

THOUGHT GAMES

```
****      by R. McGregor      ****
*****
*****

You are in a hut

Exits :
East:

You can see :
a hammer
What now?TAKE HAMMER
-----
O.K.
What now?E
-----

You are in a garden

Exits :
South:West:

You can see :
nothing.
What now?
```

HIGHER/LOWER
by R.Bailey



This program is based on the popular TV game, 'Play your cards right', where you have to predict whether the next card in a sequence will be higher or lower than the previous one.

Instructions

You begin with a credit of 250 points, and a target of 2500. You can bet between 50 and your current credit balance at each turn.

At the first, fourth and seventh card, you are offered a choice of changing the card. If you reach your target at the ninth card, you will begin a new round with your credit added to your total score. If your credit runs out or you don't reach the target, you are offered a new game.

Programming techniques

The design on the reverse of the cards is generated by PLOT commands in the procedure PROCdraw. Only one user defined character (line 80) is used in this game, and its purpose is to create a figure '10' as a single character. This means that the number fits neatly on the cards in the same way as the lower single-digit numbers.

```

10 REM Higher / Lower
20 REM by R.Bailey
30 REM BEEBUG
40 REM VERSION P 1.0
50 :
60 ON ERROR GOTO 1230
70 X=RND(-TIME)
80 VDU23,224,206,219,219,219,219,219,206,0
90 MODE5:DIMA(12),B$(9),S(12):@%=&00000908
100 BS=0
110 VDU28,4,16,19,0:VDU19,128,4,0,0,0:VDU19,2,0,0
,0,0
120 RT=0:CLS
130 C=250:MB=50:AIM=2500
140 F=0:PROCtitles:RESTORE:FORJ=1TO9:READX,Y:PROC
draw:NEXT
150 PROCset:PROCpicture:RESTORE:F=1
160 FORJ=1TO9:READX,Y:PROCdraw:PROCshow:IFJ=9THEN
200
170 IFC<MB J=9:GOTO210
180 IFJ=1ORJ=4ORJ=7THENPROCchange
190 PROCbet:PROCupdate:PROChigherlower
200 NEXT
210 CLS:RT=RT+C:IFC>=AIM THEN330
220 IFRT>BS THENBS=RT
230 IFC<MB THEN310
240 PRINTTAB(0,2)"You have failed"
250 PRINTTAB(1,4)"to reach the"
260 PRINTTAB(4,6)"TARGET"
270 PROCbestscore
280 PRINTTAB(0,8)"ANOTHER GAME?"
290 PRINTTAB(4,10)"Y/N":A$=GET$
300 IFA$="Y"THEN120ELSE CLG:PRINTTAB(0,10)"SEE YO
U SOON":TIME=0:REPEAT:UNTILTIME=200:MODE6:END
310 PRINTTAB(0,2)"You have failed"
320 PRINTTAB(2,4)"miserably":GOTO270
330 PRINTTAB(0,9)"CONGRATULATIONS"
340 FORS2%=1TO3
350 FORS%=1TO12
360 SOUND1,-15,(S%*4)+200,2

```

```

370 NEXT:NEXT
380 GOTO130
390 :
400 DATA50,10,350,10,650,10,950,10
410 DATA50,260,350,260,650,260,950,260,50,560
420 DEFFNX=150-Z*30
430 DEFFNY=200-Z*40
440 :
450 DEFPROCdraw
460 GCOL0,3
470 MOVEX,Y:DRAWX,Y+200:DRAWX+150,Y+200:PLOT85,X,
Y
480 DRAWX+150,Y:DRAWX,Y:PLOT85,X+150,Y+200:IFF=1T
HENENDPROC
490 GCOL0,1:FORZ=0TO4
500 MOVEX,Y+Z*40:PLOT1,FNX,FNY
510 MOVEX+Z*30,Y:PLOT1,FNX,FNY
520 MOVEX,Y+FNY:PLOT1,FNX,-FNY
530 MOVEX+Z*30,Y+200:PLOT1,FNX,-FNY
540 NEXT:ENDPROC
550 :
560 DEFPROCset
570 FORJ=1TO12
580 A(J)=RND(52):F=0
590 IFJ=1THEN620
600 FORK=1TOJ-1:IFA(J)=A(K)THENF=1
610 NEXT
620 IFF=1THEN580
630 NEXT
640 FORJ=1TO12:FORL=1TO4
650 IFA(J)>13THENA(J)=A(J)-13:S(J)=L
660 NEXT:NEXT:ENDPROC
670 :
680 DEFPROCchange
690 PROCdelete:PRINTTAB(2,8)"Change card?";
700 PRINTTAB(6,10)"Y/N";:A$=GET$
710 IFA$="Y" OR A$="N" THEN 720 ELSE VDU7:GOTO700
720 IFA$="N"THEN770
730 IFJ=1THENA(J)=A(10):S(J)=S(10)
740 IFJ=4THENA(J)=A(11):S(J)=S(11)
750 IFJ=7THENA(J)=A(12):S(J)=S(12)

