL10L20R20U20D20BD10D20BD10D20BR10U2
0BU10U20BU10U20R20D20BD10D20BD10D20BR10U
20BU10U20BU10
110 CIRCLE(200,190),9,2,1,0.5:CIRCLE(200
,173),9,2,1.5:CIRCLE(200,156),9,2:LINE(1
85,159)-(205,179),PSET:LINE(205,169)-(20
0,175),PSET:LINE(80,60)-(80,80),PSET
```

```
120 IFL=0 THEN330
130 SCREEN1,0
140 X=0:Y=0:F=RND(9)
150  IF F=1 THEN X=20 :Y=60
160  IF F=2 THEN X=20 :Y=90
170 IF F=3 THEN X=20 :Y=120
180 IF F=4 THEN X=50 :Y=60
190 IF F=5 THEN X=50 :Y=90
200 IF F=6 THEN X=50 :Y=120
210 IF F=7 THEN X=80 :Y=60
220 IF F=8 THEN X=80 :Y=90
230 IF F=9 THEN X=80 :Y=120
240 IF F=0 THEN GOTO 140
250 GOSUB430
260 T=T+1
270 FOR Z=1 TO 10*D:NEXTZ
280 A$=INKEY$
290 D$=STR$(F)
300 A$=" "+A$
310 IF A$=D$ THEN GOSUB 400
320 GOTO340
330  CLS4:PRINT@202,"BURNT OUT!!!";:PRIN
T@266,"SCORE= ";SC;:GOSUB520:CLS4:RUN
340 IFT=2 THEN Y =Y-7
350 IF T=3 THEN Y=Y-8
360 IF T=4 THEN CLS4:SS=RND(200):SC=SC-S
S:PRINT@266, "SCORE =";SC;
370 IFT=4 THEN PRINT@202,"fire!!!";:L =L
-1 :DRAW"C1;BM0,60E40F20E10F50E4F10E40F50
E20F20":PLAY"L10 O3ABABO2ABABO1ABAB":PAI
NT(0,188),4,1:SCREEN1,0:FOR G=1 TO 1000:
NEXTG
380 IFT=4 THEN50
```

```
390 GOTO 250
400 LINE(X,Y+5)-(200,170),PSET
410 PLAY"L15003ABCDEFGO3ABCDEFGDDEFGO1AB
CDEFG"
420 GOTO460
430 DRAW"BM"+STR$(X)+","+STR$(Y+15)+"E3F
3E5F5E3F2"
440 PAINT (X+2,Y+17),4,1
450 RETURN
460 PRINT@232,"WINDOW SAVED bonus";
470 SC=SC+RND(200)
480 PRINT@299,"SCORE= ";SC;
490 FOR Y=1 TO 1000:NEXT
500 A$="":F=0:T=0
510 PCLS:GOTO 50
520 A$=INKEY$
530 IFA$="" THEN 520
540 RETURN
550 FOR YY=1 TO 400:NEXTYY:RETURN
```

# Snap

Snap is a game of speed and reflexes for children of all ages. It is played just like the card game of Snap, except that in this electronic version the human must match his speed and skills against the mighty computer.

The program puts two cards on to the screen and then displays their values. First the card on the left and then the card on the right is 'turned over' to show a new value. If at any time the two cards on the screen show the same value, you must press a key on the keyboard before the time limit runs out. If you succeed, the number of cards which have been 'turned over' is added to your score. If you fail to press the key in time, the cards are added to the computer's score.

The game has a difficulty factor, from 1 to 10, which you may select at the start of the program. This factor alters the amount of time you have in which to press a key on the keyboard and also the speed at which the cards are turned over.

```
10 REM SNAP
20 CLS 5
30 GOSUB 730
40 GOSUB 830
50 GOSUB 140
60 GOSUB 270
70 GOSUB 590
80 GOSUB 350
90 GOSUB 310
100 GOSUB 590
```

```
110 GOSUB 350
120 GOTO 60
130 STOP
140 CLS
150 FOR AA=1 TO 5
160 POKE 1063+AA+32*BB+CC,239
170 NEXT AA
180 SOUND 200,1
190 BB=BB+1
200 IF BB=7 THEN 220
210 GOTO 150
220 IF CC=11 THEN 250
230 CC=11:BB=0
240 GOTO 150
250 BB=0:CC=0
260 RETURN
270 B=RND(10)+111
280 C=0:F=F+1
290 POKE 1162+C,B
300 GOSUB 800:RETURN
310 D=RND(10)+111
320 C=11:F=F+1
330 POKE 1162+C,D
340 GOSUB 800:RETURN
350 E$=INKEY$
360 IF B<>D THEN E$="0":RETURN
370 IF E$="" THEN 440
380 PRINT@ 288, TAB(10) "you beat me"
390 FOR AA=1 TO 10
400 SOUND 250,1
410 NEXT AA
420 PS=PS+F:F=0
430 GOSUB 590:GOTO 500
```

```
440 PRINT@ 288, TAB(11) "i beat you"
450 FOR AA=1 TO 10
460 SOUND 100,2
470 NEXT AA
480 CS=CS+F:F=0
490 GOSUB 590
500 PRINT@ 288," "
510 PRINT
520 PRINT" YOUR SCORE =";PS
530 PRINT
540 PRINT" MY SCORE =";CS
550 G=0
560 IF CS>99 THEN 610
570 IF PS>99 THEN 670
580 RETURN
590 FOR X=1 TO 2:NEXT X
600 RETURN
610 CLS 3
620 FOR AA=1 TO 16
630 PRINT TAB(13) "i won"
640 GOSUB 800
650 NEXT AA
660 GOTO 660
670 CLS 0
680 FOR AA=1 TO 16
690 PRINT TAB(12) "you won"
700 GOSUB 800
710 NEXT AA
720 GOTO 720
730 GOSUB 140
740 FOR AA=1 TO 4
750 PRINT@ 256+E, TAB(11) "***snap***"
760 E=E+64
```

```
770 GOSUB 800
780 NEXT AA
790 RETURN
800 SOUND50,2
810 RETURN
820 FOR J=1 TO 20:NEXT
830 CLS
840 PRINT "WHAT DEGREE OF DIFFICULTY WOU
LD YOU LIKE?"
850 PRINT:PRINT
860 PRINT"1= HARDEST"
870 PRINT
880 PRINT "10=EASIEST"
890 INPUT Y
900 Z=100*Y
910 RETURN
```

# Biorhythms

Here is a very useful utility program which will give you some insight into the large variety of tasks your computer can undertake. This program provides a guide to your biorhythm cycles.

According to some authorities, our lives are governed by a series of cycles which begin at birth and continue throughout our lives. The three cycles are;

*The physical cycle.* This is a 23-day cycle which is related to endurance, aggressiveness and strength.

*The emotional cycle.* This cycle relates to all emotions such as temper, optimism, depression and so on. This is a 28-day cycle.

*The intellectual cycle.* The mental cycle is a 33-day cycle and it controls the common sense and reasoning powers.

Each of these cycles is different in length, which means that you can be on an intellectual high but a physical low at the same time. This attempts to explain why a person can be very clumsy on a day when he or she is feeling mentally alert, or completely tongue-tied on a day when they feel capable of running a marathon. There are rare occasions when all cycles are on a peak level and the person concerned can do no wrong. If all cycles hit a low point on the same day, you would be well advised to stay in bed.

At the beginning of the program the computer will ask for your name. It will then ask you to enter your date of birth, the year first, followed by the month and then the day, e.g., 1959,6,17. Separate the numbers by commas. Following this the computer will ask you the date for which you wish the forecast worked out. Once again, enter the year, month and

day. The computer will then give you an assessment of your three cycles.

The program is fairly straightforward. The subroutine beginning at line 600 is called up each time you enter a date. This routine converts the date of your birth and the forecast date into a number of days. Line 230 finds the difference between the number of days in the two dates. This number of days is then divided by the length of the three cycles in the subroutine beginning at line 490. Lines 260 to 420 then print out the results of the calculations.

Following the listing, there are a couple of examples of the program in action.

```
10 REM BIORHYTHMS
20 GOSUB 640
30 Y$="YOUR LOW POINT "
40 X$="CYCLE"
50 CLS
60 PRINT:PRINT:PRINT
70 PRINT TAB(5) "PLEASE ENTER YOUR NAME"
80 INPUT Z$
90 FOR DD=1 TO 500:NEXT DD
100 PRINT:PRINT TAB(2) "HELLO ";Z$
110 FOR DD=1 TO 500:NEXT DD
120 PRINT
130 PRINT "PLEASE ENTER YOUR DATE OF BIR
TH"
140 PRINT TAB(8) "(Y,M,D)";:INPUT Y,M,D
150 GOSUB 600
160 J=BD
170 PRINT TAB(7) "DATE FOR FORECAST"
180 PRINT TAB(8) "(Y,M,D)";
190 INPUT Y,M,D
200 CLS
```

```
210 GOSUB 600
220 Z=BD
230 G=Z-J
240 C=23
250 PRINT
260 FOR DD=1 TO 1000:NEXT DD
270 PRINT TAB(7) "PHYSICAL ";X$
280 PRINT TAB(7) Y$
290 GOSUB 490
300 C=28
310 PRINT
320 FOR DD=1 TO 1000:NEXT DD
330 PRINT TAB(7) "EMOTIONAL ";X$
340 PRINT TAB(7) Y$
350 GOSUB 490
360 C=33
370 PRINT
380 FOR DD=1 TO 1000:NEXT DD
390 PRINT TAB(7) "INTELLECTUAL ";X$
400 PRINT TAB(7) Y$
410 GOSUB 490
420 PRINT
430 PRINT "DO YOU WANT TO TRY ANOTHER DA
TE"
440 PRINT TAB(13) "(Y/N)";
450 INPUT A$
460 A$=LEFT$(A$,1)
470 IF A$="Y" THEN FOR DD=1 TO 500:NEXT
DD:GOTO 50
480 GOTO 770
490 REM THIS CALCULATES POINT IN CYCLE
500 PLAY "T250V3002AAAGGGAAAAA"
510 F=G-INT(G/C)*C
520 IF F>C/2 THEN H=C-F:GOTO 540
```

```
530 IF F=0 OR C/2=F THEN PRINT TAB(7) "I
S TODAY":RETURN
540 H=C/2-F
550 H=INT(H)
560 IF H=0 THEN PRINT TAB(7) "IS TODAY":
RETURN
570 IF H<0 THEN PRINT TAB(7) "WAS";ABS(H
);"DAYS AGO":RETURN
580 PRINT TAB(7) "IS IN";H;"DAYS TIME"
590 RETURN
600 REM THIS CALCULATES THE TIME IN DAYS
610 IF M-3>=0 THEN B=B+1:GOTO 620
620 BD=INT(Y*365.25)+INT(M*30.6)+D
630 RETURN
640 CLS:PRINT@ 194, "ENTER A NUMBER FROM
 1 TO 9";
650 INPUT XX
660 GG=XX-1
670 CLS GG
680 FOR DD=1 TO 250:NEXT DD
690 GG=64
700 FOR CC=1 TO 6
710 PRINT@ GG, TAB(11) "BIORHYTHMS"
720 PLAY "T250O4V30L16GGGGGAAAAAGGGGGAAA
AAA"
730 GG=GG+64
740 NEXT CC
750 FOR DD=1 TO 500:NEXT DD
760 RETURN
770 FOR DD=1 TO 1000:NEXT DD
780 CLS 0
790 PRINT@ 192, TAB(12) "GOODBYE"
800 PLAY "T75V30O2GBAFGGGADDAAFFG"
810 GOTO 810
```

```
PLEASE ENTER YOUR NAME
JOSHUA
  HELLO JOSHUA

PLEASE ENTER YOUR DATE OF BIRTH
        (Y,M,D) 1942  7  29

        DATE FOR FORECAST
         (Y,M,D) 1996  6  1

        PHYSICAL CYCLE
        YOUR LOW POINT
        IS TODAY

        EMOTIONAL CYCLE
        YOUR LOW POINT
        IS IN 5 DAYS TIME

        INTELLECTUAL CYCLE
        YOUR LOW POINT
        WAS 14 DAYS AGO

DO YOU WANT TO TRY ANOTHER DATE
              (Y/N)
```

```
      PLEASE ENTER YOUR NAME
GILBERT
  HELLO GILBERT

PLEASE ENTER YOUR DATE OF BIRTH
       (Y,M,D) 1956  5  13


       DATE FOR FORECAST
        (Y,M,D) 1983  9  11


       PHYSICAL CYCLE
       YOUR LOW POINT
       WAS 11 DAYS AGO

       EMOTIONAL CYCLE
       YOUR LOW POINT
       IS IN 1 DAYS TIME

       INTELLECTUAL CYCLE
       YOUR LOW POINT
       IS IN 1 DAYS TIME


DO YOU WANT TO TRY ANOTHER DATE
            (Y/N)
```

# Nine Frogs

Nine Frogs makes very good use of the high resolution graphics of your computer. By using the CIRCLE command to great effect, the program draws the faces of nine frogs on the screen. After a short delay, a fly will appear over the face of one of the frogs. By pressing the correct numerical key (1 to 9 on the top row of the keyboard) you will be able to feed the frog. If you don't press the correct key or the time limit expires, the frog will starve and be removed from the screen.

Each time a frog eats a fly, you will be given a points score. Each time you fail, you will lose points as well as a frog.

The game ends when three frogs have missed their lunch.

The frogs are numbered:

1 4 7

2 5 8

3 6 9

At the beginning of the game you will be asked to enter a level of difficulty. Level 1 is the hardest and 10 is the easiest.

The GOSUB in line 20 sets up the F array with nine elements and fills each element with the value of one. This is used to check if a frog has disappeared from the game. When a frog is removed, the statement F(B)=0 (line 820) sets to zero the element of the array which corresponds to the deceased frog. When a frog has been picked at random in line 510, we then GOTO line 960 where we check to see if the frog we have just picked is still in the game. If it isn't, we go back and select another one. This prevents the fly, drawn by line 660, from being placed on a dead frog.

Lines 70 to 430 draw the frogs using the CIRCLE command. Line 460 is used to bypass the title routine if we have already been through it before.

Line 510 picks which of the nine frogs we are going to DRAW the fly on to. Lines 520 to 600 provide the X and Y coordinates for the DRAW statement in line 660.

Line 620 uses the difficulty level accepted by line 490. Your INPUT is multiplied by 10 and then set equal to the variable Q. Q is then used as the length of time the fly will stay on the frog before the program rushes on. This is done by the FOR/NEXT loop in lines 680 to 730. These lines accept the number you have pressed to feed the frog. They check to see if you have fed the correct frog and go on to the CATCH FAILED routine when the time limit expires.

Lines 810 to 820 clear out the starved frog and send the program back for another turn. Variable DF (for dead frogs) in line 800 checks to see how many frogs have been missed. When DF equals 3, the game is finished.

Nine Frogs was written by Mark Farrell.

```
10 CLS
20 GOSUB 950
30 PLAY "T10002ADADAD03FCFCFC01FGF"
40 PMODE 3,1:PCLS
50 PCLS(3)
60 DRAW"C1"
70 REM DRAW 9 FROGS
80 CIRCLE(50,40),25
90 CIRCLE(50,100),25
100 CIRCLE(50,160),25
110 CIRCLE(120,40),25
120 CIRCLE(120,100),25
130 CIRCLE(120,160),25
140 CIRCLE(190,40),25
```

```
150 CIRCLE(190,100),25
160 CIRCLE(190,160),25
170 CIRCLE(43,30),7
180 CIRCLE(60,30),7
190 CIRCLE(50,40),15,1,1,0,.5
200 CIRCLE(43,90),7
210 CIRCLE(60,90),7
220 CIRCLE(50,100),15,1,1,0,.5
230 CIRCLE(43,150),7
240 CIRCLE(60,150),7
250 CIRCLE(50,160),15,1,1,0,.5
260 CIRCLE(113,30),7
270 CIRCLE(130,30),7
280 CIRCLE(120,40),15,1,1,0,.5
290 CIRCLE(113,90),7
300 CIRCLE(130,90 ),7
310 CIRCLE(120,100),15,1,1,0,.5
320 CIRCLE(113,150),7
330 CIRCLE(130,150),7
340 CIRCLE(120,160),15,1,1,0,.5
350 CIRCLE(183,30),7
360 CIRCLE(200,30),7
370 CIRCLE(190 ,40 ),15,1,1,0,.5
380 CIRCLE(183,90),7
390 CIRCLE(200,90),7
400 CIRCLE(190,100),15,1,1,0,.5
410 CIRCLE(183,150),7
420 CIRCLE(200,150),7
430 CIRCLE(190,160),15,1,1,0,.5
440 F$="E15F15R5L33R28D5R5U5D5L35F15E15"
450 CLS4
460 A=A+1:IF A>1 THEN GOTO 510
470 PRINT@203,"NINE FROGS";
```

```
480 PRINT@265,"BY MARK FARRELL";
490 PRINT@326," ";:INPUT"DIFFICULTY (1-1
0)";C
500 IF C<1 OR C>10 THEN 490
510 B=RND(9)
520 IF B=1 THEN X=50:Y=40
530 IF B=2 THEN X=50:Y=100
540 IF B=3 THEN X=50:Y=160
550 IF B=4 THEN X=120:Y=40
560 IF B=5 THEN X=120:Y=100
570 IF B=6 THEN X=120:Y=160
580 IF B=7 THEN X=190:Y=40
590 IF B=8 THEN X=190:Y=100
600 IF B=9 THEN X=190:Y=160
610 SCREEN 1,0
620 Q=C*10
630 REM DRAW FLY
640 GOTO 960
650 PLAY "T5003ACA"
660 DRAW"S2;C2BM"+STR$(X-5)+","+STR$(Y+3
)+F$
670 REM KEYBOARD SCAN
680 FOR I=1 TO Q
690 A$=INKEY$
700 B$=STR$(B)
710 A$=" "+A$
720 IF B$=A$ THEN GOTO 750
730 NEXT I
740 GOTO 800
750 REM IF CATCH IS SUCCESSFUL
760 PLAY"L5002ABCDEFG"
```

```
770 CLS 0:PRINT@138,"tasty lunch";:L=RND
(100)+100:SC=SC+L:PRINT@202,"bonus=";L;:
PRINT@266,"score=";SC;:FOR G=1 TO 1000:N
EXT G:A$=""
780 A$="":B$="":SCREEN 1,0:GOTO 40
790 REM IF CATCH FAILS
800 CLS0:PRINT@138,"frog starved";:P=RND
(100)+100:SC=SC-P:PRINT@202,"penalty";P;
:PRINT@266,"score=";SC;:FOR G=1 TO 1000:
NEXT G:DF=DF+1:IF DF=3 THEN GOTO 840
810 REM CLEAR STARVED FROG
820 A$="":B$="":LINE(X-30,Y-30)-(X+30,Y+
30),PRESET,BF:F(B)=0
830 GOTO 510
840 CLS RND(8)
850 PRINT@ 198," "; "THREE FROGS ARE DEA
D ";
860 FOR DD=1 TO 3
870 PLAY "T20002AAF1AAGGO1GGFFAAAAP1"
880 NEXT
890 FOR DD=1 TO 2000:NEXT
900 PRINT@ 264," "; "PLAY AGAIN (Y/N)";
910 Z$=INKEY$:IF Z$="" THEN 910
920 IF Z$="N" THEN CLS:PRINT@ 204, "GOOD
BYE":GOTO 940
930 RUN
940 GOTO 940
950 FOR AB=1 TO 9:F(AB)=1:NEXT:RETURN
960 IF F(B)=0 THEN GOTO 510 ELSE GOTO 65
0
```

# Simulations

# Doge of Venice

Doge of Venice is not an easy game to play. The aim of the game is easy to state – to amass wealth through trade and to feed the population of Venice – but this is not simple to achieve. You will be responsible for the administration of the city. This involves the allocation of labour, the building of ships, the maintenance of the army, production of goods for export and the conducting of overseas trade.

You will also have to contend with pirates and with local and overseas wars, not to mention adverse economic conditions and pressures from your own population.

You will be given a points rating at the end of each game year which will give you some indication of how well or otherwise you are doing.

Each year, depending on the size of the population, you will be given a number of labour points to allocate. The way that you divide up these points is very important. You have five areas that require labour: the army, the navy, the merchant marine, the shipyards and the export trade. You can allocate labour to any of these areas (or take labour away from oversupplied areas by entering a negative number), but be very careful to make sure that you have the best possible distribution for the coming year.

At the start of the game you have a choice of how many years you wish to play for. After you have chosen, the computer will then set up the parameters of the game. These starting conditions are not necessarily balanced and may require quite a bit of juggling to produce a workable plan for the first year.

You may move labour back and forth between the five areas as much as you like, but as soon as the unallocated labour total reaches zero the computer will move on to the next phase.

We will now look at the five areas that require labour.

*The army*
The army protects your local towns and mans your overseas outposts. A strong army is required to fight your battles and protect your trade routes. The army makes no profit for you and costs 1000 ducats per point per year to maintain.

*The navy*
The war galleys of Venice protect your merchant shipping from pirates and also fight your overseas wars to keep your trade routes open. The navy makes no profit and, like the army, costs 1000 ducats per point per year to keep up.

*The merchant marine*
The backbone of Venice's wealth is the merchant fleet. These ships sail local waters supplying food and daily necessities to the people of Venice. Larger galleys sail to far-off lands in search of greater riches. The merchant fleet must be as large as you can make it because it supplies the goods which keep your people happy, plus it makes the money you need to finance the expansion of your little empire. Unfortunately your merchant ships are often the target for pirates and they suffer heavily in times of war (especially if you lose!), so they need to be protected by warships at all times.

*Export production*
Successful overseas trading will make Venice rich, so you need to have tradesmen producing fine goods to trade overseas. There is no use sending great fleets of merchant ships off around the world if they have no goods to trade. So make sure that for each point you put into the merchant fleet there is a corresponding point in export production making goods for the ship to carry. You will soon go bankrupt if you send empty ships to all corners of the known world.

*Ship building*
Venice was famous for the quality of its ships. This was largely due to the state-owned ship-building industry. Each labour point you put into this industry will build a 'ready to sail' ship.

Here it is very important to allocate your labour carefully. It would be the height of folly to build ships which then have no crews to sail them because you have insufficient labour in the navy and merchant marine to provide crews. Likewise it is very wasteful to have crews sitting idle on shore because there wasn't enough labour to build ships for them.

After your ships have been built you can then allocate them to the navy as war galleys or to the merchant marine as trading ships.

As you can see, you will be required to juggle labour very carefully between the five areas. Your success as a Doge depends on how good you are at juggling.

Your army can be augmented by hired mercenaries, but mercenary points cost more to maintain than army points. Also, they are only available in limited numbers. When you fire mercenaries there is always the danger that they will not take kindly to this and will run riot through your streets.

The value of your trade with other areas depends to some extent on how well you come out of the various wars which have a habit of breaking out all around you. For this reason it is a good idea to make sure your overseas army garrisons are well manned and your merchant ships are protected by war galleys.

After you have picked your way carefully through all these problems, you still have one more difficulty: your people. If you don't feed them well and keep them prosperous, they will deal very harshly with you.

If you survive as Doge of Venice, you will be given a final score which will relate to the size of Venice's population and to the amount of money in the city treasury at the end of the game.

Doge of Venice is a very complicated game which will present you with a stimulating challenge.

```
10 REM DOGE OF VENICE
20 GOSUB 2930
30 CLS
40 PRINT TAB(6) "***doge of venice***"
50 PRINT "HOW MANY TURNS DO YOU WISH TO
  PLAY FOR (MIN 10,MAX 20)";:INPUT TL
60 IF TL<10 THEN TL=10
70 IF TL>20 THEN TL=20
80 CLS
90 TR=50000:TN=1501:PO=100000:LP=250:MS=
RND(20)+110
100 WS=RND(10)+30:FS=RND(10)+10:VB=100
110 FD=100:LA=RND(10)+5:MA=RND(10)+10
120 MC=2000:LN=WS:GV=100:LM=MS:LE=RND(10
)+20
130 VL=100:VB=100
140 LS=RND(30)+15:UL=250-LS-LE-LM-LN-LA
150 IF UL<0 THEN 90
160 GOSUB 2020
170 CLS(3):PRINT"CHECKING LABOUR"
180 IF UL=0 THEN PRINT"YOU HAVE NO LABOU
R TO ALLOCATE":GOSUB 1980:CLS:GOTO 440
190 PRINT"LABOUR TO ALLOCATE=";UL
200 PRINT"WHERE DO YOU WISH TO EMPLOY TH
EM?"
210 PRINT"(1)=ARMY",,"(2)=NAVY",,"(3)=ME
RCHANT NAVY"
220 PRINT"(4)=EXPORT PRODUCTION","(5)=SH
IP BUILDING"
230 A$=INKEY$:IFA$=""THEN 230
240 IF A$="1" THEN 290
250 IF A$="2" THEN 320 ELSE IF A$="3" TH
EN 350
```

```
260 IF A$="4" THEN 380 ELSE IF A$="5" TH
EN 410
270 PRINT A$
280 GOTO 230
290 PRINT"LABOUR FOR ARMY";
300 INPUT B:IF B>UL THEN 300
310 LA=LA+B:UL=UL-B:GOTO 160
320 PRINT"LABOUR FOR NAVY";
330 INPUT B:IF B>UL THEN 330
340 LN=LN+B:UL=UL-B:GOTO 160
350 PRINT"LABOUR FOR MERCHANT NAVY";
360 INPUT B:IF B>UL THEN 360
370 LM=LM+B:UL=UL-B:GOTO 160
380 PRINT"LABOUR FOR EXPORT";
390 INPUT B:IF B>UL THEN 390
400 LE=LE+B:UL=UL-B:GOTO 160
410 PRINT"LABOUR FOR SHIP BUILDING";
420 INPUT B:IF B>UL THEN 420
430 LS=LS+B:UL=UL-B:GOTO 160
440 GOSUB 2020
450 PRINT"YOUR ARMY STRENGTH=";LA
460 PRINT"HOW MANY FOR LOCAL GARRISONS";
470 INPUT Q:IF Q>LA THEN 470
480 AL=Q:AO=LA-Q
490 IF MH=0 THEN 540:CLS(6)
500 PRINT "MERCENARIES HIRED=";MH
510 PRINT"HOW MANY FOR LOCAL GARRISONS?"
;
520 INPUT R:IF R>MH THEN 520
530 SL=R:SO=MH-R
540 CLS(5):PRINT"NO.OF MERCENARIES FOR H
IRE=";MA
550 PRINT"NO. OF MERCENARIES HIRED=";MH
```

```
560 PRINT"DO YOU WISH TO..."
570 PRINT"(1)-HIRE MERCENARIES(COST";MC;
")"
580 PRINT"(2)-FIRE MERCENARIES","(3)-NON
E OF THE ABOVE"
590 A$=INKEY$:IF A$="" THEN 590
600 IF A$="1" THEN 620 ELSE IF A$="2" TH
EN 650
610 IF A$="3" THEN 800
620 PRINT"HOW MANY";
630 INPUT B:IF B>MA THEN 630 ELSE IF B<0
 THEN 630
640 MH=MH+B:MA=MA-B:GOTO 540
650 PRINT"HOW MANY TO FIRE";
660 INPUT B:IF B>MH THEN 660 ELSE IF B<0
 THEN 660
670 MH=MH-B:MA=MA+B:MI=B*5:CLS(4):X=RND(
100)
680 FOR I=1 TO 500:NEXT
690 IF MI<X THEN 780 ELSE PRINT"the merc
enaries you have fired are rioting"
700 U=1000*RND(10):V=1000*RND(20):W=RND(
10)+2
710 PRINT"THEY HAVE LOOTED YOUR TREASURY
 OF";U;"DUCATS!!!"
720 PRINT"AND PUT";V;"CITIZENS TO THE SW
ORD!!!"
730 PRINT"LABOUR KILLED-"
740 PRINT"EXPORT PRODUCTION=";INT(W/3)
750 PRINT"SHIP BUILDING=";INT(W/3)
760 TR=TR-U:LS=LS-INT(W/3):LE=LE-INT(W/3
)
770 PO=PO-V:GOSUB 2830
```

```
780 PRINT"THE MERCENARIES YOU FIRED LEAV
E QUIETLY"
790 PRINT@ 448, TAB(7) " ";:GOSUB 2830
800 CLS
810 US=LS
820 IF US=0 THEN PRINT"YOU HAVE NO MORE
SHIPS TO BUILD":GOSUB 1980:GOTO 990
830 PRINT"SHIPS BUILT THIS TURN=";US
840 NW=LN-WS:MW=LM-MS
850 PRINT"NO. OF CREWS WITHOUT WAR GALLE
YS=";NW
860 PRINT"NUMBER OF CREWS WITHOUT MERCHA
NT SHIPS=";MW
870 PRINT"DO YOU WISH TO BUILD..."
880 PRINT"(1)=WARSHIPS",,"(2)=MERCHANT S
HIPS","(3)=NONE"
890 A$=INKEY$:IF A$=""THEN 890
900 IF A$="1" THEN 920 ELSE IF A$="2" TH
EN 950
910 IF A$="3" THEN 980
920 PRINT"HOW MANY GALLEYS";
930 INPUT B:IFB>US THEN 930 ELSE IF B<0
THEN 930
940 US=US-B:WS=WS+B:CLS:GOTO820
950 PRINT"HOW MANY MERCHANT SHIPS";
960 INPUT B:IF B>US THEN 960 ELSE IF B<0
 THEN 960
970 US=US-B:MS=MS+B:CLS:GOTO 820
980 IF US>0 THEN PRINT"YOU STILL HAVE SH
IPS TO BUILD":GOTO 820
990 CLS
1000 IF LN>WS THEN WA=WS
1010 IF WS>=LN THEN WA=LN
```

```
1020 IF MS>LM THEN MU=LM
1030 IF LM>=MS THEN MU=MS
1040 PRINT"WARSHIPS AVAILABLE=";WA
1050 PRINT"MERCHANT SHIPS AVAILABLE=";MU
:PRINT "MAX NO. FOR OVERSEAS TRADE=";LE
1060 PRINT "HOW MANY WARSHIPS FOR OVERSE
AS  TRADE";
1070 INPUT WO:IF B>WS THEN 1070
1080 WL=WA-WO
1090 PRINT"HOW MANY MERCHANT SHIPS FOR O
VERSEAS TRADE";
1100 INPUT MO:IF MO>MS THEN 1100 ELSE IF
 MO>LE THEN 1100
1110 ML=MS-MO
1120 CLS RND(8)
1130 LC=RND(100)
1140 GOSUB 1990:CLS RND(8):PRINT TAB(4)
"*****overseas wars*****"
1150 IF LC<86 THEN PRINT"WAR BREAKS OUT
WITH":GOTO 1170
1160 G=RND(5):PRINT"NO OVERSEAS WARS":GO
TO 1300
1170 IF LC<21 THEN PRINT"BYZANTIUM":GOTO
 1250
1180 IF LC<26 THEN PRINT"THE NORMANS":GO
TO 1280
1190 IF LC<36 THEN PRINT"CRETAN REBELS":
GOTO 1270
1200 IF LC<46 THEN PRINT"THE SARACENS":G
OTO 1260
1210 IF LC<56 THEN PRINT"GENOA":GOTO 128
0
```

```
1220 IF LC<66 THEN PRINT"THE INFIDELS AS
SAULTING JERUSALEM":GOTO 1290
1230 IF LC<76 THEN PRINT"THE TURKS IN AF
RICA":GOTO 1260
1240 PRINT"THE TURKS AT ACRE":GOTO 1260
1250 G=RND(15)+5:H=1.6:GOTO 1310
1260 G=RND(10)+5:H=1.2:GOTO 1310
1270 G=RND(5)+5:H=1:GOTO 1310
1280 G=RND(12)+5:H=1.2:GOTO 1310
1290 G=RND(14)+5:H=1.4:GOTO 1310
1300 PRINT"OVERSEAS TRADE VALUE INCREASE
S BY";G;"%":VB=VB+G:GOTO 1330
1310 GOSUB 2170:GOSUB 2000
1320 VB=VB-G:PRINT"OVERSEAS TRADE VALUE
DECREASES BY";G;"%"
1330 PP=RND(100)
1340 GOSUB 1970:CLS RND(8):PRINT TAB(6)
"(((((local wars)))))"
1350 IF PP<91 THEN PRINT"WAR BREAKS OUT
WITH":FOR I=1 TO 1500:NEXT:GOTO 1390
1360 G=RND(5):GOTO1520
1370 PRINT"NO LOCAL WARS"
1380 IF PP<6 THEN PRINT"THE PAPAL STATES
":GOTO 1440
1390 IF PP<11 THEN PRINT"COMACCHIO":GOTO
 1440
1400 IF PP<21 THEN PRINT"THE NORMANS OF
SICILY":GOTO 1450
1410 IF PP<36 THEN PRINT"FRANCE":GOTO 14
60
1420 IF PP<61 THEN PRINT"MILAN":GOTO 145
0
```

```
1430 IF PP<91 THEN PRINT"GENOA":GOTO 145
0
1440 G=RND(5)+5:H=1:GOTO 1470
1450 G=RND(10)+5:H=1.3:GOTO 1470
1460 G=RND(15)+5:H=2
1470 GOSUB 2500
1480 VL=VL-G
1490 PRINT"LOCAL TRADE VALUE DECREASES B
Y";G;"%"
1500 FOR I=1 TO 1500:NEXT:GOTO 1540
1510 VL=VL+G
1520 PRINT"LOCAL TRADE VALUE INCREASES B
Y";G"%"
1530 FOR I=1 TO 1500:NEXT
1540 YY=INT(ML*VL*4)
1550 TR=TR+YY
1560 ZZ=INT(15*VB*MO)
1570 TR=TR+ZZ
1580 GOSUB 1980:CLS RND(8)
1590 PRINT"OVERSEAS TRADE REVENUE =";ZZ;
"DUCATS"
1600 PRINT"LOCAL TRADE REVENUE =";YY;"DU
CATS"
1610 PRINT"DUCATS IN TREASURY=";TR
1620 GOSUB 2830
1630 PRINT"  pirates attack your shippin
g":PLAY"O1;AA-AA-"
1640 RL=RND(50)+50
1650 RP=INT((RL-WL*3)/10)
1660 IF RP<0 THEN RP=0
1670 IF RP>ML THEN RP=ML
1680 ML=ML-RP:MS=MS-RP:LM=LM-RP
```

```
1690 PRINT"LOCAL MERCHANT SHIPS LOST =";
RP
1700 RO=RND(50)+50
1710 RQ=INT((RO-WO*3)/10)
1720 IF RQ<0 THEN RQ=0
1730 IF RQ>MO THEN RQ=MO
1740 MO=MO-RQ:MS=MS-RQ:LM=LM-RQ
1750 PRINT"OVERSEAS MERCHANT SHIPS LOST
=";RQ
1760 GOSUB 2830
1770 GOSUB 2020
1780 CR=RND(30)+85:FD=INT(PO/1000)
1790 FV=INT(ML*CR/100)
1800 FS=FS+FV-FD
1810 PRINT"FEEDING YOUR PEOPLE"
1820 IFFS>=0 THEN 1830 ELSE 2850
1830 PRINT"THERE IS PLENTY OF FOOD FOR A
LL"
1840 GOSUB 2830
1850 CLS RND(8):PRINT"YOU ARE PAYING YOU
R:","ARMY,NAVY AND MERCENARIES"
1860 D=1000*(LA+LN):E=2000*MH
1870 TR=TR-D-E
1880 PRINT"COST FOR";MH;"MERCENARIES =";
E,"COST FOR";LA;"ARMY REGIMENTS AND","
      ";LN;"WARSHIP CREWS =";D
1890 GOSUB 2830
1900 PO=INT(PO*1.07)
1910 ML=INT(PO/2500)
1920 MA=MA+(RND(7)-1)
1930 TL=TL-1:GOSUB 2780
1940 TN=TN+1
1950 GOTO 160
```

```
1960 FOR I=1 TO 1000:NEXT
1970 FOR I=1 TO 500:NEXT
1980 FOR I=1 TO 300:NEXT
1990 FOR I=1 TO 100:NEXT
2000 FOR I=1 TO 50:NEXT
2010 RETURN
2020 PRINT"venices status for year";TN
2030 PRINT"POPULATION-      ";PO
2040 PRINT"LABOUR IN ARMY- ";LA
2050 PRINT"MERCENARIES HIRED";MH
2060 PRINT"LABOUR IN NAVY- ";LN
2070 PRINT"WAR GALLEYS-     ";WS
2080 PRINT"LABOUR IN MERCHANT MARINE";LM
2090 PRINT"NUMBER OF MERCHANT SHIPS-";MS
2100 PRINT"LABOUR IN SHIP BUILDING-";LS
2110 PRINT"LABOUR- EXPORT PRODUCTION-";L
E
2120 PRINT"LABOUR AVAILABLE- ";UL
2130 PRINT"DUCATS IN TREASURY-";TR
2140 PRINT"FOOD STOCKPILE-    ";FS
2150 PRINT"FOOD REQUIRED=     ";FD
2160 GOSUB 2830:CLS:RETURN
2170 ZB=RND(100)
2180 IF TN<1507 THEN ZD=3
2190 IF TN>1506 THEN ZD=2
2200 IF TN>1509 THEN ZD=1.5
2210 GOSUB 2000
2220 ZC=((AO+SO)+INT(WO/2))*ZD
2230 ZF=INT(ZC/H)
2240 PRINT"YOUR CHANCE TO WIN IS";ZF;"%"
:PLAY"ABCFF"
2250 IF ZB=<ZF THEN 2330
```

```
2260 PRINT TAB(5) "***venice is defeated
***"
2270 GOSUB 1980:WW=RND(10)+3
2280 MW=RND(20)+6
2290 AW=RND(3)
2300 SW=RND(3):GOSUB 2390
2310 GOSUB 2830:GOTO 1320
2320 IF INKEY$="" THEN 2320 ELSE 1320
2330 WW=RND(3)
2340 MW=RND(7)
2350 AW=RND(2)-1
2360 SW=RND(2)-1
2370 PRINT TAB(4) "<<<venice is victorio
us>>>":PLAY"O3T40CFFCFCFF":GOSUB 2390
2380 GOTO 1300
2390 WW=INT(WW*H):MW=INT(MW*H):AW=INT(AW
*H)
2400 SW=INT(SW*H):IF WO<WW THEN WW=WO
2410 IF MO<MW THEN MW=MO
2420 IF AW>MO THEN AW=AO
2430 IF SW>SO THEN SW=SO
2440 WS=WS-WW:LN=LN-WW:MS=MS-MW:LM=LM-MW
2450 LA=LA-AW:MH=MH-SW
2460 PRINT"losses:-":GOSUB 2000
2470 PRINT WW;"WARSHIPS",,MW;"MERCHANT S
HIPS":GOSUB 2000
2480 PRINT AW;"ARMY REGIMENTS",SW;"MERCE
NARY REGIMENTS":GOSUB 2000
2490 RETURN
2500 ZB=RND(100)
2510 IF TN<1507 THEN ZD=3
2520 IF TN>1506 THEN ZD=2
2530 IF TN>1509 THEN ZD=1.5
```

```
2540 GOSUB 1990
2550 ZC=((AL+SL)+INT(WL/2))*ZD
2560 ZF=INT(ZC/H):IF ZF>100 THEN ZF=100
2570 PRINT"YOUR CHANCE TO WIN IS";ZF;"%"
:PLAY"O1T10BGFFDDDO4T33BGFFAGBO1T100EEGG
BBDDFFFF"
2580 IF ZB=<ZF THEN 2640
2590 PRINT TAB(4) "***Venice is defeated
***":PLAY"O1T10BBAAAGGGG-G-G-"
2600 FOR I=1 TO 1000:NEXT:WW=RND(10)+3
2610 MW=RND(20)+6:AW=RND(3):SW=RND(3):FS
=INT(FS/H):PRINT"FOOD STOCKPILE REDUCED
BY";INT(100*(1-1/H));"%":PO=PO-1000*H
2620 IFWW>WL THEN WW=WL:IFMW>ML THEN MW=
ML:IF AW>AL THENAW=AL:IF SW>SL THEN SW=S
L
2630 GOSUB 2690:GOSUB 2830:RETURN
2640 PRINT TAB(4) "<<<Venice is victorio
us>>>":PLAY"C"
2650 FOR I=1 TO 1000:NEXT
2660 WW=RND(3):MW=RND(7):AW=RND(2)-1:SW=
RND(2)-1
2670 GOSUB 2690
2680 GOTO 1510
2690 PRINT"LOSSES:"
2700 WW=INT(WW*H):MW=INT(MW*H)
2710 AW=INT(AW*H):SW=INT(SW*H)
2720 IF WL<WW THEN WW=WL:IF ML<MW THEN M
W=ML
2730 IF AL<AW THEN AW=AL:IF SL<SW THEN S
W=SL
2740 PRINTWW;"WARSHIPS",,MW;"MERCHANT SH
IPS"
```

```
2750 PRINTAW;"ARMY REGIMENTS",SW;"MERCEN
ARY REGIMENTS"
2760 WS=WS-WW:MS=MS-MW:LA=LA-AW:MH=MH-SW
2770 LM=LM-MW:LN=LN-WW:RETURN
2780 SC=((LA+LN+LM+LE+LS)*100+TR/2+PO/50
)/(TN-1500)
2790 CLS:PRINT"YOUR SCORE IS ";INT(SC):G
OSUB 2830
2800 IF TL=0 THEN 2810 ELSE RETURN
2810 PRINT"THANK YOU FOR PLAYING DOGE OF
   VENICE"
2820 STOP
2830 PRINT"any key to cont"
2840 IF INKEY$="" THEN 2840 ELSE RETURN
2850 PRINT"your people are starving!!"
2860 PLAY"O3T5FFFF":IF FL=1 THEN 2890
2870 IF FL=2 THEN 2900
2880 PRINT"THEY HAVE GOUGED OUT YOUR EYE
S AND BEATEN YOU TO A PULP!!!":FL=FL+1:G
OTO 1850
2890 PRINT"they have revolted":GOSUB 198
0:PRINT"THEY HAVE CUT OFF YOUR LEFT HAND
 AND BEATEN YOU TO A PULP again!!":GOSUB
 1960:GOTO 1850
2900 PRINT"youve starved your people aga
in!!!"
2910 PRINT"IN PUNISHMENT YOU HAVE BEEN
   EXILED TO ROW ON THE GALLEYS FOR THE
REST OF YOUR LIFE"
2920 PRINT"YOU HAVE LOST":PLAY"O1T1GGG"
2930 CLS 0
2940 PLAY "T5002DCDCEFGO3AAGGFFO2GFDEACA
"
```

```
2950 CLS RND(8)
2960 FOR GG=1 TO 32
2970 POKE 1152+X,61
2980 X=X+1
2990 PLAY "T20003FAA04A"
3000 NEXT GG
3010 PRINT@192, TAB(8) "DOGE OF VENICE"
3020 FOR GG=1 TO 32
3030 POKE 1248+X,61
3040 X=X+1
3050 PLAY "T20003GFA02A"
3060 NEXT GG
3070 FOR DD=1 TO 500:NEXT
3080 RETURN
```

# Gold Rush

For the next twenty years you are the mayor of a booming gold-mining town in the American west of the early 1800s. To keep the population happy, you must keep them well fed. Your gold can be traded for food and for land. The amount of gold your town produces relies on the amount of land it has to work and the population of the town. Each unit of land costs ten units of gold, and each food unit also costs ten units of gold. One food unit will feed approximately forty to forty-five people.

The computer will give you the opportunity to expand the size of the town by buying more land. You will also be given the chance to buy food for the people. If you buy too little food, you may suffer badly at the hands of a hungry mob. On the other hand, if you overfeed the population, the number of people living in your town will increase dramatically, giving you further problems.

The program is quite staightforward. Variable P is the population, variable G is the amount of gold you have, and variable L is the land available. Lines 80 to 120 PRINT your yearly status.

Lines 140 and 190 allow you to buy food and land. Line 230 works out the amount of food you need to feed the people in the town and tells you if you haven't bought enough. Line 240 works out the population for the coming year, based on the amount of food you purchased. Line 260 works out how much gold the townsfolk were able to mine.

The end routines start at line 330. If you have been a good mayor and the town has prospered during your twenty years of government, lines 360 to 440 PRINT out the results of your hard work.

If you allow the population to fall below 150, the town isn't able to support itself, so lines 450 to 660 tell you that you have

made the town bankrupt. This routine is also called up if you try to spend more gold than you have in the bank. Note the odd sounds produced by the sound routine in lines 500 to 650.

Lines 670 to 860 work out what happens if you starve the townsfolk: first, line 750 tells you the people are starving, then line 760 gives you a 50–50 chance of surviving the crisis. If you weather the storm, line 780 tells you that you still have the support of the people, and line 800 sends the program back into the main part of the game so that you may attempt to correct the situation in the next year. If you lose popular support, line 810 gives you the bad news.

```
10 REM GOLD RUSH
20 GOSUB 880
30 P=400
40 G=1000
50 L=100
60 FOR Y=1 TO 20
70 CLS
80 PRINT@ 72, "YEAR:";Y;"OF TWENTY"
90 PRINT
100 PRINT "POPULATION:";INT(P)
110 PRINT "AMOUNT OF LAND:";L
120 PRINT "GOLD IN BANK:";G
130 PRINT
140 INPUT "HOW MUCH LAND TO BUY";W
150 G=G-W*10
160 IF G<0 THEN PRINT "NO GOLD LEFT":GOT
O 450
170 L=L+W
180 PRINT
190 INPUT "HOW MUCH FOOD TO BUY";W
```

```
200 G=G-W*10
210 IF G<0 THEN PRINT "NO GOLD LEFT":GOT
O 450
220 R=RND(5)+5
230 IF P-W*R*5>P/4 THEN PRINT:PRINT " no
t enough food to feed people":GOTO 670
240 P=P+RND(W*R*5-P)
250 IF P<150 THEN PRINT "THE TOWN HAS LE
SS THAN 150":PRINT "PEOPLE LEFT":GOTO 45
0
260 G=G+INT(P*L/50)
270 FOR DD=1 TO 500:NEXT
280 CLS RND(8)
290 FOR DD=1 TO 500:NEXT DD
300 NEXT Y
310 GOTO 320
320 FOR DD=1 TO 1500:NEXT DD
330 CLS 2
340 PRINT TAB(8) "you have retired"
350 PRINT:PRINT
360 PRINT "GOLD IN BANK=";G
370 PRINT
380 PRINT "LAND OWNED-  ";L
390 PRINT
400 PRINT "POPULATION IS NOW";INT(P)
410 FOR GG=1 TO 6
420 PLAY "T250V3002FFFFAAAACCCCEEEE"
430 NEXT GG
440 GOTO440
450 FOR DD=1 TO 1000:NEXT DD
460 CLS 0
470 PLAY "T100V3002GGGAAAAAFFFFFFGGGGGGG
GGGGGG"
```

```
480 PRINT@ 192, TAB(8) "YOU ARE BANKRUPT
"
490 B$="T100V30"
500 FOR GG=1 TO 50
510 XD=RND(3)
520 IF XD=1 THEN C$="01"
530 IF XD=2 THEN C$="02"
540 IF XD=3 THEN C$="03"
550 PLAY B$:PLAY C$
560 DX=RND(7)
570 IF DX=1 THEN A$="A"
580 IF DX=2 THEN A$="B"
590 IF DX=3 THEN A$="C"
600 IF DX=4 THEN A$="D"
610 IF DX=5 THEN A$="E"
620 IF DX=6 THEN A$="F"
630 IF DX=7 THEN A$="G"
640 PLAY A$
650 NEXT GG
660 GOTO 660
670 FOR DD=1 TO 1000:NEXT DD
680 CLS 3
690 FOR GG=1 TO 6
700 SOUND 10,5
710 FOR DD=1 TO 25:NEXT DD
720 SOUND 5,5
730 FOR DD=1 TO 25:NEXT DD
740 NEXT GG
750 PRINT@ 128, TAB(4) "THE PEOPLE ARE S
TARVING"
760 IF RND(10)<6 THEN GOTO 810
770 FOR DD=1 TO 1500:NEXT DD
```

```
780 PRINT@ 224, TAB(5) "they still suppo
rt you"
790 FOR DD=1 TO 2000:NEXT DD
800 GOTO 240
810 PRINT@ 224, TAB(5) "you have been ly
nched"
820 FOR GG=1 TO 10
830 PLAY "T250V30O4GGGGGAAAAAAGGGGGAAAAA
GGGGGGGAAAAAAA"
840 FOR DD=1 TO 50:NEXT DD
850 NEXT GG
860 GOTO 860