```

```

760 PROCpicture:PROCdraw:PROCshow
770 PROCdelete:ENDPROC
780 :
790 DEFPROCpicture
800 FORL=1TO9
810 B$(L)=CHR$(A(L)+49)
820 IFA(L)=9THENB$(L)=CHR$224
830 IFA(L)=10THENB$(L)="J"
840 IFA(L)=11THENB$(L)="Q"
850 IFA(L)=12THENB$(L)="K"
860 IFA(L)=13THENB$(L)="A"
870 NEXT:ENDPROC
880 :
890 DEFPROCshow
900 VDU5:IFS(J)=1ORS(J)=3THENGCOL0,1ELSEGCOL0,2
910 MOVEX+50,Y+100
920 PRINTB$(J):VDU4:ENDPROC
930 :
940 DEFPROCtitles
950 PRINTTAB(0,1)"Target=",AIM
960 PRINTTAB(0,3)"Min.Bet=",MB
970 PRINTTAB(0,5)"Credit=",C
980 PROCbestscore:ENDPROC
990 :
1000 DEFPROCbet
1010 REPEAT:PRINTTAB(1,8)"What is your"
1020 INPUTTAB(5,10)"Bet? "B:PROCdelete:VDU7:UNTILB
>=MB ANDB<=C
1030 C=C-B:PROCdelete:ENDPROC
1040 :
1050 DEFPROChigherlower
1060 PRINTTAB(0,8)"Higher or lower"
1070 REPEAT:PRINTTAB(5,10)"H/L...":A$=GET$:VDU7:UN
TILA$="H"ORA$="L"
1080 IF (A$="L"AND(A(J+1)<A(J))OR(A$="H"AND(A(J+1)>A(
J)) THENC=C+B*2:SOUND1,-10,200,2:SOUND1,-10,220,2:SO
UND1,-10,200,2:SOUND1,-10,220,2 ELSE SOUND 1,-10,20
,10:SOUND1,-10,1,10
1090 PROCdelete:PROCupdate:ENDPROC
1100 :
1110 DEFPROCdelete

```

```

1120 PRINTTAB(0,7)SPC(80):ENDPROC
1130 DEFPROCupdate
1140 PRINTTAB(8,5)SPC(8):PRINTTAB(8,5),C
1150 ENDPROC
1160 :
1170 DEFPROCbestscore
1180 PRINTTAB(1,13)"Best      Total"
1190 PRINTTAB(1,14)"Score      Score"
1200 PRINTTAB(0,15),BS,RT;
1210 ENDPROC
1220 :
1230 ON ERROR OFF
1240 IF ERR=17 A$="":GOTO 300
1250 MODE6
1260 REPORT:PRINT" at line ";ERL
1270 END

```

RETURN OF THE DIAMOND
an Adventure Game
by R. McGregor

An adventure game (in computer terms) is a game where you travel in a world described to you by the computer. The computer displays where you are, what you can see, and in which directions you can move. You communicate with the computer through a series of English words such as GET, KILL, TAKE, etc. Movement, in this particular game, can be in any one of four directions, and is accomplished by typing N, S, E or W for the four compass points.

' Return of the Diamond' is a relatively small adventure game (an unfortunate necessity for a game that has to be typed in). There are just 9 locations. It is nevertheless fun to play, and may give you a feel for adventure games (and how they are composed) before purchasing a mammoth one (or experimenting with a program of your own). The object of the game is to search for the great diamond, and return it to Diamond Castle. Of course you will meet deadly enemies in your search, and you will need to use your wits to thwart them.

```

****      by R. McGregor      ****
*****
*****
You are in a hut
Exits :
East:
You can see :
a hammer
What now?TAKE HAMMER
-----
O.K.
What now?E
-----
You are in a garden
Exits :
South:West:
You can see :
nothing.
What now?

```

As you play the game you will probably need to construct a map of the locations. In case you get stuck, a copy of the map is given at the end of this book, but without the location of the diamond.

Happy hunting.

Adventure procedures

Set-up	Set up the arrays and initialises the variables.
Title	Prints the title.
Look	Prints the current position, with room description, exits and items.
Analyse	Tests for a move, look, inventory, score, quit, or move command.
Light	Lights the lamp.
Off	Turns the lamp off.
Drop	Drops the selected item.
Kill	Tests if the player has met a gremlin or a pixie.

Other commands	This deals with the rest of the commands.
Time-passing	Decreases score, increases moves and deals with the lamp.
Move	Moves the player.
Inventory	Lists the items you are carrying.
Take	Takes selected item.
Kill-gremlin	Kills the gremlin.
Kill-pixie	Kills the pixie.
Finish	End of game.

```

10  REM Return of the Diamond
20  REM by R.McGregor
30  REM BEEBUG
40  REM VERSION P 1.0
50  :
60  MODE 6:VDU 19,0,4,0,0,0
70  PROCsetup
80  PROCtitle
90  PROClook
100 REPEAT
110 REPEAT
120 INPUT "What now",c$
130 IF LEN(c$)=0 THEN PRINT "Eh?"
140 UNTIL LEN(c$)>0
150 PRINTSTRING$(39,"-")
160 PROCanalyse
170 PROCtimepassing
180 UNTIL dead OR won
190 PROCfinish
200 END
210 :
220 DEF PROCsetup
230 DIM place$(9)
240 FOR i=1 TO 9
250 READ place$(i)

```



```

260 NEXT i
270 DATA in a hut,in a garden
280 DATA in a shrubbery,on a path
290 DATA on a lane,in a forest
300 DATA at a dead end,at diamond castle
310 DATA in a dark passage
320 DIM newpos(9,4)
330 FOR i=1 TO 9
340 FOR j=1 TO 4
350 READ newpos(i,j)
360 NEXT j
370 NEXT i
380 DATA 0,2,0,0,0,0,5,1,0,0,6,0
390 DATA 0,5,7,0,2,6,0,4,3,0,9,5
400 DATA 4,0,0,0,0,9,0,0,6,0,0,8
410 DIM item$(6),itemname$(6),itempos(6)
420 FOR i=1 TO 6
430 READ item$(i),itemname$(i),itempos(i)
440 NEXT i
450 DATA a lamp,LAMP,5
460 DATA the geat diamond,DIAMOND,7
470 DATA a sharp knife,KNIFE,3
480 DATA a hammer,HAMMER,1
490 DATA a mean looking gremlin,GREMLIN,4
500 DATA a nasty little pixie,PIXIE,9
510 DIM com$(7)
520 FOR i=1 TO 7
530 READ com$(i)
540 NEXT
550 DATA GET,TAKE,ON,LIGHT,OFF,DROP,KILL
560 DIM direct$(4)
570 FOR i=1 TO 4
580 READ direct$(i)
590 NEXT
600 DATA North,East,South,West
610 DIM bright$(2)
620 bright$(0)="( It's off )"
630 bright$(1)="( It's shining dimly )"
640 bright$(2)="( It's shining brightly )"
650 on=FALSE
660 reallit=2.9

```

```

670 lit=2
680 position=1
690 dead=FALSE
700 won=FALSE
710 moves=0
720 score=30
730 carried=0
740 ENDPROC
750 :
760 DEF PROCtitle
770 PRINT:PRINT
780 PRINT"          *****"
790 PRINT"          *****"
800 PRINT"      ****   Return Of The Diamond   ****"
810 PRINT"      ****         by R. McGregor         ****"
820 PRINT"          *****"
830 PRINT"          *****"
840 ENDPROC
850 :
860 DEF PROClook
870 IF (position=6 OR position=9) AND (NOT on O
R (itempos(1)<>position AND itempos(1)<>0)) THEN PR
INT"It is pitch dark.":ENDPROC
880 PRINT
890 PRINT"You are ";place$(position)
900 PRINT
910 PRINT"Exits : "
920 FOR i=1 TO 4
930 IF newpos(position,i)>0 THEN PRINT direct$(i
);": ";
940 NEXT i
950 PRINT
960 PRINT
970 PRINT"You can see : "
980 printed=FALSE
990 FOR i=1 TO 6
1000 IF itempos(i)=position THEN PRINT item$(i):p
rinted=TRUE
1010 IF itempos(i)=position AND i=1 AND NOT on TH
EN PRINT bright$(0)
1020 IF itempos(i)=position AND i=1 AND on THEN P