880 XX=RND(8)
890 IF XX=1 THEN GOTO 880
900 CLS XX
910 FOR SS=20 TO 100 STEP 5
920 SOUND SS,1
930 FOR DD=1 TO SS*2:NEXT DD
940 NEXT SS
950 H=160
960 FORAA=1TO6
970 PRINT@ H, TAB(10)"GOLD RUSH"
980 SOUND RND(25)+100,1
990 FOR DD=1 TO 200:NEXT DD
1000 H=H+32
1010 NEXT AA
1020 FOR DD=1 TO 500:NEXT DD
1030 RETURN
```

# Farmer Jack

Farmer Jack is a simulation of farming life in the 1950s. The program begins with you, as Farmer Jack, aged about thirty, and it ends when the goodly farmer retires at the age of fifty.

The object of the game is to manage your farm and run it at a profit to provide a comfortable retirement for your later years.

At retirement age all your assets are added up. These will include your bank account, the value of your land and a proportion of the value of your capital equipment.

Farmer Jack must juggle his available finances and make decisions about which crops to plant, whether or not to mechanize his operations, should he irrigate his crops and is it wise to buy more land. He must also cope with rising labour costs and the occasional natural disaster.

If Farmer Jack weaves his way successfully through all these decisions and problems, the computer will reward him with the title of SUPER FARMER.

At the start of each year the computer will give you a complete list of all the vital facts and figures that you will need to know about Golden Creek Farm. Read through this information very carefully each turn: many of the decisions you will have to make will be based on these facts.

If credit is available at the local bank, you will be given the opportunity to borrow money, but please be sure that you can meet the repayments. You will also find it is very good to have enough cash on hand to pay your running costs each year. If you have a negative bank balance at the end of any year, you will be declared bankrupt. The shame of this could stay with you for the rest of your life.

Each year you will also be given the opportunity to buy capital equipment and more land. Expanding the size of your operation is necessary for economic growth. You must also

consider the benefits of mechanization to your crop yield. And the computer will give you the chance to sell off parcels of land if the need for this arises.

The next decision that will face you concerns irrigation. Although you may find that this is expensive, it should give an increased yield at harvest time.

Another important decision that you will then be required to make is which crop to plant? The computer lists them in order of profitability. But beware, planting the same crop all the time may exhaust your land. It may be wise to go fallow from time to time to conserve your soil.

After Farmer Jack has made his decisions and put in much hard work, he must then wait to see how well he has done at harvest time. It is during this time that many heartbreaking disasters can occur, events that are completely out of Farmer Jack's control. Drought, flood or a blight could devastate his crops, or perhaps he might fall foul of the Farm Labourers' Union if he doesn't agree to their requests for more pay. All Farmer Jack can do at this time is just sit it out and wait to see what happens.

Inflation will make the price of land rise during the course of the game. This increase will also be reflected in the wages he must pay his farm workers. As your farm grows, it may become necessary to hire more workers to get the jobs done.

Every so often other problems will plague you. Farm machinery may let you down and need to be replaced. Or perhaps your barn may catch fire. There isn't much you can do about problems such as these except grin and bear them. A farmer's life wasn't meant to be easy!

The program of Farmer Jack is essentially just one big loop from line 130 to line 2460. We keep going through this loop until Farmer Jack's age reaches fifty (variable FA in line 2430). We drop out of the loop if the good farmer goes bankrupt. This, at first glance, may not seem to be a very good programming technique, but after trying several methods it was the one that worked the best.

Line 110 sets up your opening bank balance (BB), gives you a tractor (TR) and a truck (TK). It also gives you one barn

(BR) and sets the starting size of Golden Creek Farm at 300 acres (AC).

Line 120 gives you the amount of money the bank is prepared to lend (AL). And it sets your starting age (FA), the wages of your workers (PR), the number of workers (LB) and the current price of land (LC).

Lines 130 to 300 print out all the vital information you need each year. The routine in lines 330 to 430 allows you to borrow money from the bank and works out your repayments.

The section in lines 500 to 630 checks to see if you need to replace a tractor, truck or barn. Lines 670 to 1000 allow you to make your purchases each year. Lines 1020 to 1150 permit you to sell land if this is required.

The irrigation of crops is handled by lines 1160 to 1250. The section 1270 to 1560 allows you to choose which crops you are going to plant, and works out all the relevant computations resulting from this, such as the exhaustion of the soil and the cash value of the crop.

The section in lines 1580 to 1720 controls the natural disasters which may befall you and affect your harvest yield.

The unions and their wage demands are handled by lines 1740 to 2020. Lines 2040 to 2090 take into account all that has happened during the course of the year and work out how much money you have made from the sale of your crop.

The next routine, lines 2110 to 2220, then advises you of your expenses for the year and gives you your net income.

The breakdowns and other problems that occur from time to time are controlled by lines 2230 to 2410. Lines 2430 to 2460 check to see if Farmer Jack has gone bankrupt or if he has reached retiring age. If all is well then line 2460 returns us to the start of the loop for the next year in the life of Farmer Jack.

Lines 2470 to the end of the program arrange for the sale of the farmer's assets and work out his retirement conditions. You are then told how comfortable or otherwise your retirement is likely to be.

This type of program is an excellent one for continuous adjustment and refinement. We have given you the basis of

what could be an extremely large and complex economic simulation. It would be a good exercise for you to develop your programming skills by refining this program and tailoring it to your own requirements.

It is basically an easy program to understand and we have tried to make it simple to follow, so please your own abilities to expand it into a unique game of your own.

```
10 REM FARMER JACK
20 CLS
30 FOR P=1 TO 4
40 PLAY"T250V3003GGGCCCFFFGGGFFFAAAP2"
50 NEXT
60 CLS(4)
70 PRINT@160,"  WELCOME!! YOU ARE FARMER
 JACK. YOUR TASK IS TO MANAGE GOLDEN  CR
EEK FARM FOR THE REST OF YOUR LIFE OR UN
TIL YOU RETIRE          ";
80 FOR I=1 TO 3000:NEXT
90 PLAY "T2500lV30AAAAAAGGGGGGO2CCCCCCAA
AAAAO1CCCCCCCC"
100 CLS
110 BB=3000:TR=1:TK=1:BR=1:AC=300
120 AL=5000:FA=RND(11)+29:PR=2000:LB=0:L
C=100
130 PRINT@0,"YOUR SITUATION IS"
140 PRINT"BANK BALANCE = $";BB
150 PRINT"YOUR AGE IS ";FA
160 PRINT"YOU HAVE"
170 PRINTTR;"TRACTORS      ";
180 PRINTHD;"HEADER"
190 PRINTHV;"HARVESTER     ";
```

```
200 PRINTTK;"TRUCKS"
210 PRINTAC;"ACRES",
220 PRINTBR;"BARNS"
230 IF IB=0 THEN PRINT" NO";
240 PRINT" IRRIGATION ON YOUR FARM"
250 PRINT"EXHAUSTION LEVEL OF LAND =";EA
;"%"
260 PRINT"NUMBER OF LABOURERS = ";LB
270 PRINT"LABOURERS PAY RATE IS $";PR
280 PRINT"LAND COST /ACRE=$";LC
290 PRINT"AMOUNT OF MONEY BANK CAN LOAN
IS $";AL
300 PRINT TAB(3)"";:GOSUB1010:CLS
320 IF AL=0 THEN 480
330 PRINT@130,"DO YOU WISH TO BORROW MON
EY":PRINT TAB(6) "FROM THE BANK (Y/N)";
340 A$=INKEY$:IFA$="N" THEN 460
350 IFA$<>"Y" THEN340
360 PRINTA$:PRINT"HOW MUCH DO YOU WISH T
O BORROW";:INPUTMB
370 IF MB>AL THEN PRINT" TOO MUCH. YOU C
AN ONLY BORROW $";AL:GOTO360
380 YL=50-FA
390 IFYL>=20 THEN YL=20
400 LT=LT+MB:AL=AL-MB
410 RP=INT(LT*2/YL)+5
420 BB=BB+MB
430 PRINT"YOUR ANNUAL REPAYMENTS ARE $";
RP
460 PRINT
470 PRINT"YOUR BANK BALANCE IS $";BB:GOS
UB1010
480 IFTR=1 GOTO520
```

```
490 FG=1
500 IFTR=0 THEN PRINT"YOU MUST BUY A TRA
CTOR(COST $5000)
510 IF TR=0 THENBB=BB-5000
520 TR=1
530 IF TK=0 GOTO550
540 IF AC/TK <999 THEN 590
550 PRINT"YOU NEED BUY ANOTHER TRUCK (CO
ST $10000)"
560 FG=1
570 BB=BB-10000
580 TK=TK+1
590 IF BR=0 THEN 610
600 IFAC/BR<999 THEN660
610 PRINT"YOU MUST BUILD ANOTHER BARN":P
RINT"(COST $3000)
620 FG=1:BB=BB-3000:BR=BR+1
630 GOSUB1010
660 CLS:FG=0
670 PRINT"YOUR BANK BALANCE IS $";BB
680 PRINT" DO YOU WISH TO BUY:-"
690 IF HD=1 THEN 700 ELSE PRINT"HEADER(1
)-$15000"
700 IF HV=1 THEN720
710 PRINT"HARVESTER(2)-$15000"
720 PRINT"MORE LAND(3)-$";LC;"/ACRE"
730 PRINT"NO PURCHASES(4)"
740 C$=INKEY$:IF C$="" THEN740
750 ON VAL(C$) GOTO 800,880,960,1040
800 '
810 IF HD=1 THEN PRINT"YOU ALREADY HAVE
A HEADER":FORI=1TO1000:NEXT:GOTO660
820 BB=BB-15000
830 HD=1
```

```
840 PRINT"YOU HAVE PURCHASED A HEADER"
850 GOSUB1010
870 GOTO660
880 '
890 IFHV=1 THEN PRINT"YOU ALREADY HAVE A
 HARVESTER":FORI=1TO1000:NEXT:GOTO660
900 BB=BB-15000
910 PRINT"YOU HAVE PURCHASED A HARVESTER
"
920 GOSUB1010
940 HV=1
950 GOTO660
960 PL=1:PRINT"HOW MANY ACRES DO YOU WIS
H TO   PURCHASE";:INPUTA1
965 IF BB/LC<1 THEN PRINT "YOU CAN'T AFF
ORD TO BUY LAND":GOSUB1010:GOTO660
970 IF A1*LC>BB THEN PRINT"YOU CAN ONLY
PURCHASE";INT(BB/LC);"ACRES":GOSUB1010:G
OTO660
980 BB=BB-(LC*A1)
990 AC=AC+A1
1000 GOSUB1010:GOTO660
1010 PRINT TAB(6)"any key to continue"
1020 IF INKEY$="" THEN1020 ELSE FOR I=1
TO 1000:NEXT:RETURN
1040 IF PL=1 THEN1150 ELSE CLS
1050 PRINT@130,"DO YOU WISH TO SELL ANY
LAND":PRINT TAB(12)"(Y/N)";
1060 A$=INKEY$:IFA$=""THEN1060ELSEIFA$="
N" THEN PRINT:GOTO 1150
1070 IFA$<>"Y" THEN1050
```

```
1080 PRINT"HOW MANY ACRES DO YOU WISH TO
 SELL";:INPUTA2
1090 IFA2>AC THEN 1080
1100 AC=AC-A2
1110 BB=BB+(LC*A2)
1120 PRINT"YOU HAVE SOLD ";A2;"ACRES AT
$";LC;"/ACRE"
1150 CLS(1)
1160 PL=0:PRINT@130,"DO YOU WISH TO IRRI
GATE YOUR      LAND(COST $1/ACRE)(Y/N)";
1170 A$=INKEY$:IFA$=""THEN1170ELSEIF A$=
"N" THEN 1270
1180 IFA$<>"Y"THEN1150
1190 II=1.3
1200 BB=BB-AC
1210 PRINT
1220 PRINT"YOU HAVE IRRIGATED YOUR CROPS
   FOR THIS YEAR;"
1230 GOTO1250
1250 GOSUB1010.
1270 II=1:CLS:PRINT
1280 PRINT"WHICH CROP DO YOU WISH TO SOW
 ?"
1290 PRINT"BARLEY(1)"
1300 PRINT"WHEAT (2)"
1310 PRINT"PEAS  (3)"
1320 PRINT"FALLOW(4)"
1330 IFEA>99 THEN PRINT"YOUR LAND IS RUN
 DOWN. YOU CAN":PRINT"ONLY PLANT PEAS OR
 LEAVE IT FALLOW."
1340 C$=INKEY$:IF C$="" THEN 1340
1345 FOR I=1TO500:NEXT
```

```
1350 IFC$="1" THEN PRINT"YOU HAVE PLANTE
D BARLEY":GOTO1410
1360 IFC$="2" THEN PRINT"YOU HAVE PLANTE
D WHEAT":GOTO1470
1370 IFC$="3" THEN FOR DD=1 TO 500:NEXT:
PRINT"YOU HAVE PLANTED PEAS":GOTO1530
1380 IF C$<>"4" THEN1340
1390 FOR DD=1 TO 500:NEXT
1400 PRINT"YOUR LAND IS FALLOW THIS YEAR
":FOR DD=1 TO 500:NEXT:EA=0:CLS:GOTO 212
0
1410 AP=1
1420 IF EA>99 THEN PRINT"YOU CAN'T PLANT
 THIS CROP":FORI=1TO500:NEXT:GOTO1340
1430 AW=1
1440 AB=1+(RND(5)-1)/10
1450 EA=EA+15
1460 GOTO1560
1470 AP=1
1480 IFEA>99 THEN PRINT"YOU CAN'T PLANT
THIS CROP":FOR I=1TO100:NEXT:GOTO1340
1490 AB=1
1500 AW=.7+RND(5)/10
1510 EA=EA+10
1520 GOTO1560
1530 AB=1
1540 AW=1
1550 AP=.5+RND(5)/10
1560 GOSUB1010
1580 RF=RND(10)
1590 DR=100
1600 IFRF=1 THEN DR=50-RND(50)
1610 FL=100
```

```
1620 IFRF=10 THEN FL=50-RND(50)
1630 IFRF=1 THEN CLS(4):PRINT@192,TAB(10
)"DROUGHT":PRINT100-DR;"% OF YOUR CROP I
S RUINED"
1640 IF RF=9 THEN FL=100-RND(50)
1650 IF RF=9 THEN CLS(3):PRINT@192,TAB(8
)"EXCESSIVE RAINS":PRINT100-FL;"% OF YOU
R CROP IS RUINED"
1660 IF RF=2 THEN DR=100-RND(50)
1670 IF RF=2 THEN CLS(4):PRINT@192,TAB(8
)"LACK OF RAIN":PRINT100-DR;"% OF YOUR C
ROP IS RUINED"
1680 IF RF=10 THEN CLS(3):PRINT@192,TAB(
12)"FLOOD":PRINT 100-FL;"% OF YOUR CROP
IS RUINED"
1690 IF RF=3 THEN CLS(7):PRINT@192,TAB(1
2)"BLIGHT":PRINT"RUINS 10% OF YOUR YOUR
CROP":BL=.9:GOTO 1720
1700 BL=1
1710 IF 3<RF AND RF<=8 THEN CLS:SK=1:GOT
O1740
1720 PRINT:GOSUB1010
1730 SK=1: CLS:IF LB=0 THEN2040
1740 IF LB=0 THEN 2040:PRINT"LABOURERS D
EMAND A PAY INCREASE!"
1750 IF MU=1 GOTO 1830
1760 PI=RND(20)
1770 PRINT "THEY DEMAND A";PI;"% PAY RIS
E!"
1790 ST=RND(100)
1800 PRINT "THERE IS A";ST"% CHANCE THAT
 THEY":PRINT@128,"WILL STRIKE IF IT IS N
OT GRANTED"
```

```
1810 IF MU=0 GOTO 1880
1820 FORI=1TO1000:NEXT
1830 PRINT "BECAUSE YOU REFUSED THEIR CL
AIM LAST YEAR THE UNIONS HAVE":PRINT"INC
REASED THEIR CLAIM"
1840 PI=INT(PI*1.5)+3
1850 ST=INT(ST*2)+5
1860 PRINT"THE UNIONS NOW DEMAND A";PI;"
% PAY RISE!"
1870 PRINT"THE CHANCE OF THEM STRIKING I
S":PRINTST;"%".
1880 PRINT
1890 PRINT "DO YOU WISH TO PAY THE INCRE
ASE (Y/N)"
1900 A$=INKEY$:IF A$="" THEN1900
1910 IF A$="Y" THEN PR=INT(PR*PI/100)+PR
:MU=0:SK=1:PI=0:ST=0:GOTO 2020
1920 IFA$<>"N" THEN1890
1930 CLS
1940 PRINT"THE UNION LEADERS ARE IN":PRI
NT"CONSULTATION":FOR I=1 TO 1000:NEXT
1950 UB=RND(100)
1960 IF UB>ST THEN PRINT"THE UNION HAS D
ECIDED NOT TO    STRIKE":MU=1:SK=1:GOTO
2020
1970 PRINT"THE LABOURERS ARE OUT ON STRI
KE"
1980 PRINT"YOUR HARVEST IS GOING TO BE":
PRINT"REDUCED BY 20%!"
1990 PRINT "IT WOULD HAVE BEEN WORSE IF
YOUR FAMILY HADN'T HELPED"
2000 MU=0
2010 SK=.8
```

```
2020 GOSUB1010
2030 CLS
2040 IF HD=0 THEN DI=1 ELSE DI=1.5
2050 IF HV=0 THEN VI=1 ELSE VI=1.5
2060 PRINT@98,"I AM WORKING OUT YOUR HAR
VEST":PRINT TAB(12)"RETURN":FOR I=1 TO 1
000:NEXT
2070 HR=INT(AC*AP*AB*AW*FL/100*DR/100*BL
*SK*DI*VI*II*20*(130-EA)/100)
2080 PRINT@192,"YOUR HARVEST RETURN IS $
";HR
2090 PRINT@384,"";:GOSUB1010
2110 BB=BB+HR:CLS
2120 PRINT "YOUR BANK BALANCE IS $";BB
2130 BB=BB-(PR*LB):IF LB=0 THEN2150
2140 PRINT "LESS $";PR;"PER LABOURER FOR
    WAGES LEAVES YOU WITH $";BB
2150 BB=BB-RP:PRINT"LESS YOUR LOAN REPAY
MENT OF":PRINT"$";RP;"LEAVES YOU WITH $"
;BB
2160 LB=INT(AC/500)
2170 X=RND(2)
2180 IF X=1 THEN AL=AL+1000
2190 LI=RND(15)
2200 LC=LC+INT(LI*LC/100)
2210 GOSUB1010
2220 CLS RND(8)
2230 FG=0
2240 FOR P=1 TO 2
2250 PLAY"T250V3001CCCFFFGGGGGP2D2AAAFFF
CCCCCP2"
2260 NEXT
2270 X=RND(33-INT(AC/1000))
```

```
2280 IF AC>32000 THEN X=1
2290 IF X=1 AND TR>0 THEN PRINT "YOUR TR
ACTOR BREAKS DOWN":TR=0:FG=1
2300 X=RND(20-TK)
2310 IF TK>19 THEN X=1
2320 IF X=1 AND TK>0 THEN PRINT "ONE OF
YOUR TRUCKS BREAKS DOWN":TK=TK-1:FG=1
2330 IF RF=1 THEN F3=15
2340 X=RND(20-F3):F3=0
2350 IF X=3 AND BR>0 THEN PRINT"YOUR BAR
N BURNS DOWN":BR=BR-1:FG=1
2360 X=RND(20)
2370 IF X=1 AND HD=1 THEN PRINT"HEADER B
REAKS DOWN":HD=0:FG=1
2380 IF X=2 AND HV=1 THEN PRINT"HARVESTE
R BREAKS DOWN":HV=0:FG=1
2390 IF X=3 THEN PRINT "IRRIGATION BORE
DRIES UP":PRINT"PAY COST OF $2000 TO SIN
K NEW   BORE":BB=BB-2000:FG=1
2400 IF FG=0 THEN2430
2410 GOSUB1010
2430 FA=FA+1:FG=0
2440 IF FA=50 THEN FOR DD=1 TO 1000:NEXT
:CLS:PRINT@192, "*** FARMER JACK HAS RET
IRED ***":GOTO 2470
2445 IF RE=0 AND BB+AL<0 THEN PRINT "YOU
 ARE IN THE RED. THE BANK HAS DECIDED TO
 GIVE YOU CREDIT THIS YEAR":RE=1:GOSUB10
10:GOTO130
2450 IF RE=1 AND BB+AL=<0 THEN PRINT "FA
RMER JACK IS BANKRUPT":GOTO2510
2455 IF BB+AL>0 THEN RE=0
2460 GOTO 130
```

```
2470 FS=BB+(AG*LC)+(TR*2000)+(TK*1000)+(
BR*1000)+(HD*5000)+(HV*5000)
2480 FOR I=1 TO 2500:NEXT:CLS
2490 PRINT"YOU HAVE FINALLY RETIRED AND
HAVE":PRINT"SOLD ALL OF YOUR ASSETS":PRI
NT"YOUR ASSETS ARE VALUED AT $";FS
2500 J=INT(FS/100000)
2510 IF J=0 THEN PRINT"IT'S THE POORHOUS
E FOR YOU":PRINT"MY FRIEND"
2520 PLAY"T200V30O1AAAAACCCCCFFFFFCCCCCC
CCCC"
2530 IF J=1 THEN PRINT"YOU ARE A POOR ST
RUGGLING FARMER":PRINT"THERE ISNT MUCH T
O SHOW FOR ALL":PRINT"THOSE YEARS EXCEPT
 THE PENSION"
2540 IF J=2 THEN PRINT"YOU ARE AN AVERAG
E FARMER":PRINT"RETIREMENT WILL NOT BE":
PRINT"TOO UNCOMFORTABLE"
2550 IF J=3 THEN PRINT"YOU ARE A GOOD FA
RMER!!!":PRINT"YOU WILL LIVE IN COMFORT
DURING":PRINT"YOUR RETIREMENT"
2560 IF J=4 THEN PRINT"YOU ARE AN EXCELL
ENT FARMER!!!":PRINT"YOU WONT LACK MANY
LUXURIES":PRINT"WHILST YOU ARE RETIRED"
2570 IF J>4 THEN PRINT@134,"you are a su
per farmer":PRINT TAB(2)"congratulations
 on mastering":PRINT TAB(9)"this bby gam
e."
2575 FORI=1TO3000:NEXT:GOSUB1010
2580 FOR C=0 TO 8:CLSC:FOR DD=1 TO 100:N
EXT:NEXT
2590 FOR P=1 TO 4:PLAY"T150V3004GGGGGCCC
CCAAAAAFFFFFP2":NEXT
```

```
2600 CLS:PRINT"PLAY AGAIN (Y/N)";:INPUTX
$
2610 IF X$="Y" THEN RUN ELSE GOTO 2620
2620 CLS RND(8):PRINT@192, TAB(12)"GOODB
YE"
2630 GOTO 2630
```

# Avalanche

In Avalanche you are the mayor of an Alpine village which suffers from very serious avalanches. Your job is to manage the village's finances so that you are building sufficient avalanche shelters for your people without bankrupting the town.

The money sitting in the town treasury attracts a healthy interest rate each year. The aim is to survive for twenty years in office. Each year the risk of an avalanche increases, so you must decide whether you are going to increase your spending on shelters or will take the risk and leave the money in the bank to attract the best possible interest for future use.

If you decide incorrectly and an avalanche results in a serious loss of life, you will be removed from office.

```
10 REM AVALANCHE
20 CLS 3
30 GOSUB 420
40 ST=1
50 M=RND(5000)+9087
60 P=RND(1000)+2278
70 S=0:D=0:N=0
80 FOR Y=1 TO 20
90 CLS 3
100  IF AL=1 THEN AL=0:GOTO 130
110  IF Y=20 THEN 310
120 PRINT TAB(9) "NO AVALANCHES"
130 FOR XX=1 TO 2
140 PLAY "O2T4L25EEEEEEEEEEFFFFFFFFFFGGG
GGGGGGG"
```

```
150 NEXT XX:CLS
160 PRINT "year";Y
170 PRINT "THE CHANCE OF AN AVALANCHE IS
  ";20-Y;"TO 1"
180 PRINT:PRINT "THE VILLAGE HAS";P;"PEO
PLE"
190 IF S<>0 THEN PRINT:PRINT "YOU HAVE";
S;"SQ. YDS OF":PRINT " SHELTER"
200 PRINT:PRINT "THE VILLAGE TREASURY HO
LDS","$";M
210 IF D<>0 THEN PRINT:PRINT "THERE ARE"
;D;"VILLAGERS DEAD"
220 IF N<>0 THEN PRINT:PRINT "THERE HAVE
 BEEN";N;"AVALANCHES"
230 PRINT:INPUT "HOW MUCH WILL YOU SPEND
 ON      SHELTER";A
240 IF M-A<0 OR S+A<0 THEN 230
250 M=M-A
260 S=S+INT(2.7*A)
270 M=M+INT(M/10+.5)
280 IF RND(20-Y)=1 THEN GOSUB 390
290 P=INT(P+.05*P)
300 NEXT Y
310 PRINT:PRINT "WELL DONE MAYOR GLUGENH
EIMER"
320 PRINT:PRINT "YOU MANAGED TO SURVIVE
20 YEARS"
330 PRINT "IN OFFICE,AND END YOUR TERM W
ITH:"
340 PRINT:PRINT TAB(4) "$";M;"IN THE TRE
ASURY"
350 PRINT:PRINT TAB(4);P;"PEOPLE SURVIVI
NG"
```

```
360 PRINT:PRINT TAB(4) ;S;"CUBIC YARDS O
F              SHELTER"
370 PRINT:PRINT "YOUR MAYORAL RATING IS"
;10*M+20*P+P+S-12*D
380 END
390 REM ***AVALANCHE***
400 AL=1
410 FOR XX=1 TO 10:SOUND 250,1:NEXT XX
420 FOR J=0 TO 56
430 PRINT TAB(J/3); "**avalanche**"
440 FOR E=1 TO 220-10*J:NEXT E
450 SOUND 100,1
460 NEXT J
470 IF ST<1 THEN RETURN
480 Q=INT(S/10+.5)
490 IF Q>P THEN Q=P
500 PRINT:PRINT Q;"PEOPLE ARE SAFE..."
510 X=P-Q
520 IF X<1 THEN X=0
530 PRINT:PRINT "BUT";X;"PEOPLE WERE KIL
LED!!!"
540 D=D+X
550 IF D>P THEN GOTO 630
560 P=Q
570 S=S-INT(S/10+.5)
580 PRINT:PRINT "THERE ARE";S;"CUBIC YAR
DS OF":PRINT " SHELTER LEFT"
590 FOR E=1 TO 3000:NEXT E
600 IF X>P THEN 620
610 RETURN
620 PRINT:PRINT
630 PRINT "the loss of life was"
640 SOUND 50,2
```

```
650 PRINT "catastrophic...YOU HAVE"
660 SOUND 50,2
670 PRINT "BEEN FORCED TO STEP DOWN"
680 SOUND50,2
690 PRINT "FROM THE OFFICE OF MAYOR"
700 SOUND50,2
710 END
```

# Gambling Games

# Multipack Twenty-one

We will now shuffle the cards and play a few hands of Pontoon, or Twenty-one. This program plays a real game of Pontoon and it plays very well. You will find that you will go bankrupt much more often than you are able to break the bank.

You are given a total of fifty chips at the beginning of the game and you will be asked at the start of each hand how many chips you wish to bet. There is a house limit of five chips. Each time you 'buy' a card from the bank, it will cost you the number of chips you have nominated. The object of the game is to keep buying cards until the face value of the cards is twenty-one or as near to twenty-one as you dare go. Going over twenty-one automatically makes you lose. When you want to stop buying cards, just tell the computer that you wish to stand.

An ace can count as a one or as eleven. When you draw an ace the computer will ask you which value you wish to use. If you choose eleven and, at a later point in that game, your total goes over twenty-one, the computer will change the value of your ace back to a one.

When you have decided to stand, the computer will then take its turn at drawing cards. The player with the total closest to twenty-one will win.

The bank will also pay out for a total of five under twenty-one but you will lose if the game is a draw.

The game ends either when you have lost all your chips or when you have managed to break the bank.

```
10 REM MULTI PACK TWENTY-ONE
20 GOSUB 790
30 BB=50:NC=1
40 GOTO 1580
50 GOSUB420
60 GOSUB 880
70 GOSUB 540
80 GOSUB 340
90 GOTO 500
100 CLS
110 GOSUB 850
120 GOTO 210
130 CLS 0
140 IF CT<22 THEN GOTO 260
150 PRINT@ 192, TAB(9) "OVER 21!! BUST"
160 GOSUB 850
170 BB=BB-CB:IF BB<1 THEN GOTO 730
180 CT=0:ZU=0:U=0
190 CU=0:ZT=0:NC=1
200 GOTO 50
210 IF CS=11 THEN GOTO 230
220 GOTO 130
230 CT=CT-10
240 CS=0
250 GOTO 500
260 IF CT>21 THEN GOTO 110
270 INPUT C
280 IF C=2 THEN GOTO 1240
290 GOSUB 1690
300 GOSUB 880
310 CB=CB+IB
320 GOSUB 340
330 GOTO 500
```

```
340 U=U+6
350 POKE 1117+U,ZX
360 POKE 1118+U,ZV
370 ZU=ZU+1
380 IF ZU>1 THEN GOTO 410
390 GOSUB 880
400 GOTO 340
410 RETURN
420 CLS
430 PRINT TAB(7) "YOU HAVE";BB;"CHIPS"
440 PRINT TAB(5) "HOW MANY WILL YOU BET"
;
450 INPUT IB
460 IF IB>BB THEN 450
470 IF IB<1 OR IB>5 THEN PRINT "PICK A N
UMBER BETWEEN 1 AND 5":GOTO 450
480 CB=IB
490 RETURN
500 PRINT@ 192, "TOTAL=";CT;;"   YOU HAV
E";BB;"CHIPS"
510 PRINT "CHIPS BET=";CB
520 PRINT "WILL YOU...";"1=BUY","2=STAND
"
530 GOTO 260
540 CLS
550 PRINT TAB(5) "I AM DEALING THE CARDS
"
560 FOR CO=1 TO 5
570 POKE 1056+CO+CA+CD,239
580 IF CO=1 THEN SOUND 200,1
590 NEXT CO
600 CA=CA+6
610 IF CA=30 THEN CA=0:GOTO 630
```

```
620 GOTO 560
630 CD=CD+32
640 IF CD=160 THEN CD=0:CA=0:PRINT@ 0,""
:RETURN
650 GOTO 560
660 CLS 5
670 FOR AA=1 TO 8
680 PRINT
690 PRINT TAB(6) "you broke the bank!!"
700 PLAY "L15002T1 DDDDDDDDDDEEEEEEEEEF
FFFFFFFFF"
710 NEXT AA
720 GOTO 720
730 CLS 4
740 FOR DD=1 TO 25:NEXT DD
750 PRINT@ 160, TAB(9) "YOURE BANKRUPT"
760 FOR DD=1 TO 250:NEXT DD
770 PRINT@ 288, TAB(12) "YOU LOSE"
780 GOTO 780
790 CLS 8
800 FOR JJ=1 TO 16
810 PRINT TAB(INT(JJ/1.4)) "muit; pack t
wenty-one"
820 SOUND 200,2
830 NEXT JJ
840 RETURN
850 PLAY "T5001AAAFFFGGGDDDEEEFFFAAA"
860 FOR DD=1 TO 1000:NEXT DD
870 RETURN
880 ZY=RND(13)
890 IF ZY=1 THEN ZX=65
900 IF ZY=2 THEN ZX=114
910 IF ZY=3 THEN ZX=115
```

```
920 IF ZY=4 THEN ZX=116
930 IF ZY=5 THEN ZX=117
940 IF ZY=6 THEN ZX=118
950 IF ZY=7 THEN ZX=119
960 IF ZY=8 THEN ZX=120
970 IF ZY=9 THEN ZX=121
980 IF ZY=10 THEN ZX=113
990 IF ZY=11 THEN ZX=74
1000 IF ZY=12 THEN ZX=81
1010 IF ZY=13 THEN ZX=75
1020 CV=ZY
1030 IF ZY>10 THEN CV=10
1040 GOSUB 1090
1050 IF ZY<>10 THEN ZV=239
1060 IF ZY=10 THEN ZV=112
1070 CT=CT+CV:NC=NC+1
1080 RETURN
1090 IF CV=1 THEN GOTO 1110
1100 RETURN
1110 GOSUB 1690
1120 IF ZT=1 THEN GOTO 1210
1130 PRINT@ 320, "THIS CARD IS AN ACE"
1140 PRINT "THIS IS CARD NUMBER";NC
1150 PRINT "CARD TOTAL BEFORE THIS HAND=
";CT
1160 PRINT "WILL THIS CARD BE 1 OR 11";
1170 INPUT A$:IF A$="" THEN 1170
1180 IF A$="1" THEN CV=1:GOSUB 1690:RETU
RN
1190 IF A$="11" THEN CV=11:CS=11:GOSUB 1
690:RETURN
1200 GOTO 1130
1210 IF CT+11>21 THEN CV=1:RETURN
```

```
1220 CV=11:CS=11
1230 RETURN
1240 PS=CT:ZT=1:PC=NC
1250 ZU=0:NC=1:CT=0
1260 CS=0:U=0
1270 GOTO 1580
1280 GOSUB 540
1290 GOSUB 880
1300 GOSUB 340
1310 GOTO 1320
1320 PRINT@ 192, "YOU HAVE BET";CB
1330 PRINT "YOUR HAND=";PS
1340 PRINT "MY TOTAL SO FAR=";CT
1350 PRINT "I WILL....."
1360 GOSUB 850
1370 IF CT>21 THEN GOTO 1400
1380 IF CT>16 THEN GOTO 1540
1390 PRINT "BUY A CARD":GOTO 1290
1400 IF CS<>11,THEN GOTO 1420
1410 CS=0:CT=CT-10:GOTO 1320
1420 CLS 0
1430 PRINT@ 192, TAB(11) "I BUSTED!!"
1440 GOSUB 850
1450 PRINT@ 320, TAB(5) "HOW DID YOU FLU
KE THAT?"
1460 GOSUB 850
1470 CLS
1480 IF PS=21 AND PC=3 THEN BB=BB+(CB*2)
:GOTO 1510
1490 IF PC=6 THEN BB=BB+(CB*2):GOTO 1510
1500 BB=BB+CB
1510 ZT=0
1520 CT=0:NC=1:ZU=0
```

```
1530 U=0:GOTO 1580
1540 CC=CT:CN=NC
1550 ZT=0:ZU=0:CS=0
1560 NC=1:U=0:CT=0
1570 GOTO 1740
1580 IF ZT=1 THEN GOTO 1650
1590 IF BB<1 THEN GOTO 730
1600 IF BB>99 THEN GOTO 660
1610 CLS
1620 PRINT "ITS YOUR TURN"
1630 GOSUB 850
1640 ZT=0:GOTO 50
1650 CLS
1660 PRINT "ITS MY TURN"
1670 GOSUB 850
1680 GOTO 1280
1690 PRINT@ 320, ""
1700 PRINT
1710 PRINT
1720 PRINT
1730 RETURN
1740 PRINT "STAND      ":REM 5 SPACES
1750 GOSUB 850
1760 CLS
1770 IF CC=21 AND CN=3 THEN GOTO 1840
1780 IF PS=21 AND PC=3 THEN GOTO 1860
1790 IF CN=6 THEN GOTO 1880
1800 IF PC=6 THEN GOTO 1900
1810 IF CC=>PS THEN GOTO 1920
1820 PRINT TAB(12) "YOU WON!":SOUND 200,
3
1830 BB=BB+CB:GOTO1950
```

```
1840 PRINT TAB(7) "I HAVE A PONTOON!":SO
UND 100,5
1850 BB=BB-(CB*2):GOTO 1950
1860 PRINT TAB(5) "YOU HAVE A PONTOON!":
SOUND 200,3
1870 BB=BB+(CB*2):GOTO 1950
1880 PRINT TAB(7) "I HAVE FIVE UNDER!":S
OUND 100,5
1890 BB=BB-(CB*2):GOTO 1950
1900 PRINT TAB(5) "YOU HAVE FIVE UNDER":
SOUND 200,3:GOTO 1950
1910 BB=BB+CB
1920 IF CC=CS THEN PRINT TAB(6) "I BEAT
YOU ON A DRAW":SOUND100,5:GOTO 1940
1930 PRINT TAB(6) "I BEAT YOU";CC;"TO";P
S:SOUND 100,5
1940 BB=BB-CB
1950 GOSUB 850:CC=0:PC=0:PS=0:CS=0
1960 GOTO 1580
```

# Dragon Dice

Dragon Dice is actually three games in one. Your clever computer, always ready to humble a mere human, has put together three dice games to form one big program.

When the program is first RUN, and after the end of each game, the program uses lines 100 to 140 to pick one of the games at random, then, after a brief delay, begin playing the game. So the poor old human doesn't even get to choose which game he is going to be beaten at.

The three games are dice versions of the card games Chemin de Fer, Shark and Seven times Seven. We will now talk about each game in turn so that you understand the rules.

## Chemin de Fer

In this game, you and your computer take it in turns to roll five dice, adding the pips as you go. You are aiming to get a higher total than the computer.

However, this game is not just a simple 'adding the pips' one. Any die which falls showing a 5 or a 2 must be thrown again, and your total is just the final digit of the answer (that is, a total of 27 is reduced to 7 and a total of 13 is counted as 3. If you are unlucky enough to throw a 10, it is counted as zero).

There are three special totals – 7 (a Natural), 8 (La Petite) and 9 (La Grande). The names are only bestowed when the relevant total is achieved on the first throw (i.e., the dice which came up 5 or 2 have not been, and in this case are not, rethrown). You (and the computer) always stand on a 7, 8 or 9 thrown with the first toss of the five dice.

Line 170 clears the screen and the game counter (GA), and when the variables for your score (HS) and the computer's

score (CS) have been initialized to zero, the program goes to line 490 where the game proper begins.

As in many other programs in this book, there is a delay loop routine at the end of the listing which is called up a number of times throughout the game to improve the speed at which the game advances, and to give you a chance to read what is printed on the screen before the program races on to something else. In this game there are two loops, one longer than the other.

Line 490 calls a delay and then clears the screen before incrementing the counter GA in the next line. You are told which game it is by line 510 and then the computer announces that, because it is taking the role of banker, who always goes first, it will have the first roll of the dice. The 'roll the dice' routine from line 210 is then triggered. You'll see that the loop counter G is used (line 220) to set the variable A to a randomly chosen value between 1 and 6 each time through the loop. If the die comes up 2 or 5 (line 230) then it is reset to zero before the value is printed by the following line.

The running total is clocked up on variable D, and then the leading digits are stripped from this (i.e., 27 is first cut down to 17 and then to 7) by line 300. The changing total is printed on the screen each time. After a delay the computer prints up "Total on the first roll is . . ." and then checks (lines 330, 340 and 350) to see if one of the special rolls has happened.

If so, a message to that effect is printed. If not, the computer gets to line 360 where it can see if any dice have to be rolled again (C is incremented by one each time a 2 or a 5 is rolled, see line 230). If there is none which has to be rerolled, the computer moves on to line 570 to print out "So my final total is . . .". If, however, C is not equal to zero, then the computer runs through another loop (390 to 460) to throw those dice again. Once again, any die coming up 2 or 5 is discarded (line 440).

Once the computer has had its roll, it is your turn to play. Your involvement is pretty small at this stage. Once you've pressed ENTER (lines 600 and 610) the computer takes over for you, using the same routine near the beginning of the game

to roll the dice for you, and incrementing your total as it does so.

Once both of you have had your turns, the computer then decides if it has beaten you, or you have defeated it, or whether the two of you have drawn ("Standoff"). The aim of Chemin de Fer is to win the majority of nine scoring games (and a Standoff does not count as a scoring game).

"The totals so far in Chemin de Fer are . . ." you are told by the computer in lines 850 and 860 and – if nine scoring games have not been played – the computer makes some comment on the game ("Looks like I'm in front") before returning to the main program.

Once a total of nine scoring games has been detected (by line 900) then the routine from line 990 comes into action. "Well, sir, we seem to have come to the end of the game . . .". The computer then determines who has been the winner.

After the game is over the program returns to line 100 where the computer picks the next game it wants to play.

## Shark

Shark demands some cool thinking under pressure. You and your computer take it in turns to throw a pair of dice. You add the total of the pips, and in turn add this to your score.

You can roll the dice as many times as you like, but if you roll a 7, you automatically lose. Therefore, as you can see, the program demands that you make decisions based on whether you should be careful and perhaps lose the round by not rolling a high enough total, or whether you should be greedy and go for the higher score and risk losing by rolling a 7.

The time you take to press ENTER to roll the dice has some effect on the score you achieve. You can see, in lines 1520 to 1550, that the variable N (set to zero in line 1520) is incremented by one each time around the 1540/1550 loop, until you press either the "Q" key or the "R" key.

If you press "R" to indicate that you wish to roll the dice,

the computer goes to the subroutine from line 1780 where the screen is cleared (1790) and then two dice (X and Y) are rolled. The Q loop is covered five times, gradually slowing down as it is traversed (by the inner loop, in line 1850), until finally – with the line 1890 – the results of the dice rolls are printed. The variable Z is set equal to the tally of the two rolls (in line 1900) before the computer goes to line 2150 for the delay and to print a line across the screen, and then returns to the start of the program to line 1590, the one after that which sent it to the 'roll the dice' routine. Here the computer checks that the total is not 7 (line 1590) and if it is not, then adds the result of the latest roll to your tally.

If, however, you signal that you wish to quit, the action goes to the routine from line 1630. After printing STAND BY, the computer goes to the delay loop, and line print routine, and then uses the same routine as the human did for rolling the dice, and – on returning from the subroutine – checking that the total wasn't 7, and if not, adding the new total to the score. The computer has a very simple means of deciding if it will roll the dice again: if it has less than the human it takes the risk.

You'll find that it is relatively easy to program games like this on your computer. Once you've worked out the 'mechanical routines' which do such things as roll the dice and increment the score for the human player, it is not very difficult to work out a routine to enable the computer to use the same mechanical routines. Most dice games do not demand much 'intelligence' and their strategy can often be reduced to a few IF/THEN statements. Read books on dice games – such as the excellent *Dice Games New and Old* by William E. Tredd (The Oleander Press, New York, 1981) – to get ideas for games to turn into programs, and for simple ideas on how to play the game as well as possible. It is these ideas which you should find relatively easy to turn into simple 'intelligent algorithms' to enable your computer to play well against you.

And if, as in this case, you stipulate that the human must go first, the computer knows exactly what target it is aiming

at, and therefore starts a round with a considerable advantage, which helps overcome the machine's inherent stupidity.

After you and the computer finish a game of Shark, the program goes back to line 100 so that the computer can pick another game to play.

## Seven times Seven

Seven times Seven is a computer adaptation of the dice game usually called 'Under and Over Seven'. In this game, you bet on the likelihood of the total of a pair of dice landing so that the total is:

- less than seven
- exactly equal to seven
- greater than seven

This is a game which, at first glance, seems to offer pretty good odds to the player. However, if you played the game forever with a pair of perfect dice, you'd find that your losses would outweigh your wins by about 17 per cent.

The program structure is not hard to follow. After clearing the screen in line 2340, the computer sets the starting stake (called M for money) to 30 (that is $30). Variable GG is also set (GG is used to count the number of turns). The game finishes after seven turns. Line 2360 sends the action to the subroutine from line 2860 where your current money is printed up ("You now have $30") before returning to line 2380 where the delay subroutine (lines 2890, 2900 and 2910) is called.

Lines 2390 to 2460 ask you to place your bet, giving you the key for entering it. Lines 2470 and 2480 read the keyboard using INKEY$, rejecting (line 2480) any input which is not A, B or C.

Having received a valid choice, the odds are printed up by lines 2500 to 2530, and then line 2560 asks you to enter the amount of your bet. Of course, you can't bet more than you

have (not in this game, anyway) so line 2590 checks your bet (A) against your money (M) and if you haven't got enough prints up: "You haven't got that much!"

The next section rolls the dice, assigning random values between one and six to variables B and C, and adding them to produce total D in line 2670. You are then told of the total by line 2700.

The outcome is determined by the routine from lines 2710 to 2750 where a loss is first assumed (line 2710, with variable W, for win, assigned to the negative of the amount you wagered). This is changed, if needed, into the correct amount for a win if one has, in fact, taken place. You can see that you get paid four times your bet for correctly specifying that the dice will come up seven (line 2720), and even money for either over or under seven (lines 2730 and 2740).

Lines 2770 and 2780 tell you about your win or loss, and then line 2820 checks that you still have some money in hand. If you have, the computer goes back for another round of the game. If not, you are dismissed with the bankrupt message.

Line 2840 tests GG (incremented in line 2370) to see if it equals seven. If so, the game ends and you are told how well or badly you performed. Then the program returns to line 100 for the next game to be picked.

```
10 REM DRAGON DICE
20 CLS 0
30 PRINT@ 192, TAB(6) "### DRAGON DICE #
##"
40 FOR TT=1 TO 3
50 PLAY "T250V3004ADGADG03ADG04ADGADG"
60 FOR DD=1 TO 50:NEXT DD
70 NEXT TT
80 FOR DD=1 TO 500:NEXT DD
90 CLS
```

```
100 PRINT @ 192, TAB(2) "I WILL NOW SELE
CT THE GAME WE"
110 PRINT TAB(6) "ARE GOING TO PLAY"
120 FOR DD=1 TO 1000:NEXT DD
130 X=RND(3)
140 ON X GOTO 150,1400,2320
150 REM CHEMIN DE  FER
160 GOSUB 1310
170 CLS
180 GA=0
190 B1=0:P1=0
200 GOTO 490
210 D=0:C=0
220 FOR G=1 TO 5:A=RND(6)
230 IF A=2 OR A=5 THEN C=C+1
240 PRINT A;
250 SOUND 75,1
260 GOSUB 1120
270 IF A=2 OR A=5 THEN A=0
280 D=D+A:NEXT G
290 PRINT:PRINT D;
300 IF D>9 THEN D=D-10:PRINT:PRINTD:GOTO
 300
310 GOSUB 1100
320 PRINT:PRINT "TOTAL ON THE FIRST ROLL
 IS";D
330 IF D=9 THEN PRINT "AND THAT'S LA GRA
NDE...":RETURN
340 IF D=8 THEN PRINT "AND THAT'S LA PET
ITE...":RETURN
350 IF D=7 THEN PRINT "AND THAT'S A NATU
RAL...":RETURN
360 IF C=0 THEN RETURN
```

```
370 GOSUB 1120
380 PRINT:PRINT C;"MUST BE ROLLED AGAIN"
390 FOR A=1 TO C
400 GOSUB 1120
410 E=RND(6)
420 PRINT E;
430 SOUND 75,1
440 IF E=2 OR E=5 THEN E=0
450 D=D+E
460 NEXT A
470 IF D>9 THEN D=D-10:PRINT D;:GOTO 470
480 RETURN
490 GOSUB 1120:CLS
500 GA=GA+1
510 PRINT:PRINT TAB(5) ">>> THIS IS GAME
";GA;"<<<":PRINT
520 PRINT "**********************************
***"
530 PRINT "NOW I WILL ROLL AS BANKER..."
540 PRINT "**********************************
***"
550 GOSUB 210
560 GOSUB 1100
570 PRINT "SO MY FINAL TOTAL IS";D
580 GOSUB 1100
590 PRINT "**********************************
***"
600 PRINT "PRESS ENTER TO ROLL YOUR DICE
";:INPUT A$
610 PRINT "**********************************
***"
620 J=D
630 GOSUB 1120
```

```
640 GOSUB 210
650 PRINT:PRINT "SO YOUR FINAL TOTAL IS"
;D
660 PRINT:PRINT
670 GOSUB 1120
680 PRINT "COMPUTER","HUMAN"
690 GOSUB 1100
700 PRINT J,D
710 GOSUB 1120:GOSUB 1120:PRINT
720 PRINT "$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$"
730 PRINT TAB(3);
740 IF J=D THEN PRINT "    THAT'S A STAN
DOFF":GOTO 780
750 IF J>D THEN PRINT "I'M THE";:B1=B1+1
760 IF D>J THEN PRINT "YOUR'E THE";:P1=P
1+1
770 PRINT " WINNER THAT TIME"
780 PRINT:PRINT "$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$"
790 PRINT:PRINT
800 FOR TT=1 TO 3
810 PLAY "T100V30O3AGAGDFDFO2AFGF"
820 FOR DD=1 TO 10:NEXT DD
830 NEXT TT
840 GOSUB 1100
850 PRINT "THE TOTALS SO FAR IN"
860 PRINT "CHEMIN DE FER ARE"
870 PRINT B1;"FOR ME, AND";P1;"FOR YOU"
880 PRINT:PRINT
890 GOSUB 1120
900 IF B1+P1=9 THEN 990
910 IF B1>P1 THEN PRINT "LOOKS LIKE I'M
IN FRONT"
```