```

```

RINT bright$(lit)
1030 NEXT i
1040 IF NOT printed THEN PRINT"nothing."
1050 ENDPROC
1060 :
1070 DEF PROCanalyse
1080 IF LEN(c$)=1 THEN IF INSTR("NESW",c$)>0 THEN
PROCmove:ENDPROC
1090 IF c$="LOOK" THEN PROClook:ENDPROC
1100 IF LEFT$(c$,3)="INV" THEN PROCinventory:ENDP
ROC
1110 IF c$="SCORE" THEN PRINT"Your score is ";sco
re;".":ENDPROC
1120 IF c$="MOVES" THEN PRINT"Moves made : ";move
s:ENDPROC
1130 PROCothercommands
1140 ENDPROC
1150 :
1160 DEF PROCtimepassing
1170 score=score-1
1180 moves=moves+1
1190 dimmed=FALSE
1200 IF on THEN reallit=reallit-0.1:dimmed=TRUE
1210 lit=INT(reallit)
1220 IF dimmed AND lit=0 THEN PRINT"Your lamp jus
t went out.":on=FALSE
1230 won=(position=8 AND itempos(2)=8)
1240 ENDPROC
1250 :
1260 DEF PROCmove
1270 dir=INSTR("NESW",c$)
1280 IF newpos(position,dir)=0 THEN PRINT"You can
't move in that direction.":ENDPROC
1290 IF (position=6 OR position=9) AND (NOT on OR
(itempos(1)<>position AND itempos(1)<>0)) THEN PRI
NT"You have fallen into a snake pit!":dead=TRUE:END
PROC
1300 position=newpos(position,dir)
1310 PROClook
1320 ENDPROC
1330 :

```

```

1340 DEF PROCinventory
1350 PRINT
1360 PRINT
1370 PRINT"You are carrying : "
1380 printed=FALSE
1390 FOR i=1 TO 6
1400 IF itempos(i)=0 THEN PRINT item$(i):printed=
TRUE
1410 IF itempos(i)=0 AND i=1 AND NOT on THEN PRIN
T bright$(0)
1420 IF itempos(i)=0 AND i=1 AND on THEN PRINT br
ight$(lit)
1430 NEXT i
1440 IF NOT printed THEN PRINT"nothing."
1450 ENDPROC
1460 :
1470 DEF PROCothercommands
1480 comno=FNcommand
1490 thingno=FNthing
1500 IF comno=0 OR thingno=0 THEN PRINT"Sorry.I d
on't understand.":ENDPROC
1510 ON comno GOTO 1520,1520,1530,1530,1540,1550,
1560
1520 PROCTake:ENDPROC
1530 PROClight:ENDPROC
1540 PROCoff:ENDPROC
1550 PROCdrop:ENDPROC
1560 PROCkill:ENDPROC
1570 ENDPROC
1580 :
1590 DEF FNcommand
1600 no=0:i=0
1610 REPEAT
1620 i=i+1
1630 IF LEFT$(c$,LEN(com$(i)))=com$(i) THEN no=i
1640 UNTIL no>0 OR i=7
1650 =no
1660 :
1670 DEF FNthing
1680 no=0:i=0
1690 REPEAT

```

```

1700  i=i+1
1710  IF RIGHT$(c$,LEN(itemname$(i)))=itemname$(i)
THEN no=i
1720  UNTIL no>0 OR i=6
1730  =no
1740  :
1750  DEF PROCtake
1760  IF itempos(thingno)<>position THEN PRINT"I d
on't see that here.":ENDPROC
1770  IF thingno=5 OR thingno=6 THEN PRINT"You'll
be lucky!":ENDPROC
1780  IF carried=3 THEN PRINT"You can't carry any
more.":ENDPROC
1790  itempos(thingno)=0
1800  PRINT"O.K."
1810  carried=carried+1
1820  ENDPROC
1830  :
1840  DEF PROClight
1850  IF itempos(thingno)<>0 THEN PRINT"I would if
you had it.":ENDPROC
1860  IF thingno<>1 THEN PRINT"You're joking!":END
PROC
1870  IF on THEN PRINT"It's already on.":ENDPROC
1880  IF lit=0 THEN PRINT"It won't relight.":ENDPR
OC
1890  PRINT"O.K."
1900  on=TRUE
1910  ENDPROC
1920  :
1930  DEF PROCoff
1940  IF itempos(thingno)<>0 THEN PRINT"You're not
carrying that.":ENDPROC
1950  IF thingno<>1 THEN PRINT"Come off it!":ENDPR
OC
1960  IF NOT on THEN PRINT"It's already off.":ENDP
ROC
1970  PRINT"O.K."
1980  on=FALSE
1990  ENDPROC
2000  :