```
920 IF P1>B1 THEN PRINT " AND YOU SEEM T
O HAVE THE EDGE!"
930 GOSUB 1100
940 FOR TT=1 TO 5
950 PLAY "T250V30O3AFGAFGAFGO2AFGFGFG"
960 FOR DD=1 TO 20:NEXT DD
970 NEXT TT
980 GOTO 490
990 REM END OF GAME
1000 PRINT:PRINT
1010 FOR TT=1 TO 5
1020 PLAY "T250V30O3AFGO4AFO3AFGD2AFG"
1030 FOR DD=1 TO 10:NEXT DD
1040 NEXT TT
1050 PRINT "WELL, SIR, WE SEEM TO HAVE C
OME"
1060 PRINT "TO THE END OF THE GAME...WI
H A"
1070 PRINT "TOTAL OF NINE SCORING ROUNDS
..."
1080 GOSUB 1100:GOSUB 1100
1090 GOTO 1140
1100 FOR DD=1 TO 1000:NEXT DD
1110 RETURN
1120 FOR DD=1 TO 500:NEXT DD
1130 RETURN
1140 CLS
1150 PRINT:PRINT:PRINT:PRINT
1160 IF P1>B1 THEN PRINT "AND FOR ONCE I
T IS YOU WHO":PRINT "BEATEN ME!!":GOSUB
1260:GOTO 1180
```

```
1170 IF B1>P1 THEN PRINT "AND ONCE AGAIN
 THE MIGHTY DRAGON":PRINT TAB(14) "WINS!
!":GOSUB 1260:GOTO 1180
1180 FOR DD=1 TO 1500:NEXT DD
1190 PRINT:PRINT:PRINT:PRINT "  THANK YO
U FOR THE GAME, HUMAN"
1200 PRINT:PRINT "WE WILL PLAY AGAIN SOM
ETIME SOON"
1210 FOR SS=100 TO 200 STEP 5:SOUND SS,1
:NEXT SS
1220 FOR SS=205 TO 95 STEP -5:SOUND SS,1
:NEXT SS
1230 FOR DD=1 TO 3000:NEXT DD
1240 CLS
1250 GOTO 100
1260 FOR TT=1 TO 5
1270 PLAY "T250V30O1ACDFGO2GFDECO3ADFGED
O4AFAFAO3ADFGFDO2AFAFAFO2AFAFGGGG"
1280 FOR DD=1 TO 30:NEXT DD
1290 NEXT TT
1300 RETURN
1310 CLS 4
1320 FOR TT=1 TO 5
1330 PRINT@ 192, " "
1340 FOR DD=1 TO 100:NEXT DD
1350 PRINT@ 192, TAB(8) "CHEMIN DE FER"
1360 PLAY "T250V30 O3AFGAFGGFAGFA"
1370 FOR DD=1 TO 100:NEXT DD
1380 NEXT TT
1390 RETURN
1400 REM SHARK
1410 GOSUB 2220
1420 CLS
```

```
1430 H=0:CZ=0
1440 HS=0:CS=0
1450 PRINT "PLEASE STAND BY....."
1460 GOSUB 2150
1470 CLS:PRINT:PRINT:PRINT
1480 PRINT "YOUR TOTAL IS";HS:PRINT
1490 GOSUB 2160
1500 PRINT "PRESS 'R' TO ROLL"
1510 PRINT "PRESS 'Q' TO QUIT"
1520 N=0
1530 W$=INKEY$
1540 N=N+1
1550 IF W$<>"Q" AND W$<>"R" THEN GOTO 15
30
1560 SOUND 125,2
1570 IF W$="Q" THEN 1630
1580 GOSUB 1780
1590 IF Z=7 THEN 1750
1600 HS=HS+Z
1610 PRINT:PRINT "YOUR TOTAL IS";HS
1620 GOTO 1460
1630 PRINT:PRINT:PRINT "STAND BY..."
1640 GOSUB 2150
1650 GOSUB 1780
1660 IF Z=7 THEN 1740
1670 CS=CS+Z
1680 PRINT:PRINT "MY TOTAL IS";CS
1690 PRINT "YOUR TOTAL IS";HS
1700 IF CS<HS THEN 1630
1710 IF CS=HS THEN PRINT "IT'S A DEAD HE
AT!!"
1720 IF CS>HS THEN 1760
1730 GOTO 1930
```

```
1740 PRINT:PRINT "YOU WIN!":H=H+1
1750 GOTO 1930
1760 PRINT:PRINT "I WIN!!!":CZ=CZ+1
1770 GOTO 1930
1780 REM ROLL THE DICE
1790 CLS
1800 FOR Q=1 TO 5
1810 X=RND(6)
1820 Y=RND(6)
1830 PRINT:PRINT "DIE ONE:";X;"   DIE TW
O:";Y
1840 PLAY "T250V30O2AFAFA"
1850 FOR P=1 TO 2*Q:NEXT P
1860 NEXT Q
1870 CLS
1880 GOSUB 2160
1890 PRINT "DIE ONE:";X;"   DIE TWO:";Y
1900 Z=X+Y
1910 GOSUB 2150
1920 RETURN
1930 PRINT:PRINT "THE SCORE IS:"
1940 PRINT, "YOU:";H
1950 PRINT, "ME :";CZ
1960 IF CZ+H=9 THEN 2000
1970 GOSUB 2160
1980 IF H>CZ THEN PRINT:PRINT "YOU ARE I
N THE LEAD"
1990 IF CZ>H THEN PRINT:PRINT "AND I AM
IN THE LEAD"
2000 GOSUB 2150
2010 CLS
2020 IF CZ+H=9 THEN 2040
2030 GOTO 1440
```

```
2040 CLS
2050 FOR TT=1 TO 3
2060 PLAY "T250V30O2AFGO3AFGO2AFG"
2070 FOR DD=1 TO 50:NEXT DD
2080 NEXT TT
2090 PRINT "WELL, THAT'S THE END OF THE
GAME"
2100 PRINT:PRINT:PRINT
2110 PRINT "YOUR FINAL SCORE WAS:";H
2120 PRINT "MY FINAL SCORE WAS  :";CZ
2130 IF H<CZ THEN PRINT:PRINT TAB(6) "I'
M THE WINNER!!!!":GOSUB 2170:CLS:GOTO 10
0
2140 PRINT TAB(6) "YOU'RE THE WINNER!!!"
:GOSUB 2170:CLS:GOTO 100
2150 FOR DD=1 TO 1000:NEXT DD:RETURN
2160 PRINT " ----------------------------
---":PRINT:RETURN
2170 FOR TT=1 TO 10
2180 PLAY "T100V30O3AFCDCDCDAFO4AFGAFGAF
G"
2190 FOR DD=1 TO 1000/TT:NEXT DD
2200 NEXT TT
2210 RETURN
2220 CLS RND(8)
2230 FOR GG=2 TO 510 STEP 5
2240 PRINT@ GG, "#";
2250 PLAY "T200V30O3ACEG"
2260 FOR DD=1 TO 10:NEXT DD
2270 NEXT GG
2280 CLS
2290 PRINT@ 192, TAB(14) "SHARK"
2300 FOR DD=1 TO 1500:NEXT DD
```

```
2310 RETURN
2320 REM SEVEN TIMES SEVEN
2330 GOSUB 3190
2340 CLS
2350 M=30:GG=1
2360 GOSUB 2860
2370 GG=GG+1
2380 GOSUB 2890
2390 PRINT "OK, HUMAN IT'S TIME TO PLACE
"
2400 PRINT "YOUR BET....."
2410 SOUND 200,1
2420 GOSUB 2830
2430 SOUND 200,1
2440 PRINT:PRINT "ENTER 'A' TO BET UNDER
 7"
2450 PRINT "       'B' TO BET ON SEVEN, O
R"
2460 PRINT "       'C' TO BET OVER 7"
2470 A$=INKEY$
2480 IF A$<>"A" AND A$<>"B" AND A$<>"C"
THEN 2470
2490 SOUND 50,2
2500 PRINT:PRINT "THE ODDS ARE:"
2510 PRINT "  A-PAYS EVEN MONEY"
2520 PRINT "  B-PAYS FOUR TO ONE"
2530 PRINT "  C-PAYS EVEN MONEY"
2540 GOSUB 2890
2550 SOUND 200,1
2560 PRINT:PRINT "HOW MUCH WOULD YOU LIK
E TO BET";:INPUT A
2570 SOUND 50,2
2580 GOSUB 2890
```

```
2590 IF A>M THEN PRINT "YOU HAVEN'T GOT
THAT MUCH":GOTO 2560
2600 B=RND(6)
2610 SOUND 175,3
2620 PRINT "DIE ONE CAME UP";B
2630 GOSUB 2890
2640 C=RND(6)
2650 SOUND 175,3
2660 PRINT "DIE TWO CAME UP";C
2670 D=B+C
2680 GOSUB 2890
2690 SOUND 75,2
2700 PRINT "SO THE TOTAL IS";D
2710 W=-A
2720 IF D=7 AND A$="B" THEN W=4*A
2730 IF D<7 AND A$="A" THEN W=A
2740 IF D>7 AND A$="C" THEN W=A
2750 M=M+W
2760 GOSUB 2830
2770 IF W>0 THEN PRINT "YOU'VE JUST WON
$";W
2780 IF W<0 THEN PRINT "AND SO YOU LOSE
$";(W*-1)
2790 GOSUB 2890
2800 GOSUB 2860
2810 GOSUB 2890
2820 IF M<1 THEN PRINT "YOU ARE FLAT BRO
KE":GOTO 2920
2830 CLS
2840 IF GG>7 THEN GG=7:GOTO 3030
2850 GOTO 2360
2860 PRINT "YOU NOW HAVE $";M
2870 PRINT "THIS IS TURN";GG
```

```
2880 RETURN
2890 FOR DD=1 TO 1000:NEXT DD
2900 PRINT:PRINT
2910 RETURN
2920 FOR HH=1 TO 5:CLS 0
2930 FOR TT=1 TO 5
2940 PLAY "T250V30O1ADGBCEO2AFGO1AFGAFG"
2950 FOR DD=1 TO 20:NEXT DD
2960 NEXT TT
2970 PRINT@ 192, TAB(3) "## BANKRUPT ##
BANKRUPT ##"
2980 FOR DD=1 TO 250:NEXT DD
2990 NEXT HH
3000 FOR DD=1 TO 1000:NEXT DD
3010 CLS
3020 GOTO 100
3030 FOR SS=1 TO 8
3040 CLS SS
3050 SOUND SS*2+10,3
3060 NEXT SS
3070 PRINT@ 96, TAB(8) "THE GAME IS OVER
"
3080 FOR DD=1 TO 500:NEXT DD
3090 PRINT@ 160, TAB(6) "YOU BEGAN WITH
$ 30"
3100 FOR DD=1 TO 500:NEXT DD
3110 PRINT@ 224, TAB(8) "YOU NOW HAVE $"
;M
3120 FOR TT=1 TO 5
3130 PLAY "T250V30O2ADGADGO1FGFGO2ADGADG
"
3140 FOR DD=1 TO 25:NEXT DD
3150 NEXT TT
```

```
3160 FOR DD=1 TO 500:NEXT DD
3170 CLS
3180 GOTO 100
3190 CLS 7
3200 FOR SS=10 TO 200 STEP 10:SOUND SS,1
:NEXT SS
3210 FOR SS=190 TO 20 STEP -10:SOUND SS,
1:NEXT SS
3220 FOR DD=1 TO 100:NEXT DD
3230 FOR TT=1 TO 5
3240 PRINT@ 192, TAB(4) "+++ +++++ +++++
 +++++ +++"
3250 FOR DD=1 TO 100:NEXT DD
3260 PRINT@ 192, TAB(4) "+++ SEVEN TIMES
 SEVEN +++"
3270 PLAY "T250V30O3ACFBAGDC"
3280 FOR DD=1 TO 100:NEXT DD
3290 NEXT TT
3300 FOR DD=1 TO 500:NEXT DD
3310 RETURN
```

# Malibu

The computer challenges you to roll the dice and try your hand at Malibu. You and the computer take it in turns to roll three dice each. Various combinations, and the total of the three dice, are worth a different number of points. For example, if the total of the pips showing is thirteen ('Lucky Joe') on the computer's dice, the computer gets six points, and the human loses six points.

You start the game with fifty points each and there are five rounds to the game. As I said, there are certain winning combinations (such as three sixes, called 'High Roller') and certain winning totals (such as nine or twelve, a 'Straight Road'). You can score more than once from a single roll, so if your total of nine was gained by rolling two fours and a one, you'd get the 'Straight Road' score of three points and the 'Two of a Kind' score of three points for the two fours.

The only exception to this is rolling three of a kind ('Triple Crown'). You can score most of the other possible combinations, although you cannot get 'Two of a Kind' and 'Triple Crown' for the same roll.

Now, this may all seem very confusing. It is much easier to play than it is to read about, because the computer rolls the dice and also works out the combinations. All you need to worry about is being beaten by the electronic whizzkid.

Here's a complete list of the winning throws (and remember that when one player gains a certain number of points, they are also subtracted from the other player's score. The total of both scores always equals 100).

| *Total* | *Name* | *Points* |
|---|---|---|
| 13 plus a pair | Sough | 10 |
| 6 or 15 | Easy Rider | 4 |
| 9 or 12 | Straight Road | 3 |
| All the same | Triple Crown | 5 |
| Two the same | Two of a Kind | 3 |
| 13 | Lucky Joe | 6 |
| 3 | Low and Mean | 7 |
| 18 | High Roller | 12 |

The only losing throw is seven ('Dreaded Seven') which costs you two points and adds two points to your opponent's score.

The program itself is quite easy to follow. Lines 240 to 440 assess the results of the rolls and print out the types of combinations obtained. The actual dice-rolling routine is in lines 880 to 990.

Lines 510 to 530 check to see if the game has ended and keep it going if both players are still in the game. Lines 540 to 630 call up the end routines depending on who has won the game.

All this is controlled by the main program which is contained in lines 40 to 230.

```
10 REM MALIBU
20 GOSUB 1230
30 GOSUB 730
40 REM MAIN PROGRAM ROUTINE
50 FOR T=1 TO 5
60 CLS
70 PRINT:PRINT:PRINT "THIS IS ROUND";T
80 SOUND 25,3
90 IF CS*HS<>0 THEN PRINT A$;":";HS;"  C
OMPUTER:";CS
```

```
100 GOSUB 870
110 PRINT:PRINT "FIRST I WILL ROLL THREE
 DICE FOR";
120 PRINT "MYSELF....STAND BY..."
130 PRINT
140 GOSUB 870
150 FOR Z=1 TO 3:C(Z)=0:H(Z)=0:NEXT Z
160 GOSUB 880
170 PRINT:PRINT "NOW IT'S TIME TO ROLL F
OR YOU"
180 PRINT
190 GOSUB 880
200 CLS
210 PRINT:PRINT "I ROLLED";C(1);C(2);C(3
)
220 PRINT:PRINT "YOU ROLLED";H(1);H(2);H
(3)
230 PRINT
240 REM ASSESS RESULT OF ROLLS
250 HT=H(1)+H(2)+H(3)
260 CT=C(1)+C(2)+C(3)
270 IF (C(1)=C(2) OR C(2)=C(3) OR C(1)=C
(3)) AND CT=13 THEN PRINT "THAT'S SOUGH
FOR ME!":CS=CS+10:HS=HS-10
280 IF (H(1)=H(2) OR H(2)=H(3) OR H(1)=H
(3)) AND HT=13 THEN PRINT "THAT'S SOUGH
FOR YOU!":HS=HS+10:CS=CS-10
290 IF HT=15 OR HT=6 THEN HS=HS+4:CS=CS-
4:PRINT "## EASY RIDER FOR YOU ##"
300 IF CT=15 OR CT=6 THEN CS=CS+4:HS=HS-
4:PRINT "## EASY RIDER FOR ME ##"
310 IF CT=9 OR CT=12 THEN CS=CS+3:HS=HS-
3:PRINT ">> I CRACK A STRAIGHT ROAD <<"
```

```
320 IF HT=9 OR HT=12 THEN HS=HS+3:CS=CS-
3:PRINT ">> YOU CRACK A STRAIGHT ROAD <<
"
330 IF C(1)=C(2) AND C(2)=C(3) THEN PRIN
T "A TRIPLE CROWN FOR ME!":CS=CS+5:HS=HS
-5:GOTO 350
340 IF C(1)=C(2) OR C(2)=C(3) OR C(1)=C(
3) THEN PRINT "$$ TWO OF A KIND FOR ME $
$":CS=CS+3:HS=HS-3
350 IF H(1)=H(2) AND H(2)=H(3) THEN PRIN
T "A TRIPLE CROWN FOR YOU!":HS=HS+5:CS=C
S-5:GOTO 370
360 IF H(1)=H(2) OR H(2)=H(3) OR H(1)=H(
3) THEN PRINT "$$ TWO OF A KIND FOR YOU
$$":HS=HS+3:CS=CS-3
370 IF HT=13 THEN PRINT "FOR YOU...LUCKY
 JOE!":HS=HS+6:CS=CS-6
380 IF CT=13 THEN PRINT "FOR ME...LUCKY
JOE!":CS=CS+6:HS=HS-6
390 IF HT=3 THEN PRINT "LOW AND MEAN...F
OR YOU!"*HS=HS+7:CS=CS-7
400 IF CT=3 THEN PRINT "LOW AND MEAN...F
OR ME!":CS=CS+7:HS=HS-7
410 IF HT=18 THEN PRINT "HIGH ROLLER FOR
 THE HUMAN..":HS=HS+12:CS=CS-12
420 IF CT=18 THEN PRINT "HIGH ROLLER FOR
 THE MACHINE...":CS=CS+12:HS=HS-12
430 IF HT=7 THEN PRINT "YOU TRIPPED THE
DREADED SEVEN":HS=HS-2:CS=CS+2
440 IF CT=7 THEN PRINT "I TRIPPED THE DR
EADED SEVEN":CS=CS-2:HS=HS+2
450 REM PRINT SCORES
460 PRINT:PRINT
```

```
470 IF T<5 THEN PRINT ">> AFTER ROUND";T
;"THE SCORES ARE:":PRINT ">> ";A$;":";HS
;"  COMPUTER:";CS
480 FOR SS=50 TO 180 STEP 5:SOUND SS,1:N
EXT SS
490 FOR SS=175 TO 55 STEP -5:SOUND SS,1:
NEXT SS
500 GOSUB 870
510 REM CHECK BOTH ARE STILL IN GAME
520 IF CS<1 OR HS<1 THEN T=5
530 NEXT T
540 REM RESULT OF FIVE ROUNDS
550 PRINT:PRINT "WELL, ";A$;", THAT'S TH
E END OF"
560 PRINT "OUR LITTLE GAME OF MALIBU....
."
570 PRINT:PRINT A$;"'S FINAL SCORE IS";H
S
580 PRINT "AND MINE IS";CS
590 GOSUB 870
600 IF CS>HS THEN GOSUB 1000
610 IF HS>CS THEN GOSUB 1080
620 IF HS=CS THEN GOSUB 1160
630 GOSUB 870:GOSUB 870
640 CLS RND(8):PRINT@ 192, TAB(1) "DO YO
U WANT ANOTHER GAME (Y/N)"
650 PRINT
660 F$=INKEY$
670 IF F$<>"Y" AND F$<>"N" THEN GOTO 660
680 IF F$="N" THEN PRINT@ 256, "OK THANK
S FOR THE GAME, GOODBYE":END
690 PRINT@ 256, "OK.....STANDBY....
........."
700 GOSUB 870
```

```
710 GOSUB 800
720 GOTO 50
730 REM INITIALISE
740 CLS
750 DIM H(3),C(3)
760 PRINT@ 192, TAB(2) "WELCOME TO THE G
AME OF MALIBU"
770 PRINT@ 256, TAB(2) "WHAT IS YOUR NAM
E";
780 INPUT A$
790 FOR DD=1 TO 750:NEXT DD
800 CLS:PRINT
810 HS=50:CS=50
820 PRINT@ 192, A$;", PRESS THE SPACE BA
R"
830 PRINT TAB(2) "WHEN YOU ARE READY TO
PLAY"
840 IF INKEY$="" THEN 840
850 RETURN
860 REM DELAY ROUTINE
870 FOR DD=1 TO 1500:NEXT DD:RETURN
880 REM DICE ROLL
890 FOR Z=1 TO 3
900 PRINT " ROLLING DIE";Z;
910 FOR DD=1 TO 750:NEXT DD
920 K=RND(6)
930 PRINT "WHICH CAME UP";K
940 PLAY "T250V3003DEC04AFGFA03GFADEC"
950 PRINT " ----------------------------
--"
960 IF C(Z)=0 THEN C(Z)=K ELSE H(Z)=K
970 NEXT Z
980 GOSUB 870
```

```
990 RETURN
1000 CLS RND(8)
1010 FOR TT=1 TO 5
1020 PRINT@ 192, TAB(3) "***** *********
****** *****
1030 PLAY "T250V30O3DEFCGGAO4EAFFGAO3DFA
FG"
1040 PRINT@ 192, TAB(3) "***** I AM THE
WINNER *****"
1050 FOR DD=1 TO 10:NEXT DD
1060 NEXT TT
1070 RETURN
1080 CLS RND(8)
1090 GG=64
1100 PRINT@ GG, TAB(4) "!! YOU ARE THE W
INNER !!"
1110 PLAY "T250V30O1GFAFGO2AFAGAO1GEDFCA
"
1120 FOR DD=1 TO 20
1130 GG=GG+64
1140 IF GG>416 THEN GG=0:RETURN
1150 GOTO 1100
1160 CLS RND(8)
1170 PLAY "T200L8AAAFFFGGGO3FFGGAAFFO2FG
EDDDEEECCC"
1180 PRINT@ 192, TAB(7) "### IT'S A DRAW
 ###"
1190 FOR DD=1 TO 500:NEXT DD
1200 TD=TD+1
1210 IF TD>5 THEN TD=0:RETURN
1220 GOTO 1160
1230 CLS 4
1240 FOR GG=192 TO 223 STEP 2
```

```
1250 PRINT@ GG, ">"
1260 PLAY "T250V30O2ACACGGFGFCA"
1270 NEXT GG
1280 PRINT TAB(13) "MALIBU"
1290 FOR DD=1 TO 100:NEXT DD
1300 FOR GG=256 TO 287 STEP 2
1310 PRINT@ GG, ">"
1320 PLAY "T250V30O2CDEAGFO3ACB"
1330 NEXT GG
1340 RETURN
```

# Brain Games

# Follow the Leader

Although the listing is not very long, you will find Follow the Leader a pretty hard act to follow.

The program generates a sequence of numbers which you must copy. First it will put one number on the screen (such as 2) which will then vanish after a short time. You must then type in the number. After a pause the first number will reappear, followed by a second number (such as 27). You then enter these two numbers and if you got it right, the computer will add a third number to the series. The process continues until the computer thinks you have had enough, or until you make a mistake.

If you survive the sequence the words YOU DID IT CHAMP!! appear on the screen. Just as you're breathing a sigh of relief at having survived the ordeal, the words STANDBY FOR A MORE DIFFICULT ONE confront you. The game begins again, but this time the length of time the numbers stay on the screen is shortened. The computer will put you to the test five times, cutting the time you get to see the numbers each game. Finally, if you survive the final round, you are congratulated on your great effort.

If, at any stage, you make a mistake, line 230 will tell you that you blew it and give you a final score.

To make the game easier or harder, adjust the value of T in line 30. T controls the length of time the numbers are shown on the screen. Don't forget to also alter line 300 so that the game runs for five turns. You may also alter line 320 so that the number subtracted from T after each game is smaller or larger. Take care to make the game neither ridiculously easy nor totally impossible.

```
10 REM FOLLOW THE LEADER
20 GOSUB 350
30 T=800
40 A=1000*RND(9999)+RND(9999)
50 A$=STR$(A)
60 A$="2"+RIGHT$(A$,7)
70 Z=1
80 IF MID$(A$,Z,1)=" " THEN 40
90 IF Z<8 THEN Z=Z+1:GOTO 80
100 Q=1
110 CLS
120 PRINT:PRINT
130 IF Q=8 THEN 250
140 FOR Z=1 TO Q
150 PRINT MID$(A$,Z,1);
160 NEXT Z
170 FOR L=1 TO T:NEXT L
180 CLS
190 PLAY "T250V3003EDC02AFG03BCDEF"
200 IF INKEY$<>"" THEN 200
210 PRINT:PRINT:INPUT B$
220 IF B$="" THEN 210
230 IF B$<>LEFT$(A$,Q) THEN PRINT TAB(6)
 "WHOOPS YOU BLEW IT":PRINT TAB(6) "YOUR
 SCORE WAS";37*(1100-T+10*Q):FOR TT=1 TO
 5:PLAY "T250V3004AFGAFG03GFAGFA":NEXT T
T:END
240 IF Q<8 THEN Q=Q+1:GOTO 110
250 PRINT:PRINT TAB(6) "YOU DID IT CHAMP
!!!"
260 FOR TT=1 TO 5
270 PLAY "T250V3001AFG02AFG03AFG04AFG"
280 FOR DD=1 TO 15:NEXT DD
290 NEXT TT
```

```
300 IF T=300 THEN PRINT:PRINT:PRINT TAB(
2) "YOU ARE THE ULTIMATE CHAMP!!!":FOR T
T=1 TO 10:PLAY "T250V3004AFGO3CDE04AFG":
NEXT TT:END
310 PRINT:PRINT:PRINT "STANDBY FOR A MOR
E DIFFICULT ONE"
320 T=T-100
330 FOR Z=1 TO 3000:NEXT Z
340 GOTO 40
350 A$="*** FOLLOW THE LEADER *** "
360 L=32
370 CLS 4
380 PRINT@ 160, " "
390 FOR I=1 TO L
400 IF (L-1)<1 THEN 440
410 PRINT@ 192, TAB(L-I) LEFT$(A$,I)
420 GOSUB 490
430 NEXT
440 FOR I=1 TO 31
450 PRINT@ 192, MID$(A$,I,L-I)
460 GOSUB 520
470 NEXT
480 RETURN
490 FOR DD=1 TO 25:NEXT DD
500 SOUND 100,1
510 RETURN
520 FOR DD=1 TO 30:NEXT DD
530 SOUND 100,1
540 RETURN
```

# Idaho Squares

Idaho Squares are very similar to magic squares in that the horizontal and vertical columns of numbers all add up to the same total. The square is made up of a three-by-three grid.

You simply enter any number which you think may make up a part of the grid and then the computer checks to see if your number is one or more of the missing numbers in the square. If your guess is correct, the relevant zero (or zeros) in the grid changes magically into the number you have entered. A tally is kept of the number of guesses you have taken and the number of correct numbers in the square.

Here is the listing for Idaho Squares. You can change the nines in lines 240 to 280 to lower or higher numbers to make solving the puzzle easier or harder.

After the listing of the program I have included a few sample runs of the game so that you may see what the screen display should look like.

```
10 REM IDAHO SQUARES
20 GOSUB 600
30 GOSUB 240:REM INITIALISE
40 J=J+1
50 CLS
60 PRINT:PRINT:PRINT:PRINT TAB(9)
70 M=0
80 FOR Z=1 TO 9
90 PRINT B(Z);" ";
100 IF 3*INT(Z/3)=Z THEN PRINT:PRINT:PRI
NT TAB(9)
110 IF B(Z)=A(Z) THEN M=M+1
```

```
120 NEXT Z
130 PRINT
140 IF M=9 THEN GOTO 480
150 PRINT:PRINT "THIS IS GUESS NUMBER";J
160 PRINT:PRINT "YOU HAVE";M;"CORRECT"
170 PRINT:PRINT "ENTER YOUR GUESS";:INPU
T X
180 FOR Z=1 TO 9
190 IF B(Z)=0 AND A(Z)=X THEN B(Z)=X
200 NEXT Z
210 GOTO 40
220 END
230 REM INITIALISE
240 DIM A(9),B(9)
250 A=RND(9)
260 B=RND(9)
270 C=RND(9)
280 D=RND(9)
290 IF A=B OR B=C OR A=C OR A=D OR B=D O
R C=D THEN 260
300 A(1)=A+B
310 A(2)=A-(B+C)
320 A(3)=A+C
330 A(4)=A-B+C
340 A(5)=A
350 A(6)=A+B-C
360 A(7)=A-C
370 A(8)=A+B+C
380 A(9)=A-B
390 FOR Q=1 TO 9
400 B(Q)=A(Q)
410 NEXT Q
420 B(A)=0
430 B(B)=0
```

```
440 B(C)=0
450 B(D)=0
460 J=0
470 RETURN
480 FOR DD=1 TO 1000:NEXT DD
490 CLS RND(8)
500 PRINT@ 192, TAB(6) "YOU'VE SOLVED IT
!!!"
510 FOR T=1 TO 5
520 PLAY "T250V30O3ADCFGO2GFAO3AGFCAG"
530 FOR DD=1 TO 10:NEXT DD
540 NEXT T
550 PRINT@ 256, TAB(6) "YOU TOOK";J;"GUE
SSES"
560 FOR SS=50 TO 150 STEP 5:SOUND SS,1:N
EXT SS
570 FOR SS=145 TO 55 STEP -5:SOUND SS,1:
NEXT SS
580 FOR DD=1 TO 2000:NEXT DD
590 RUN 20
600 CLS RND(8)
610 FOR GG=192 TO 222 STEP 2
620 PRINT@ GG, "#";
630 PLAY "T250V30O3AFGO2FFAAO3GGAA"
640 NEXT GG
650 PRINT@ 192, TAB(2) "*#####  IDAHO SQ
UARES  *#####"
660 FOR T=1 TO 5
670 PLAY "T100V30O3GAGA"
680 FOR DD=1 TO 10:NEXT DD
690 PLAY "T100V30O2AGAG"
700 FOR DD=1 TO 20:NEXT DD
710 NEXT T
720 RETURN
```

```
16  -7   0

3   0   11

0   21   0
```

```
THIS IS GUESS NUMBER 1

YOU HAVE 5 CORRECT

ENTER YOUR GUESS  5
```

```
16  -7   0

3   0   11

0   21   0
```

```
THIS IS GUESS NUMBER 2

YOU HAVE 5 CORRECT

ENTER YOUR GUESS  3
```

```
16  -7   0

3   0   11

0   21   0
```

```
THIS IS GUESS NUMBER 3

YOU HAVE 5 CORRECT

ENTER YOUR GUESS -3
```

```
16  -7   0

3   0   11

0   21   0
```

```
THIS IS GUESS NUMBER 4

YOU HAVE 5 CORRECT

ENTER YOUR GUESS  12
```

```
16  -7  12

3   0  11

0  21   0
```

```
THIS IS GUESS NUMBER 5

YOU HAVE 6 CORRECT

ENTER YOUR GUESS  7
```

```
16  -7  12

3   7  11

0  21   0
```

```
THIS IS GUESS NUMBER 6

YOU HAVE 7 CORRECT

ENTER YOUR GUESS  2
```

```
16  -7   12

3   7   11

2   21   0
```

```
THIS IS GUESS NUMBER 7

YOU HAVE 8 CORRECT

ENTER YOUR GUESS -2
```

```
16  -7   12

3   7   11

2   21  -2
```

# Pentominos

This is a two-player game in which the computer is used as an electronic game board.

For the information of the two players, the computer has two graphic displays. The first is the screen which shows the game board – an eight-by-eight checkerboard with red and yellow squares. Also displayed on this screen is the name of the player having the current turn and the numbers of the used pentominos. Each pentomino can only be used once.

The second screen displays the eleven pentominos used to play the game. A pentomino is a playing piece which is made up of five squares. Each pentomino is a different shape. To switch between these two screens, press the "B" key to see the board and the "P" key to see the display of pentominos. Pressing the "X" key at any time will end the game and, after a short delay, the program will be RUN again.

The object of the game is for the two players to take turns selecting a pentomino and then placing it on the playing board. The strategy is to use up as many spaces on the board as possible each time you place a pentomino so that your opponent can't find room to place his pentomino when it is his turn. The player who puts the last piece on the board is the winner. So the game goes on with first one then the other of the two players selecting a pentomino and placing it on the board until there is no room to place the next piece. When this stage is reached, the player who couldn't move loses the game. To escape from the program and start another game, press the "X" key.

To select a pentomino from the display, use the forward cursor key until the indicator arrow is under the pentomino you wish to use. Then press the RETURN key. Then, to place the pentomino on the board, use the four cursor keys to indicate the five squares you want that shape to cover. After

you have indicated each square, press the RETURN key to enter the square into the computer's memory. After you have chosen all five squares, the computer will then colour the shape blue.

The computer is used only as an electronic board for this game, so there is no routine to prevent you from cheating. You and your opponent will have to keep each other honest. However, the computer will not let you select a pentomino which has already been used.

```
10 REM PENTOMINOS
20 GOSUB 1390
30 CLS
40 DIM BL(6),XX(6),PQ(6),:M=3:N=1:DIM MR(
5),NR(5),P(11),Q(11)
50 GOSUB 230:REM DRAW PENTOMINOS
60 CLS:GOSUB 1220
70 CLS RND(8)
80 PRINT@ 96, "WHAT IS FIRST PLAYER'S NA
ME":INPUT A$
90 IF LEN(A$)>5 THEN A$=LEFT$(A$,5)
100 PRINT@ 192, "WHAT IS SECOND PLAYER'S
 NAME":INPUT B$
110 IF LEN(B$)>5 THEN B$=LEFT$(B$,5)
120 GOSUB 500:REM PRINT BOARD
130 IF RND(2)=1 THEN C$=A$ ELSE C$=B$
140 T=0:PRINT@ 24, C$;"'S":PLAY "T200V30
O3EDAFG":FOR I=1 TO 750:NEXT
150 GOSUB 680:REM SELECT PENTOMINO
160 GOSUB 1270:REM USED PENTOMINOS
170 IF PU=1 THEN PU=0:GOTO 150
180 GOSUB 860:REM DRAW PENTOMINO
190 IF C$=A$ THEN C$=B$:GOTO140
```

```
200 IF C$=B$ THEN C$=A$
210 GOTO 140
220 GOTO 220
230 PMODE 4,1
240 PCLS:GET(0,0)-(14,14),XX
250 LINE(0,0)-(14,14),PSET,BF
260 GET(0,0)-(14,14),BL
270 PUT(0,0)-(14,14),XX
280 FOR I=1 TO 55
290 READ A,B
300 PUT(A,B)-(A+14,B+14),BL
310 NEXT
320 DATA 8,32,24,16,24,32,24,48,40,32
330 DATA 72,16,72,32,88,32,88,48,104,32
340 DATA 136,32,136,48,152,48,168,32,168
,48
350 DATA 200,32,216,32,216,48,232,48,232
,32
360 DATA 8,80,8,96,24,96,40,96,40,112
370 DATA 72,80,72,96,72,112,88,80,104,80
380 DATA 136,80,152,80,152,96,152,112,16
8,80
390 DATA 200,80,216,80,216,96,232,96,232
,112
400 DATA 16,160,32,144,32,160,48,160,64,
160,96,160,112,160,128,160,144,160,144,1
44,176,160,192,160,208,160,208,144,224,1
44
410 DRAW "BM32,66;NF6ND12G6"
420 GET(26,66)-(38,78),PO
430 PUT(24,64)-(39,79),XX
440 A=26:B=66
450 GOSUB 470
```

```
460 RETURN
470 FOR I=1 TO 5:PLAY "O4T100CDAAB"
480 NEXT
490 RETURN
500 CLS:PRINT@ 56, " TURN"
510 PRINT@ 88, " USED"
520 PRINT@ 120, " PENT-"
530 PRINT@ 152, " OMINO'S"
540 FOR X=0 TO 18 STEP 6
550 FOR Y=0 TO 384 STEP 128
560 FOR I=0 TO 2
570 PRINT@ X+I+Y, CHR$(191);
580 PRINT@ X+I+32+Y, CHR$(191);
590 PRINT@ X+I+64+Y, CHR$(159);
600 PRINT@ X+I+96+Y, CHR$(159);
610 PRINT@X+I+3+Y,CHR$(159);
620 PRINT@ X+I+35+Y, CHR$(159);
630 PRINT@ X+I+67+Y, CHR$(191);
640 PRINT@ X+I+99+Y, CHR$(191);
650 NEXT I,Y,X
660 SCREEN 1,1
670 RETURN
680 SCREEN 1,1
690 PUT(A,B)-(A+11,B+11),PO
700 GOSUB 780
710 IF FG=1 THEN FG=0:RETURN
720 PUT(A-0,B-2)-(A+14,B+12),XX,PSET
730 A=A+64+C
740 IF A=282 THEN B=B+64:A=26
750 IF B=194 THEN B=178:A=32:C=16
760 IF A+B=450 THEN A=26:B=66:C=0
770 GOTO 690
780 E$=INKEY$:IF E$=""THEN 780
```

```
790 IF E$=CHR$(9) THEN RETURN
800 IF E$=CHR$(13) THEN FG=1:RETURN
810 IF E$="B" THEN GOSUB 840
820 IF E$="X" THEN RUN
830 GOTO 780
840 SCREEN 0,0:G$=INKEY$:IF G$="" THEN 8
40
850 SCREEN 1,1:RETURN
860 PN=POINT(M-1,N):PRINT@ 24,C$;"'S"
870 RESET(M,N):FL=0
880 E$=INKEY$:IF E$="" THEN 880
890 IF E$="P" THEN GOSUB 1050
900 IF E$="X" THEN RUN
910 IF E$=CHR$(9) THEN M=M+6:MX=M-6:NX=N
:FL=1
920 IF M=51 THEN M=3:GOTO 1030
930 IF E$=CHR$(8) THEN M=M-6:MX=M+6:NX=N
:FL=1
940 IF M<3 THEN M=45:GOTO 1030
950 IF E$=CHR$(10) THEN N=N+4:NX=N-4:MX=
M:FL=1
960 IF N=33 THEN N=1:GOTO 1030
970 IF E$=CHR$(94) THEN N=N-4:NX=N+4:MX=
M:FL=1
980 IF N<1 THEN N=29:GOTO 1030
990 IF PN=1 OR PN=3 THEN IF E$=CHR$(13)
OR E$="P" THEN 860 ELSE 1030
1000 IF E$=CHR$(13) THEN GOSUB 1070
1010 IF T=5 THEN RETURN
1020 IF FL=0 THEN 880
1030 SET(MX,NX,PN)
1040 GOTO 860
```

```
1050 SCREEN 1,1:G$=INKEY$:IFG$="" THEN 1
050
1060 SCREEN 0,1:RETURN
1070 FOR I=-3 TO 2
1080 FOR J=-1 TO 2
1090 SET(M+I,N+J,1)
1100 NEXT J,I
1110 PN=1
1120 T=T+1
1130 MR(T)=M:NR(T)=N
1140 IF T=5 THEN GOSUB 1160
1150 RETURN
1160 FOR K=1 TO 5
1170 FOR I=-3 TO 2
1180 FOR J=-1 TO 2
1190 SET(MR(K)+I,NR(K)+J,3)
1200 NEXT J,I,K
1210 RETURN
1220 FOR I=1 TO 11
1230 READ P(I)
1240 NEXT
1250 DATA -7,57,121,185,-39,25,89,153,-5
7,23,103
1260 RETURN
1270 FOR I=1 TO 11
1280 IF P(I)=A-.5*B THEN IF Q(I)=1 THEN
GOSUB 1330 ELSE Q(I)=1
1290 NEXT
1300 FOR I=1 TO 11
1310 IF Q(I)=1 THEN PRINT@ 152+I*32, "NO
.";I;
1320 NEXT:RETURN
1330 PU=1
```

```
1340 FOR I=1 TO 10
1350 PLAY "01T200AAAACCCDDEDEDCCFFCFCFDE
"
1360 IF Z=0 THEN Z=1:SCREEN 1,0:GOTO 138
0
1370 IF Z=1 THEN Z=0:SCREEN 1,1
1380 NEXT:RETURN
1390 FOR GG=1 TO 3:FOR FF=1 TO 2
1400 IF FF=1 THEN CLS 3 ELSE CLS 4
1410 PRINT@ 160, " ***** ***** ***** ***
** ***** "
1420 PRINT@ 224, TAB(5) "##### PENTOMINO
S #####"
1430 PRINT@ 288, " ***** ***** ***** ***
** ***** "
1440 PLAY "T250V3001FGA02CDE03DEF04FDC03
DC02AGF01AFGA"
1450 FOR DD=1 TO 10:NEXT DD
1460 NEXT:NEXT
1470 RETURN
```

# Wizard of Wall Street

Your computer will now give you the opportunity to try your skill and judgement on the stock exchange. The computer will set up a register of four companies and display a balance sheet on the screen. The balance sheet will show the number of the company, the number of shares you hold in that company, the current price of the shares, the price you paid for the shares and the net profit or loss so far.

The headings for these columns are put on the screen by lines 140 and 150, so be sure to type these two lines exactly as they appear in the listing. The spaces between the words are important.

Underneath the balance sheet, the computer will display the amount of money you have in the bank and the bank interest for that turn. The bank interest will fluctuate throughout the game.

The computer will then give you the opportunity to buy shares in one of the companies. You will be asked which company you wish to invest in and, after you have answered this question, how many shares you wish to purchase. If you wish to sell shares, enter a negative number.

The shares you have nominated will be purchased by the computer, or sold if required. It will then go on to work out the fluctuations in the share market. A new balance sheet will then be printed showing the updated figures plus your current bank balance. You are now well on your way to making your first million.

If you attempt to purchase shares you can't pay for, the computer will declare you bankrupt and send you out of the game in disgrace. This condition is tested in line 290.

If you try to sell shares you do not own, the computer will accuse you of fraud and terminate the game. Line 310 is the line which keeps you honest.

Just as in real life, every now and then the stock market will collapse, wiping out your fortune. The collapse is worked out by line 380.

The companies and all the relevant information are kept in an array which is set up in line 80. A(1,A) holds the current price of the shares, A(2,A) records the shares that you own and A(3,A) holds the price you paid for the shares. The variable Z is the profit or loss figure (Z is calculated in line 170).

Line 100 sets the starting price of the shares, and they are altered each turn by line 330.

Lines 350 and 360 calculate the bank interest and add it to your bank balance.

```
10 REM STOCK EXCHANGE
20 CLS
30 GOSUB 870
40 CLS4
50 X=4
60 1=10
70 B=1000
80 DIM A(3,4)
90 FOR A=1 TO X
100 A(1,A)=A+RND(100)
110 A(3,A)=0
120 NEXT A
130 CLS
140 PRINT "CO.  YOU   PRICE  START  PROF
IT"
150 PRINT "NO.  HOLD   NOW   PRICE   /LO
SS"
160 FOR A=1 TO X
170 Z=A(1,A)-A(3,A)
```

```
180 PRINT A;TAB(X+1) A(2,A);TAB(11) A(1,
A);TAB(18) A(3,A);TAB(25) Z
190 NEXT A
200 PRINT
210 PRINT "BANK $";B:PRINT "INTEREST RAT
E IS";I;"%"
220 PRINT
230 INPUT "WHICH COMPANY(1-4)";A
240 PRINT
250 PRINT "HOW MANY SHARES"
260 INPUT "(A NEGATIVE MEANS SELL)";C
270 IF C>0 THEN A(3,A)=A(1,A)
280 B=B-C*A(1,A)
290 IF B<=0 THEN GOTO 490
300 A(2,A)=A(2,A)+C
310 IF A(2,A)<0 THEN GOTO 650
320 FOR A=1 TO X
330 A(1,A)=INT(ABS(A(1,A)+A*50*(RND(10)-
RND(10))/10))
340 NEXT A
350 I=RND(10)+5
360 B=INT(ABS(B+B*I/100))
370 IF B>999999 THEN GOTO 760
380 Y=RND(50):IF Y<>1 THEN GOTO 130
390 REM COLLAPSE
400 FOR DD=1 TO 1000:NEXT DD
410 CLS0
420 FOR T=1 TO 5
430 PLAY "T100V30O1GGGGGAAAAEEEDDDFFFFF"
440 FOR DD=1 TO 50:NEXT DD
450 NEXT T
460 PRINT@ 192, TAB(4) "THE MARKET HAS C
OLLAPSED"
```

```
470 PLAY "T5001V30ADFFGEAADEFFGGGGG"
480 GOTO 480
490 REM BANKRUPT
500 FOR DD=1 TO 1000:NEXT DD
510 CLS4
520 FOR SS=1 TO 200 STEP 10
530 SOUND SS,1
540 NEXT SS
550 FOR SS=200 TO 1 STEP -10
560 SOUND SS,1
570 NEXT SS
580 FOR TT=1 TO 5
590 CLS4
600 FOR DD=1 TO 100:NEXT DD
610 PRINT@ 192, TAB(8) "YOU ARE BANKRUPT
"
620 FOR DD=1 TO 100: NEXT DD
630 NEXT TT
640 GOTO 640
650 REM FRAUD
660 FOR DD=1 TO 1000:NEXT DD
670 CLS8
680 PLAY "T1500V30GGGGGAAAAAFFFFFGGGGO2
AAAFFFGGGGGAAAAA"
690 G=96
700 FOR LL=1 TO 9
710 PRINT@ G, TAB(12) "FRAUD"
720 FOR DD=1 TO 250:NEXT DD
730 G=G+32
740 NEXT LL
750 GOTO 750
760 REM MILLIONAIRE
770 FOR DD=1 TO 1000:NEXT DD
```

```
780 CLS 3
790 G=96
800 FOR LL=1 TO 9
810 PRINT@ G, TAB(5) "you have made a mi
llion"
820 PLAY "T200V3003AAAAAGGGGGFFFFFDDDDDA
AAAAAAAAAA"
830 FOR DD=1 TO 250:NEXT DD
840 G=G+32
850 NEXT LL
860 GOTO 860
870 CLS7
880 FOR C=1 TO 3
890 PRINT@ 192, TAB(5) "$$$ STOCK EXCHAN
GE $$$"
900 FOR T=1 TO 3
910 PLAY "T20003AAAFFFGGGAAAAA"
920 FOR DD=1 TO 50:NEXT DD
930 NEXT T
940 PRINT@192, TAB(5) "$$$$$$$$$$$$$$$$$
$$$$$"
950 PLAY "T20002GGGFFFDDDAAAA"
960 FOR DD=1 TO 100:NEXT DD
970 NEXT C
980 RETURN
```

```
CO.    YOU   PRICE   START   PROFIT
NO.   HOLD    NOW    PRICE    /LOSS
 1     33     13      78     -65
 2     0      122     32      90
 3     0      183     33      150
 4     0      94      34      60

BANK $ 20687
INTEREST RATE IS 11 %

WHICH COMPANY(1-4) 1

HOW MANY SHARES
(A NEGATIVE MEANS SELL) 1000


CO.    YOU   PRICE   START   PROFIT
NO.   HOLD    NOW    PRICE    /LOSS
 1     1033   7       13     -6
 2     0      202     32      170
 3     0      153     33      120
 4     0      114     34      80

BANK $ 8532
INTEREST RATE IS 11 %

WHICH COMPANY(1-4) 4

HOW MANY SHARES
(A NEGATIVE MEANS SELL) 5
```

```
CO.    YOU  PRICE  START  PROFIT
NO.   HOLD   NOW   PRICE   /LOSS
 1     1033  17     13     4
 2     0     192    32     160
 3     0     93     33     60
 4     5     194    114    80

BANK $ 8678
INTEREST RATE IS 9 %

WHICH COMPANY(1-4) 1

HOW MANY SHARES
(A NEGATIVE MEANS SELL) 200


CO.    YOU  PRICE  START  PROFIT
NO.   HOLD   NOW   PRICE   /LOSS
 1     1233  2      17    -15
 2     0     172    32     140
 3     0     153    33     120
 4     5     134    114    20

BANK $ 6069
INTEREST RATE IS 15 %

WHICH COMPANY(1-4) 1

HOW MANY SHARES
(A NEGATIVE MEANS SELL) 2000
```

```
CO.    YOU  PRICE  START  PROFIT
NO.   HOLD   NOW   PRICE   /LOSS
 1     3233  33     2 ,     31
 2     0     162    32      130
 3     0     153    33      120
 4     5     94     114    -20

BANK $ 2296
INTEREST RATE IS 11 %

WHICH COMPANY(1-4) 1

HOW MANY SHARES
(A NEGATIVE MEANS SELL)-1500


CO.    YOU  PRICE  START  PROFIT
NO.   HOLD   NOW   PRICE   /LOSS
 1     1733  48     2       46
 2     0     132    32      100
 3     0     198    33      165
 4     5     54     114    -60

BANK $ 56457
INTEREST RATE IS 9 %

WHICH COMPANY(1-4)
```

# Concentration

Concentration is a traditional card game which will entertain children of all ages. It is a game which, as its name suggests, involves a great deal of concentration and tests a player's ability to remember accurately what he or she has seen.

The game involves placing a full deck of playing cards face down on a table or any other suitable surface. The cards are usually placed in four rows of thirteen. Each player then takes turns at turning over two cards. If the cards make up a pair, for instance two aces, then the player picks up these two cards and chooses two more until he fails to pick a pair. If the cards aren't a pair, they are turned face down again and it is then the next player's turn to pick two cards.

The strategy of the game is to remember the positions of as many cards as possible to enable you to pick a maximum number of pairs at each turn.

In this computer version your computer will, at the start of the game, shuffle a deck of electronic cards and lay them out neatly on the screen in four rows of thirteen cards. The backs of the cards are coloured alternately red and yellow to aid you in distinguishing their position.

The human player has the first turn. The screen will show a cursor or pointer, a green square, in the upper left-hand corner of the first card. This cursor is used to indicate which card you wish to turn over. You can advance this cursor along the rows of cards by pressing the right arrow key. Each time you press this key the cursor will move one card to the right. When the cursor reaches the end of the row it will drop down to the next row. When it reaches the last card in the deck it will return to the first card on the screen. So if you happen to go past the card you wish to select, never fear, just keep going and you will get back to it eventually. When you have reached the card you wish to turn over, press the "T" key. The

computer will then show you which card you have selected. This card will remain on the screen until after the next card has been picked. If the two cards are a pair, both cards will be removed from the screen, your score will be incremented and you will be given another turn.

If the two cards are not a pair, they will remain on the screen for a few seconds to give you a chance to memorize their positions and then they will be turned back over to show the backs of the cards again. It is then the computer's turn to select two cards. This game cycle continues until all the cards have been paired up. The player with the highest number of paired cards to his credit wins the game.

As we have said, the game of Concentration involves the ability to remember what you have seen. This created a problem in the early programming stages because, of course, a computer never forgets. An infallible computer makes a very frustrating opponent. To make the game interesting and a challenge to play, we had to provide a method for the computer to 'forget' where cards were placed on some occasions. This was done by giving the player a choice of five levels of difficulty. Level one is the easiest, whereas level five is very difficult indeed. At the beginning of the game you are asked to choose which level you wish to play. The INPUT from this question in line 1960, HF, is used in a computation in line 340 which gives the computer a built-in fallibility factor. The lower the level of difficulty, the more likely the computer is to 'forget' where a certain card is located. You will find that at level five the computer has an excellent, but not perfect memory. The program provides quite a good challenge and I think you will find that it requires a great deal of practice before you are able to beat the computer on level five.

The program makes extensive use of subroutines. This aided development of the program and makes for efficient running while the game is being played.

After the title sequence at the beginning of the program, the GOSUB 2060 in line 20, we RETURN to line 30 which DIMensions two arrays. The A array contains the values of the cards in a random order and is used to place the cards in the

four rows on the screen. Line 40 sends the program to the subroutine beginning at line 1830 and running down to line 1990.

This routine first prints the stand-by message on the screen at line 1840 and then uses the loop in lines 1850 to 1890 to read four sets of the numbers from one to thirteen into randomly picked positions in the array. The position is picked in line 1860 and is checked in line 1870 to see if it already contains a 'card'. If it does, the program goes to line 2000 where this routine puts the card into the next available blank location in the array.

When all the array positions have been filled, lines 1910 to 1940 play a little fanfare and then line 1950 asks you which level of difficulty you wish to use. Your INPUT is accepted by line 1960 and then the program is RETURNed to line 50.

We then go to the subroutine in lines 1670 to 1820. This routine uses a series of loops and SET statements to print the backs of the cards on the screen.

Back in the main part of the program again, the loop in lines 60 to 90 sets up the B array. These lines set each location in the B array to minus one. This array is used by the computer to 'remember' the locations of the cards that have been picked by either the human player or the computer. When a card is 'turned over' by either player the value of the card is put into the location in the B array which corresponds with the card's position in the A array. After the computer has selected a card it then searches through the B array for the location of a matching card. If it finds one, the value of that location in the B array is set to zero to show that it is now empty. If the computer doesn't find a matching card in the B array, it will choose its second card at random.

So the A array contains values of either zero or the value of a card, and the B array can contain values of minus one for cards which have yet to be turned over, the value of a card which has been turned over, or the value of zero which indicates that a card has been matched with its pair.

Line 100 prints the current scores. Variable MT is the computer's score. This is incremented in line 570, and the

human's score, YT, is incremented in line 540. The final scores are printed out by lines 710 and 720.

Line 110 sends the program to the subroutine in lines 1060 to 1300. These lines allow the human player to select the cards and turn them over. Line 1060 uses a row of black spaces to make sure the necessary section of the screen is clear for the message printed by lines 1070 and 1080.

The GOSUB 630 in line 1090 goes to the routine which checks to see if there are any cards left to be selected. This routine scans through the B array and if all the locations are set to zero it drops out of the routine 1060 to 1300 and goes to the end routine.

Variables L and M are the coordinates of the cards on the screen. P is used to check if a screen position is green. This means that the card has either been turned over or has been removed from the screen. If P=1 then the screen location is green. The computer will jump the cursor over this location and move on to the next card. The cursor is moved from card to card each time CHR$(9) is pressed. This is the forward arrow key. When "T" is pressed, the card indicated by the cursor is turned over. LL and MM remember the coordinates of the card, and then the program drops out of the loop and returns to line 120.

Variable Q is used to check if the player has made one or two card selections. The GOSUB 1500 in line 130 is for the routine that 'turns' the cards over and prints the face value of the card on the screen. Variables B and C are used to remember the colour of the back of the card so it can be reprinted correctly by another routine. The rest of the routine prints the face value of the card in its correct position. Line 1560 checks to see if the card is an ace, ten, Jack, Queen or King. If so, the program drops down to lines 1600 to 1660 where the correct symbol is printed. (ACE=AA, TEN=10, JACK=JJ, QUEEN=QQ, KING=KK.)

Back in line 140 the B array element is made equal to the position in the A array that has just been selected. Lines 150 and 160 test the value of Q to see if the player has had one or two choices. If only one card has been selected, we go back to

line 100 for another choice. If two cards have been selected, the program jumps down to line 290.

In line 170 variables N and O are used to remember the first card, and line 180 directs the program back to line 100.

We need to know if the human player has picked a pair, so this is done in line 190. If a pair has been selected, the subroutine in lines 510 to 560 is called up. Line 510 uses blank spaces to clear the section of the screen used by line 520 to tell you that a pair has been picked. Line 530 provides a short delay to give you a chance to see what is going on. Your score is incremented by line 540, and line 550 calls up the subroutine 1310 to 1380.

This routine removes the pair of cards from the screen and sets the A and B arrays' locations to zero so that these cards can't be picked again. Line 560 then returns the program to line 100 so that the human player can pick another two cards.

If a pair wasn't picked, line 200 blanks out the previous printed message. Line 210 tells you that you have failed to pick a pair, then line 220 provides a short delay so that you have a chance to memorize the positions of the cards.

Line 230 GOSUBs to lines 1390 to 1490. This routine is used to turn the cards over again to show the backs. Variables B and C are checked to see what colours the cards were. A series of PRINT@ statements using CHR$ are used to print the backs of the cards back on to the screen in the correct places.

Line 240 once again calls up subroutine 630 where a check is made to see if the game is over.

Line 250 sends the program to the subroutine in lines 870 to 1050. This routine is used by the computer to select its first card. Line 870 prints the computer's message, and line 880 clears out the previous message in the next line.

The loop from line 900 to line 1010 is the loop which selects the card. Variables L and M are randomly set to values between 6 and 54, and 6 and 18 respectively. By doing this we choose a position in the A array which becomes the computer's first card. Line 970 checks to see if this card has already been turned over. If it has, the program goes to the next position in

the array. Line 980 checks to see if the card has been removed from the screen; if it has, the program moves on to the next available position in the array.

In line 1020, if Z=2, a card has been selected; we then go down to 1050. If Z=1, then a card hasn't been found between the position selected by L and M and the end of the array. If this occurs then line 1040 sends the program back to line 900 where the loop is started off again.

GOSUB 630 is called up once again by line 1030. We call this routine up frequently to keep checking that there are still cards in the array to be picked.

When a card has been selected, LL and MM in line 1050 remember the coordinates of the card and we go back to line 260. Variable Q is used once again to see if one or two cards have been picked. The GOSUB 1500 in line 270 prints the selection on the screen.

In line 310 the values of the cards are put into the B array. The N and O variables in line 330 remember the coordinates of the first selection.

If only one card has been picked, we use line 340 to see whether the computer will remember that it has seen a matching card. This line gives the computer a degree of probability and selects a random number. If the number is outside the range of probability then line 350 sends the program back to line 250 so that the second card may be picked at random.

If the random number is within the probability range then the computer is given the chance to see if it can remember the position of a matching card. Lines 410 to 470 scan the B array for a card which matches the first one the computer has picked. If one is found, line 450 checks to see that it isn't the card the computer has already picked. When a pair is found, II and JJ in line 460 remember the pair.

Line 480 checks to see if II=0; if so, a pair hasn't been found in the B array. In this case we go back to line 250 so that the second card can be picked at random.

If a pair has been found, line 590 prints the message on the screen. Line 600 sends us to the routine which removes them

from the screen. The program then goes back to line 250 so that the computer can select another two cards.

If the computer doesn't select a pair, the cards are turned back over and the program goes back to line 100 so that the human player can have his next turn.

At the first reading you may find that this explanation of the program is a bit confusing. Rather than run through the program line by line, we have gone through it in the sequence the computer uses when it is playing the game. So we suggest that you play the game a few more times until you are familiar with the order of play and then read through the explanation again, following through the listing of the program as you do so. This should give you an excellent idea of how the program works. Here is the program sequence in step form:

The program first sets up two arrays.
The cards are sorted into the A array.
The backs of the cards are printed on the screen.
The elements of the B array are set to minus one.
The player selects a card.
The card is printed on the screen.
The value of the card is placed in the B array.
A second card is selected and printed on the screen.
The value of the card is placed in the B array.
If a pair has been picked, they are removed from the screen.
If they are a pair, the player gets another turn.
If a pair wasn't picked, the backs of the cards are reprinted.
The computer selects a card at random.
It checks to see if it will remember another card.
If it does remember, a search is made of the B array for a matching card.
If it doesn't remember, or can't find a matching card, a second card is picked at random.
The cards are printed on the screen and placed in the B array.
The cards are either cleared from the screen or the backs are reprinted, depending on whether or not a pair was picked.
This game sequence continues until all the cards have been paired off.

```
10 REM CONCENTRATION
20 GOSUB 2060
30 DIM A(13,4):DIM B(13,4)
40 GOSUB 1830
50 GOSUB 1670
60 FOR X=1 TO 13
70 FOR Y=1 TO 4
80 B(X,Y)=-1
90 NEXT:NEXT
100 PRINT@416,"SCORE:  COMPUTER";MT;"  H
UMAN";YT
110 GOSUB 1060
120 IFQ=2 THEN Q=0
130 GOSUB 1500
140 B(X,Y)=A(X,Y)
150 Q=Q+1
160 IF Q>1 THEN 190
170 N=X:O=Y
180 GOTO 100
190 IF A(X,Y)=A(N,O) THEN 510
200 PRINT@0,"
    "
210 PRINT@10,"NOT A PAIR"
220 FOR DD =1 TO 700:NEXT
230 GOSUB 1390
240 GOSUB 630
250 GOSUB 870 REM COMPUTER SELECTS
260 IF Q=2 THEN Q=0
270 GOSUB 1500 REM PRINT SELECTION
280 FOR I=1 TO 1000:NEXT
290 E=0
300 Q=Q+1
310 B(X,Y)=A(X,Y)
```

```
320 IF Q>1 THEN 360
330 N=X:O=Y
340 IF RND(100)<50+10*HF THEN GOTO 410
350 GOTO 250
360 IF A(X,Y)=A(N,O) THEN 570
370 PRINT@0,"
    "
380 PRINT@10,"I GOOFED"
390 FOR I=1 TO 600:NEXT
400 GOSUB 1390:GOTO 100
410 II=0:JJ=0
420 FOR I=1 TO 13
430 FOR J=1 TO 4
440 IF B(I,J)<>A(X,Y) THEN 470
450 IF I=X AND J=Y THEN 470
460 II=I:JJ=J
470 NEXT:NEXT
480 IF II=0 THEN 250
490 X=II:Y=JJ:E=10
500 GOTO 270
510 PRINT@0,"
    "
520 PRINT@12,"A PAIR"
530 FOR DD=1 TO 700:NEXT
540 YT=YT+1
550 GOSUB 1310
560 GOTO 100
570 MT=MT+1
580 PRINT@0,"
    "
590 PRINT@8,"I PICKED A PAIR"
600 GOSUB 1310
610 FOR I=1 TO 600:NEXT
```

```
620 GOTO 250
630 B=0
640 FOR X=1 TO 13:FOR Y=1 TO 4
650 IF B(X,Y)<>0 THEN B=1
660 NEXT:NEXT
670 IF B=0 THEN 690
680 RETURN
690 GOSUB 2460
700 CLS RND(8)
710 PRINT@128,TAB(7)"MY SCORE IS";MT
720 PRINT TAB(7)"YOUR SCORE IS";YT
730 PRINT TAB(3)"DO YOU WISH TO PLAY AGA
IN"
740 PRINT TAB(10)"(YES/NO)";
750 INPUT Q$
760 IF Q$="YES" THEN 40
770 PRINT "PLEASE PLAY ME AGAIN I NEED T
HE PRACTICE"
780 FOR I=1 TO 1000:NEXT:GOTO 40
790 Z=0
800 FOR I=1 TO 13
810 FOR J=1 TO 4
820 IF Z=1 THEN 850
830 IF B(I,J)<>B(N,O) THEN 850
840 II=I:JJ=J:Z=1
850 NEXT:NEXT
860 RETURN
870 PRINT@5,"IT IS MY TURN TO SELECT"
880 PRINT@32,"
    "
890 Z=0
900 FOR M=6 TO 18 STEP 4
910 FOR L=6 TO 54 STEP 4
```

```
920 IF Z=1 THEN 960
930 IF Z=2 THEN 1010
940 M=2+4*RND(4):L=2+4*RND(13):Z=1
950 IF Z=2 THEN 1010
960 P=POINT(L,M)
970 IF P=-1 THEN 1010
980 IF P=1 THEN 1010
990 LL=L:MM=M:K=P
1000 Z=2
1010 NEXT:NEXT
1020 IF Z=2 THEN 1050
1030 GOSUB 630
1040 IF Z=1 THEN 900
1050 L=LL:M=MM:RETURN
1060 PRINT@0,"
     "
1070 PRINT@0,"USE FORWARD ARROW TO SELEC
T CARD"
1080 PRINT@38,"PRESS T TO TURN CARD"
1090 GOSUB 630
1100 LL=30:MM=30:K=1
1110 A$=""
1120 FOR M=6 TO 18 STEP 4
1130 FOR L=6 TO 54 STEP 4
1140 IF A$="T" THEN 1260
1150 X=(L-2)/4:Y=(M-2)/4
1160 IF B(X,Y)=0 THEN 1260
1170 P=POINT(L,M)
1180 IF P=-1 THEN 1260
1190 IF P=1 THEN 1260
1200 SET(L,M,1):SET(LL,MM,K)
1210 LL=L:MM=M:K=P
1220 A$=INKEY$
```

```
1230 IF A$="T" THEN 1260
1240 IF A$=CHR$(9) THEN 1260
1250 GOTO 1220
1260 NEXT:NEXT
1270 SOUND 200,1
1280 IF A$=CHR$(9) THEN 1120
1290 L=LL:M=MM
1300 RETURN
1310 FOR H=0 TO 1
1320 B(X,Y)=0:A(X,Y)=0
1330 PRINT@T,CHR$(143);CHR$(143);
1340 PRINT@T+32,CHR$(143);CHR$(143);
1350 T=99+2*(N-1)+64*(O-1)
1360 X=N:Y=O
1370 NEXT
1380 RETURN
1390 FOR H=0 TO 1:Q=0
1400 IF B=4 THEN 1440
1410 PRINT@T,CHR$(159);CHR$(159);
1420 PRINT@T+32,CHR$(159);CHR$(159);
1430 IF B=2 THEN 1460
1440 PRINT@T,CHR$(191);CHR$(191);
1450 PRINT@T+32,CHR$(191);CHR$(191);
1460 T=99+2*(N-1)+64*(O-1)
1470 B=C
1480 NEXT
1490 RETURN
1500 IF Q=0 THEN C=K
1510 B=K:H=0
1520 IF E=10 THEN 1540
1530 X=(L-2)/4:Y=(M-2)/4
1540 FOR A=0 TO 32 STEP 32
1550 T=97+2*X+64*(Y-1)+A:T2=48+A(X,Y)
```

```
1560 IF T2<50 OR T2>57 THEN 1600
1570 PRINT@T,CHR$(T2);CHR$(T2);
1580 NEXT:T=T-32
1590 RETURN
1600 IF T2=49 THEN PRINT@T,CHR$(65);CHR$
(65);
1610 IF T2=58 THEN PRINT@T,CHR$(49);CHR$
(48);
1620 IF T2=59 THEN PRINT@T,CHR$(74);CHR$
(74);
1630 IF T2=60 THEN PRINT@T,CHR$(81);CHR$
(81);
1640 IF T2=61 THEN PRINT@T,CHR$(75);CHR$
(75);
1650 IF T2=48 THEN PRINT@T,CHR$(63);CHR$
(63);
1660 GOTO 1580
1670 REM BACK OF CARDS
1680 FOR D=0 TO 1
1690 FOR B=0 TO 6
1700 FOR A=0 TO 2 STEP 2
1710 SET(6+8*B+A,6+8*D,4)
1720 SET(6+8*B+A,8+8*D,4)
1730 IF B=6 THEN 1760
1740 SET(10+8*B+A,6+8*D,2)
1750 SET(10+8*B+A,8+8*D,2)
1760 SET(6+8*B+A,10+8*D,2)
1770 SET(6+8*B+A,12+8*D,2)
1780 IF B=6 THEN 1810
1790 SET(10+8*B+A,10+8*D,4)
1800 SET(10+8*B+A,12+8*D,4)
1810 NEXT:NEXT:NEXT
1820 RETURN
```

```
1830 CLS RND(8)
1840 PRINT@192, TAB(4)"I AM SHUFFLING TH
E CARDS"
1850 FOR X=1 TO 13:FOR Y=1 TO 4
1860 XX=RND(13):YY=RND(4)
1870 IF A(XX,YY)>0 THEN 2000
1880 A(XX,YY)=X
1890 NEXT Y:NEXT X
1900 U=0:W=0
1910 FOR SS=100 TO 200 STEP 10
1920 SOUND SS,1:NEXT
1930 FOR SS=180 TO 100 STEP -10
1940 SOUND SS,1 :NEXT
1950 PRINT@192, TAB(5)"LEVEL OF PLAY (1-
5)  ";
1960 INPUT HF
1970 CLS 1
1980 FOR P=1 TO 6:PLAY"T200V3004ABCDEFGP
2":NEXT
1990 RETURN
2000 FOR G=1 TO 13:FOR H=1 TO 4
2010 IF A(G,H)<>0 THEN 2040
2020 XX=G:YY=H
2030 G=13:H=4
2040 NEXT:NEXT
2050 GOTO 1880
2060 PCLS 5
2070 PMODE 0,1
2080 SCREEN 1,1
2090 COLOR 0,5
2100 FOR RR=10 TO 150 STEP 5
2110 CIRCLE (128,96),RR
2120 NEXT RR
```

```
2130 COLOR 5,0
2140 FOR RR=150 TO 10 STEP -5
2150 CIRCLE (128,96),RR
2160 NEXT RR
2170 COLOR 0,5
2180 XX=0:YY=0
2190 WW=258:ZZ=192
2200 FOR TT=1 TO 20
2210 LINE (XX,YY)-(WW,ZZ),PSET,B
2220 XX=XX+5:YY=YY+5
2230 WW=WW-5:ZZ=ZZ-5
2240 NEXT TT
2250 XX=100:YY=100
2260 WW=158:ZZ=92
2270 FOR TT=20 TO 1 STEP -1
2280 LINE (XX,YY)-(WW,ZZ),PRESET,B
2290 XX=XX-5:YY=YY-5
2300 WW=WW+5:ZZ=ZZ+5
2310 NEXT TT
2320 CLS RND(8)
2330 FOR HH=1 TO 5
2340 PRINT@160," "
2350 PRINT@192,"**************************
********"
2360 FOR TT=1 TO 10
2370 SS=100+TT*5
2380 SOUND SS,1
2390 NEXT
2400 PRINT@192,"*******  CONCENTRATION
********"
2410 FOR TT=1 TO 10
2420 SS=175+TT*5
2430 SOUND SS,1
```

```
2440 NEXT:NEXT
2450 FOR DD=1 TO 300:NEXT:RETURN
2460 FOR CC=1 TO 8
2470 CLS CC
2480 SOUND 100+CC*5,1
2490 NEXT CC
2500 FOR CC=8 TO 1 STEP -1
2510 CLS CC
2520 SOUND 100+CC*5,1
2530 NEXT CC
2540 PCLS 5
2550 PMODE 1,1
2560 SCREEN 1,1
2570 FOR PP=10 TO 250 STEP 10
2580 CIRCLE (PP,96),50
2590 NEXT
2600 PAINT (0,0),7,8
2610 PAINT (0,192),7,8
2620 RETURN
```

# Appendices

# How to create moving graphics

Several of the programs in this book, such as Escape and Maze Runner, use a very easy method of providing moving graphics on the text screen. Instead of using PRINT or PRINT@ to print messages on the screen, we have used the statement POKE to place a character directly on to the screen. A section of the computer's memory is set aside to control what appears on the screen. By POKEing values directly into these locations we can place characters directly on to the screen and move them around much faster than using any of the BASIC commands.

The text screen is divided up into 16 rows down the screen, the Y axis; each of these rows is made up of 32 spaces across the screen, the X axis, a total of 512 locations. The memory locations which control these screen spaces are numbered from 1024 in the top left-hand corner down to 1536 in the bottom right-hand corner. The numbers run across the screen.

To produce moving graphics we simply POKE a value into one of these memory locations, let it stay there long enough to be seen, then rub it out and POKE it into a new location nearby. If we do this over and over again, the object on the screen appears to move. To control this movement we need several things: a method of keeping track of the character's location, a method of making it move (that is, changing its location), and a method of controlling the direction in which it is moving.

To keep track of a character's position on the screen we use a simple formula: POKE 1024 + X + Y × 32. 1024 is the first location on the screen. This gives a constant point of reference to work from. Variable X represents the number of spaces our character is across the screen. Variable Y (the character's distance down the screen) must be multiplied by 32 to give the correct number of screen locations to add to X.

We now need a method of altering the values of X and Y so that we can move our character about.

Type in and RUN this program and, after you have seen it in action, we will discuss it line by line.

```
10 REM BOUNCING BALL
20 CLS
30 X=16:Y=8
40 XX=1:YY=1
50 EX=X:EY=Y
60 POKE 1024+X+Y*32,79
70 X=X+XX
80 Y=Y+YY
90 IF X<1 THEN XX=-XX
100 IF X>30 THEN XX=-XX
110 IF Y<1 THEN YY=-YY
120 IF Y>14 THEN YY=-YY
130 POKE 1024+EX+EY*32,96
140 GOTO 50
```

Line 10 is a REM statement, so that if you come across this program again in a few months' time you will know what it is supposed to do. Line 20 clears the screen so that we can see the following action.

Line 30 sets up the variables X and Y. These variables will remember the position of our bouncing ball. This line gives them the starting position of the ball, sixteen spaces across the screen and eight rows down. This puts the ball right in the centre of the screen for the start of the program.

In line 40, variables XX and YY are the amounts we are going to use when we increment X and Y. In this program we are going to move the ball one space at a time.

Line 50 creates the variables EX and EY (these stand for Erase X and Erase Y). We will use them to blank out the ball

before we POKE it into a new location. Line 50 is also the start of our loop.

Line 60 is the line which places the ball on to the screen. We POKE memory location 1024 plus the value of X plus the value of Y multiplied by 32. The number 79 which follows is the POKE value for the letter O.

Lines 70 and 80 increment the values of X and Y by the values of XX and YY. This moves the ball across one space on the X axis and down one space on the Y axis.

Lines 90 to 110 are very important. These lines check to see if the ball is about to hit the edges of the screen and, if it is, the value of XX or YY, as the case may be, is then converted to a negative. The next time through our loop this value will be added to the variable X or Y and, because adding a negative subtracts it, will change the direction of the ball. This will make it appear to bounce off the edge of the screen. If XX or YY already hold a negative value when this occurs, these lines will convert them back to positive (a double negative creates a positive).

Line 130 uses EX and EY (the location of the ball on the screen, X and Y have since been incremented by lines 70 and 80) to blank out the ball. Value 96 is the POKE code for a blank space. When using POKE the value 32 produces a reverse space.

Line 140 sends the program back to line 50 where EX and EY are given the current values of X and Y, and we go through the whole process again.

The speed of the ball can be controlled by placing a delay loop at line 65.

As you may have gathered by now, the character set available when using POKE is different from the character set we get with CHR$. From value 128, the start of the graphic set, onwards, they are the same, but the alphabet, the numerals and mathematical symbols are different.

The following short program will display the character set on the screen for you.

```
10 REM POKE CHARACTER DISPLAY
20 CLS
30 FOR X=1 TO 255
40 POKE 1023+X,X
50 NEXT X
60 GOTO 60
```

A final word of warning: when using POKE to move things around the screen, make sure you put in safeguards so that the X and Y values don't get outside of the range 1024 to 1536. These memory locations are the 512 which control the screen. If the ball goes off the top of the screen – that is, into memory locations lower than 1024 – or out the bottom of the screen – locations past 1536 – you are then POKEing a value into memory locations which control other functions. All sorts of disasters can befall you if this happens. I have had the keyboard freeze up, resulting in the computer having to be switched off before control could be regained. This, of course, caused the loss of the program I was working on. I have also had a program scrambled into garbage by a 'lost ball'.

Study the bouncing ball program carefully and use this basic principle to produce your own programs.

# Games to convert

You may well find that, while you may have little trouble writing a game for your computer once the idea for it has been presented, getting the idea itself may not be so simple. In this section of the book, I'll be looking at a number of game ideas which seem ideal for conversion.

Many times I've discovered, when trying to convert a 'traditional' (that is, non-computer) game to my computer, that the program which I finally end up with plays a game with little resemblance to the original idea. I don't think that matters at all. So long as the game is fun to play, the route followed to that game is unimportant.

You may also find that just reading through the games presented here will spark an idea for a program which is totally unrelated to the outline given. This section of the book has been designed to act as a series of 'idea-starters'. Make your own choice on whether you intend to follow the outlines, or simply use them as take-off points for your own creativity.

## Checker derivatives

You have probably seen games which could be described as 'checker derivatives'; that is, games which are based on – but differ in at least one important particular from – checkers (or 'draughts' as the game is generally known in the UK). It doesn't take too much thought to realize that the eight-by-eight checkerboard and the normal checkermen and their moves can form the basis of a whole series of intelligent board games.

All you need to do is to postulate a 'what if', and then write your game from that point. Here are some examples which

should explain the 'what if' idea. You're sure to be able to think of others.

Use a board which is nine-by-nine, or seven-by-seven, with a corresponding change in the number of men.

Remove multiple jumps.

Allow men to move in all directions, as though you had a board of kings.

Change the shape of the board, to become hexagonal, rectangular or five-sided.

Add one or two special pieces, such as one which can move like the queen in chess.

Introduce a new method of capture, such as 'squeezing' a piece between two of the opponent's pieces.

Change the object of the game, so that the player who first makes three kings wins, or the first player to get rid of all of his or her pieces is the victor.

Make the game three-handed, and allow the computer to play two of the hands, with the human as the third player.

And so on. I am sure there are many other ideas you can think of which fit under the umbrella of 'checker derivatives'.

## Mancala

One of man's oldest known games has a number of names, including 'Mancala', 'Owari', 'Awari' and 'Kalah'. Mancala boards have been discovered in 3000-year-old Egyptian temples. It is still widely played in Africa and Asia, often under slightly different rules.

The basis of the game is for both players to have a number of 'cups' (or depressions in the ground) in a row in front of them. There is a larger, 'home cup' at each end of the rows.

The playing field looks like this, with the home cups designated by the letter H:

```
                Computer's side
              O O O O O O
Human's home> H                 H <Computer's home
              O O O O O O
                Human's side
```

At the start of the game, each cup contains a number of seeds (generally four or six), with the home cups empty. The players take it in turn to move. All of the stones in one cup are picked up, and the player then moves in a clockwise (or anti-clockwise, in some versions of the game) direction, dropping a single seed into each cup (except the home cups) as it is passed over.

If the final seed lands in an empty cup on the player's side, then all the seeds in the facing, opponent's cup are transferred into the player's home cup. The game continues until one of the players cannot move, which will happen when there are no seeds on his or her side. The winner is the player with the most seeds in the home cup.

Other variations of the game allow for a seed to be dropped in the player's home cup, but not the opponent's one, on each move. If the final seed lands in a player's home cup, the player has a second move.

This seems a simple game, and it is relatively easy to program. In fact, in an article on the game in *Creative Computing* magazine (May 1980) David Levy points out that 'the game presents an ideal programming exercise because the rules are simple . . . and it is relatively easy to devise a satisfactory evaluation function'.

Despite the simplicity of play, Sid Sackson claims in his book *A Gamut of Games* (Pantheon Books, New York, 1982) that although the game is 'at first glance deceptively simple . . . with repeated play the many subtleties of the strategy begin to emerge'. L. D. and P. H. Stander, in an article in *Microcomputing* (November 1979) agree, claiming that 'competently played, Mancala can be as challenging as chess'.

## Card games

I won't even attempt to list card games in this section, as you're sure to know several of them, and books of card game rules are easy to acquire. Most card games can become computer games by (a) simplifying the rules; (b) reducing the number of cards; and/or (c) replacing the cards with a set of dice.

Poker, for example, can be viewed as the basis for many games in which particular combinations of cards (or numbers, if you're using dice) have higher values than others. The rules by which you exchange unwanted cards or numbers for new ones can determine the nature of the game. No exchange, and you have a simple game of chance. Limited exchange, and chance still plays a part, although skill is required to know which cards to discard. Unlimited exchange, and the skill factor becomes far more important.

## Out on the field

You may wish to try and write a game which simulates an outdoor sport, such as football, tennis or baseball.

The simulation can be based on similar games played with cards or dice, or you can follow a completely different route, and use the graphics potential of your computer, to produce a game which is played in real-time.

## Proprietary board games

Although you cannot develop such programs for commercial sale (as they may infringe the copyright on the original version), proprietary games are a rich source of program ideas.

Obvious targets such as Monopoly (Parker Brothers) and Diplomacy (Avalon Hill) suggest themselves, but there are

many others which deserve consideration. These include Clue (Parker Brothers), in which a murder must be solved; Oregon Trail (Fantasy Games Unlimited), in which you attempt to cross the western United States while coping with Indians and a far from hospitable environment; and Kensington (Samuel Ward) in which players move interlocking shapes in an attempt to relocate opponents' pieces by surrounding a triangle or square, and in which surrounding a hexagon wins.

Have a look through any large toy or games shop and you'll discover many worthwhile ideas.

## Dice games

A book which concentrates on these is the obvious place to get detailed breakdowns of the game and suggested strategy, but I'll mention a few here which may prove interesting and of value:

**Craps.** Two dice are rolled. If the total is 7 or 11, the player wins immediately. If the total is 2, 3 or 12 ('craps') the player loses, and the dice are handed to the other player for his or her move. Any other total is called the 'point'. The player attempts to roll his or her point again, before rolling a 7. If a 7 is rolled first, the player loses.

**Pig.** A single die is used. Players try to get a total of 100 points. Players can end their turn at any time, and hand the die to the opponent. A player's total reverts to zero if a 1 is rolled.

**Chuck-a-Luck.** Three dice are used in this game based on Crown and Anchor. The player (or players) chooses a number from 1 to 6, and places a bet on this number. The three dice are thrown and each one which comes up with the player's number pays back the value of the bet. Therefore, if just one of the three dice lands showing the chosen number, the player gets his or her money back. Two dice showing the number, and twice the value of the bet is returned, and so on.

**Fantasy and role-playing**

If you enjoy games like TSR Hobbies' Dungeons and Dragons (the game and name are registered trademarks of TSR) then you're sure to be interested in games which explore the same fantasy worlds. There are many commercial programs which cover this field. Examining these will give you some clues on how to proceed in writing your own.

A particularly useful book to assist you in this is *Writing BASIC Adventure Programs for the TRS-80* by Frank Dacosta (TAB Books Inc., 1982). Despite being written specifically for the TRS-80, you'll find a great deal of value in it, no matter which computer you use.

## OTHER SOURCES OF IDEAS

There are many games books which can give you ideas for programs. Among those I've found most suitable for this task are:

*The Complete Book of Indoor Games* edited by Peter Arnold (Rigby Publishers Ltd, 1982)

*A Player's Guide to Table Games* by John Jackson (Stackpole Books, 1975)

*A Gamut of Games* by Sid Sackson (Pantheon Books, 1982)

*Games for Two* by John Wasley (Proteus Publishing Co. Inc., 1981)

*Dice Games New and Old* by William E. Tredd (The Oleander Press, 1981)

*Discovering Old Board Games* by R.C. Bell (Shire Publications Ltd, 1980)

*Discovering Dice and Dominoes* by R.C. Bell (Shire Publications Ltd, 1980)

*Dicing with Dragons, An Introduction to Role-Playing Games* by Ian Livingstone (Routledge and Kegan Paul, 1982)

*Everyman's Indoor Games* by Gyles Brandreth and J.M. Dent Sons Ltd, 1982)

# Error trapping

You'll find that it is worth spending time with your programs, once you get them up and running, to make them as 'robust' as possible. That is, you should try to make them as error-proof as you can, so that entering the wrong kind of input (such as hitting a letter key when a number is expected) does not cause the program to crash.

The name 'error trapping' is given to any lines which are contained within a program which will catch input errors before they cause the program to crash.

A program which is left running in an unattended computer must be robust. If the program is expecting a numeric input, and a user keys in words, the program should be able to cope and respond. It can do this either simply by ignoring the input, or by giving an error warning, and a fresh chance to give the correct input. The program should certainly not crash.

The following short program – which was written to demonstrate error trapping – is a simple one to generate mathematics problems. Although the program would probably be most suitable for children aged between six and nine, it can easily be adapted for other groups.

```
10 SC=0
20 CLS
30 PRINT@ 66, "TYPE IN YOUR NAME"
40 INPUT A$
50 IF A$="" THEN GOTO 40
60 IF LEN(A$)>11 THEN GOTO 40
70 FOR S=1 TO 10
```

```
80 CLS
90 A=RND(10):B=RND(10)
100 IF B=A THEN GOTO 90
110 PRINT@ 98, "NOW THEN ";A$:PRINT "  C
HOOSE WHICH TYPE OF SUM":PRINT "  YOU WO
ULD LIKE"
120 PRINT@ 194, "1 - ADDITION"
130 PRINT@ 226, "2 - SUBTRACTION"
140 PRINT@ 258, "3 - MULTIPLICATION"
150 PRINT@ 290, "ENTER 1, 2 OR 3"
160 W$=INKEY$
170 IF W$="" THEN GOTO 160
180 IF W$<>"1" AND W$<>"2" AND W$<>"3" T
HEN GOTO 160
190 W=VAL(W$)
200 IF W=1 THEN AN=A+B:B$="PLUS"
210 IF W=2 THEN AN=A-B:B$="MINUS"
220 IF W=3 THEN AN=A*B:B$="TIMES"
230 CLS
240 PRINT@ 132, "RIGHT THEN, ";A$
250 PRINT@ 196, "WHAT IS";A;B$;B;
260 INPUT Y$
270 IF Y$="" THEN GOTO 260
280 CLS
290 X=VAL(Y$)
300 IF X=AN THEN PRINT@ 260, "WELL DONE
";A$:SC=SC+1
310 IF X<>AN THEN PRINT@ 258, "SORRY, ";
A$;" YOU ARE WRONG":PRINT "    THE ANSWE
R WAS";AN
320 PRINT:PRINT " PRESS ANY KEY FOR ANOT
HER TURN"
330 W$=INKEY$
```

```
340 IF W$="" THEN GOTO 330
350 FOR DD=1 TO 500:NEXT DD
360 CLS
370 NEXT S
380 CLS
390 PRINT@ 192, "YOU SCORED";SC;"OUT OF
TEN"
400 PRINT:PRINT "PRESS ANY KEY TO PLAY A
GAIN"
410 W$=INKEY$
420 IF W$="" THEN GOTO 410
430 FOR DD=1 TO 500:NEXT DD
440 RUN
```

Let's have a look at the program, line by line:

30 This line requires an input for a name.
50 This checks to make certain that something is keyed in, so that just pressing ENTER is not allowed.
60 In this line the length of the entered name is checked. This can be important if the name is to be displayed on the screen. If it is too long, it can lead to an ugly display later on, with broken lines and the like. It is also important to check the length of the name if it is to be later used in an array. If the name is too long, it may simply be chopped off.
90 This generates two random numbers in the range zero to ten. These values can easily be changed if required.
100 This line checks that the second number is different from the first.
160–180 Only the numbers 1, 2 and 3 are allowed here. These are entered as strings rather than as numbers. It is much easier to error-trap strings than it is to check on

numbers. Once the string input has passed the error traps, it can be changed into a number using VAL.

200, 210, 220 The required sum is generated here, along with its corresponding description.

260 The answer is accepted here, again as a string.

270 This disallows the 'empty string' (that is, it rejects just pressing ENTER).

Line 170 forces the computer to wait until a key has been pressed. As you know, you do not need to use ENTER when INKEY$ is employed for accepting input.

Line 180 makes sure that no other option – other than one of the three given options – can be accepted.

# Glossary

**Accumulator** part of the computer's logic unit which stores the intermediate results of computations.

**Address** a number which refers to a location, generally in the computer's memory, where information is stored.

**Algorithm** the sequence of steps used to solve a problem.

**Alphanumeric** generally used to describe a keyboard, and signifying that the keyboard has alphabetical and numerical keys. A numeric keypad, by contrast, only has keys for the digits one to nine, with some additional keys for arithmetic operations, much like a calculator.

**APL** this stands for Automatic Programming Language, a language developed by Iverson in the early 1960s, which supports a large set of operators and data structures. It uses a non-standard set of characters.

**Application software** these are programs which are tailored for a specific task, such as word processing, or to handle mailing lists.

**ASCII** stands for American Standard Code for Information Exchange. This is an almost universal code for letters, numbers and symbols, which has a number between 0 and 255 assigned to each of these, such as 65 for the letter A.

**Assembler** this is a program which converts another program written in an assembly language (which is a computer program in which a single instruction, such as ADD, converts into a single instruction for the computer) into the language the computer uses directly.

**BASIC** stands for Beginner's All-purpose Symbolic Instruction Code, the most common language used on microcomputers. It is easy to learn, with many of its statements being very close to English.

**Batch** a group of transactions which are to be processed by a computer in one lot, without interruption by an operator.

**Baud** a measure of the speed of transfer of data. It generally stands for the number of bits (discrete units of information) per second.

**Benchmark** a test which is used to measure some aspect of the performance of a computer, which can be compared to the result of running a similar test on a different computer.

**Binary** a system of counting in which there are only two symbols, 0 and 1 (as opposed to the ordinary decimal system, in which there are ten symbols, 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9). Your computer 'thinks' in binary.

**Boolean Algebra** the algebra of decision-making and logic, developed by English mathematician George Boole, and at the heart of your computer's ability to make decisions.

**Bootstrap** a program, run into the computer when it is first turned on, which puts the computer into the state where it can accept and understand other programs.

**Buffer** a storage mechanism which holds input from a device such as keyboard, then releases it at a rate which the computer dictates.

**Bug** an error in a program.

**Bus** a group of electrical connections used to link a computer with an ancillary device, or another computer.

**Byte** the smallest group of bits which makes up a computer word. Generally a computer is described as being 'eight bit' or 'sixteen bit', meaning the word consists of a combination of eight or sixteen zeros or ones.

**Central Processing Unit (CPU)** the heart of the computer, where arithmetic, logic and control functions are carried out.

**Character code** the number in ASCII (see ASCII) which refers to a particular symbol, such as 32 for a space and 65 for the letter A.

**COBOL** stands for Common Business Orientated Language, a standard programming language, close to English, which is used primarily for business.

**Compiler** a program which translates a program written in a high-level (human-like) language into a machine language which the computer is able to understand directly.

**Concatenate** to add (adding two strings together is known as 'concatenation').

**CP/M** these initials stand for Control Program/Microcomputer, an almost universal disk operating system developed and marketed by Digital Research, Pacific Grove, California.

**Data** a general term for information processed by a computer.

**Database** a collection of data, organized to permit rapid access by computer.

**Debug** to remove bugs (errors) from a program.

**Disk** a magnetic storage medium (further described as a 'hard disk', 'floppy disk' or even 'floppy') used to store computer information and programs. The disks resemble, to a limited extent, 45 rpm sound records, and are generally eight, five and a quarter, or three and a half inches in diameter. Smaller 'microdisks' are also available for some systems.

**Documentation** the written instructions and explanations which accompany a program.

**DOS** stands for Disk Operating System (and generally pronounced 'doss'), the versatile program which allows a computer to control a disk system.

**Dot-matrix printer** a printer which forms the letters and symbols by a collection of dots, usually on an eight-by-eight, or seven-by-five, grid.

**Double-density** adjective used to describe disks when recorded using a special technique which, as the name suggests, doubles the amount of storage the disk can provide.

**Dynamic memory** computer memory which requires constant recharging to retain its contents.

**EPROM** stands for Erasable Programmable Read Only Memory, a device which contains computer information in a semi-permanent form, demanding sustained exposure to ultra-violet light to erase its contents.

**Error messages** information from the computer to the user, sometimes consisting only of numbers or a few letters, but

generally of a phrase (such as 'Out of memory') which points out a programming or operational error which has caused the computer to halt program execution.

**Field** a collection of characters which form a distinct group, such as an identifying code, a name or a date; a field is generally part of a record.

**File** a group of related records which are processed together, such as an inventory file or a student file.

**Firmware** the solid components of a computer system are often called the 'hardware', the programs, in machine-readable form on disk or cassette, are called the 'software', and programs which are hardwired into a circuit are called 'firmware'. Firmware can be altered, to a limited extent, by software in some circumstances.

**Flag** this is an indicator within a program, with the 'state of the flag' (i.e. the value it holds) giving information regarding a particular condition.

**Floppy disk** see disk.

**Flowchart** this is a written layout of program structure and flow, using various shapes, such as a rectangle with sloping sides for a computer action, and a diamond for a computer decision. A flowchart is generally written before any lines of program are entered into the computer.

**FORTRAN** a high-level computer language, generally used for scientific work (from FORmula TRANslation).

**Gate** a computer 'component' which makes decisions, allowing the circuit to flow in one direction or another, depending on the conditions to be satisfied.

**GIGO** acronym for 'Garbage In Garbage Out', suggesting that if rubbish or wrong data is fed into a computer, the result of its processing of such data (the output) must also be rubbish.

**Global** a set of conditions which affects the entire program is called 'global', as opposed to 'local'.

**Graphics** a term for any output of the computer which is not alphanumeric, or symbolic.

**Hard copy** information dumped to paper by a printer.

**Hardware** the solid parts of the computer (see 'software' and 'firmware').

**Hexadecimal** a counting system often used by machine code programmers because it is closely related to the number storage methods used by computers, based on the number 16 as opposed to our 'ordinary' number system which is based on 10.

**Hex pad** a keyboard, somewhat like a calculator, which is used for direct entry of hexadecimal numbers.

**High-level languages** programming languages which are close to English. Low-level languages are closer to those which the computer understands. Because high-level languages have to be compiled into a form which the computer can understand before they are processed, high-level languages run more slowly than do their low-level counterparts.

**Input** any information which is fed into a program during execution.

**I/O** stands for Input/Output port, a device the computer uses to communicate with the outside world.

**Instruction** an element of programming code, which tells the computer to carry out a specific task. An instruction in assembly language, for example, is ADD which (as you've probably guessed) tells the computer to carry out an addition.

**Interpreter** converts the high-level ('human-understandable') program into a form which the computer can understand.

**Joystick** an analogue device which feeds a signal into a computer which is related to the position which the joystick is occupying; generally used in games programs.

**Kilobyte** the unit of language measurement.

**Line printer** a printer which prints a complete line of characters at one time.

**Low-level language** a language which is close to that used within the computer (see high-level language).

**Machine language** the step below a low-level language; the language which the computer understands directly.

**Memory** the device or devices used by a computer to hold information and programs being currently processed, and for the instruction set fixed within a computer which tells it how to carry out the demands of the program. There are basically two types of memory (see RAM and ROM).

**Microprocessor** the 'chip' which lies at the heart of your computer. This does the 'thinking'.

**Modem** stands for MOdulator/DEModulator, and is a device which allows one computer to communicate with another via the telephone.

**Monitor** (a) a dedicated television-screen for use as a computer display unit, contains no tuning apparatus; (b) the information within a computer which enables it to understand and execute program instructions.

**Motherboard** a unit, generally external, which has slots to allow additional 'boards' (circuits) to be plugged into the computer to provide facilities (such as high resolution graphics, or 'robot control') which are not provided with the standard machine.

**Mouse** a control unit, slightly smaller than a box of cigarettes, which is rolled over the desk, moving an on-screen cursor in parallel to select options and make decisions within a program. 'Mouses' work either by sensing the action of their wheels, or by reading a grid pattern on the surface upon which they are moved.

**Network** a group of computers working in tandem.

**Numeric pad** a device primarily for entering numeric information into a computer, similar to a calculator.

**Octal** a numbering system based on eight (using the digits 0, 1, 2, 3, 4, 5, 6 and 7).

**On-line** device which is under the direct control of the computer.

**Operating system** this is the 'big boss' program or series of programs within the computer which controls the computer's operation, doing such things as calling up routines when they are needed and assigning priorities.

**Output** any data produced by the computer while it is processing, whether this data is displayed on the screen or dumped to the printer, or is used internally.

**Pascal** a high-level language, developed in the late 1960s by Niklaus Wirth, which encourages disciplined, structured progamming.

**Port** an output or input 'hole' in the computer, through which data is transferred.

**Program** the series of instructions which the computer follows to carry out a predetermined task.

**PILOT** a high-level language, generally used to develop computer programs for education.

**RAM** stands for Random Access Memory, and is the memory on board the computer which holds the current program. The contents of RAM can be changed, while the contents of ROM (Read Only Memory) cannot be changed under software control.

**Real-time** when a computer event is progressing in line with time in the 'real world', the event is said to be occurring in real-time. An example would be a program which showed the development of a colony of bacteria which developed at the same rate that such a real colony would develop. Many games, which require reactions in real-time, have been developed. Most 'arcade action' programs occur in real-time.

**Refresh** the contents of dynamic memories (see memory) must receive periodic bursts of power in order for them to maintain their contents. The signal which 'reminds' the memory of its contents is called the refresh signal.

**Register** a location in computer memory which holds data.

**Reset** a signal which returns the computer to the point it was in when first turned on.

**ROM** see RAM.

**RS-232** a standard serial interface (defined by the Electronic Industries Association) which connects a modem and associated terminal equipment to a computer.

**S-100 bus** this is also a standard interface (see RS-232) made up of 100 parallel common communication lines which are used to connect circuit boards within microcomputers.

**SNOBOL** a high-level language, developed by Bell Laboratories, which uses pattern recognition and string manipulation.

**Software** the program which the computer follows (see firmware).

**Stack** the end point of a series of events which are accessed on a last in, first out basis.

**Subroutine** a block of code, or program, which is called up a number of times within another program.

**Syntax** as in human languages, the syntax is the structure rules which govern the use of a computer language.

**Systems software** sections of code which carry out administrative tasks, or assist with the writing of other programs, but which are not actually used to carry out the computer's final task.

**Thermal printer** a device which prints the output from the computer on heat-sensitive paper.

**Time-sharing** this term is used to refer to a large number of users, on independent terminals, making use of a single computer, which divides its time between the users in such a way that each of them appears to have the 'full attention' of the computer.

**Turnkey system** a computer system (generally for business use) which is ready to run when delivered, needing only the 'turn of a key' to get it working.

**Volatile memory** a memory device which loses its contents when the power supply is cut off.

Turn your Dragon 32 into an all-singing, all-dancing games machine! This giant collection of games programs will show you just how effective and exciting a computer your Dragon can be. Outstanding programs include:

**CRAZY HIGHWAY:** a game full of tension as you try to get a poor little cat across a busy highway

**PRO GOLF:** in this stunning arcade-standard program you see the hole in high-resolution graphics

**METEOR:** control your laser base on the ground to destroy the hired spaceship hovering above you

**CAVERN OF THE SHADOW THIEVES:** adventure gaming comes to your Dragon as you stumble through the labyrinth

**AVALANCHE:** as mayor of an Alpine village, the lives of the people are in your hands

**Plus** you'll learn how to create moving graphics on your Dragon and how to trap player input errors. The appendix includes a series of games ideas for you to turn into programs, and there's a handy glossary of computer words.

ISBN 0-00-636807-7

U.K. £3.95

9 780006 368076 90000

FONTANA

GIANT BOOK OF GAMES FOR YOUR DRAGON

6807