```

```

2010 DEF PROCdrop
2020 IF itempos(thingno)<>0 THEN PRINT"But you ha
ven't got that.":ENDPROC
2030 itempos(thingno)=position
2040 PRINT"O.K."
2050 carried=carried-1
2060 ENDPROC
2070 :
2080 DEF PROCkill
2090 IF itempos(thingno)<>position THEN PRINT"I d
on't see that here.":ENDPROC
2100 IF thingno=5 THEN PROCkillgremlin:ENDPROC
2110 IF thingno=6 THEN PROCkillpixie:ENDPROC
2120 PRINT"You're joking!"
2130 ENDPROC
2140 :
2150 DEF PROCkillgremlin
2160 IF itempos(3)=0 THEN PRINT"You slash your kn
ife at the gremlin and kill it easily.":itempos(5)=
-1:score=score+10:ENDPROC
2170 IF itempos(4)=0 THEN PRINT"You throw your ha
mmer at the gremlin,but it catches it and throws it
back.":ENDPROC
2180 PRINT"You fight the gremlin bare handed,but
only succeed in getting killed."
2190 dead=TRUE
2200 ENDPROC
2210 :
2220 DEF PROCkillpixie
2230 IF itempos(4)=0 THEN PRINT"You throw your ha
mmer at the pixie.....A hit!":itempos(6)=-1:score=
score+10:ENDPROC
2240 IF itempos(3)=0 THEN PRINT"You slash your kn
ife at the pixie but it dodges.":ENDPROC
2250 PRINT"You fight the pixie bare handed but
it is stronger than you thought.You get killed."
2260 dead=TRUE
2270 ENDPROC
2280 :
2290 DEF PROCfinish
2300 PRINT

```

```

2310 PRINT
2320 IF won THEN PRINT"      Congratulations!!!---
You won!!!"
2330 IF dead THEN PRINT"      Bad Luck!!!---You los
t!!!":score=0
2340 PRINT
2350 PRINT
2360 PRINT"      You took ";moves;" moves,"
2370 PRINT"      and your final score was ";score;".
"
2380 ENDPROC

```

FIVE-DICE
by B. J. Kilby

FIVE DICE						
VALUE		CATEGORY			POINTS	
dice*1		1	ONES			2
dice*2		2	TWOES			4
dice*3		3	THREES			3
dice*4		4	FOURS			0
dice*5		5	FIVES			
dice*6		6	SIXES			
If 1-6 total over 62 then BONUS						
add dice		7	3 OF A KIND			
add dice		8	4 OF A KIND			
25		9	FULL HOUSE			
30		10	4 STRAIGHT			
40		11	5 STRAIGHT			
50		12	FIVE DICE			
add dice		13	CHANCE			
						TOTAL
No.	1	2	3	4	5	
						ROLL (Y/N)?
	1	5	3	3	4	

This program simulates the dice game ' Yahtzee®, in which five dice are thrown and a score is given for their value, depending on the category in which the player puts the throw. The game requires a fair amount of skill and judgement, although luck inevitably plays a part.

You are allowed a maximum of three re-throws on

any number of dice, and then you must place the result in one of the categories listed.

Full instructions can be called when the program is first run (the instructions appear in a novel way), and a high score is kept. If you are unfamiliar with the game, do not be daunted by the rules, these are soon picked up.

(NOTE: when playing, if you get 5 numbers of the same value e.g. 66666 this CANNOT be put down as a full house.)

Program Notes

This is a well structured game. The main game loop appears in lines 60 to 190 and this code consists largely of procedure calls. The rest of the program consists of procedure and function definitions. Here is a brief summary of their function.

Procedures and functions

Init	Initialises variables, clears screen.
Screen	Prints the title and categories.
Roll	Throws dice and allows two more goes, calls FNchoice, calls test subroutine, and awards points.
FNtest1 (number%)	Checks if dice (1-5)=number% and returns value of quantity* number%, tests for categories 1-6.
FNtest11	Sorts dice into numerical order and checks if either 12345 or 23456 exist.
FNtest10	Checks if dice equal 1234, 2345 or 3456, other dice can be any value.
FNtest9	Checks for a ' Full House' (doesn't allow five of a kind).
FNtest7 (number%)	Tests for 3,4, or 5 of a kind.
Sort	' Bubble' sorts the dice into numerical order.

FNadd	Returns the total value of the dice.
Throw(dice\$)	Assigns new value to dice chosen.
Clear	Clears the screen.
Bonus	If categories 1-6 add up to more than 62, then a bonus of 30 is added to the scoresheet.
Choice	Main part. Asks if another roll is required, which category to score, and returns value of category.
Dice	Prints dice.
End	Prints total score and best score.
Wait	Waits for the space-bar to be pressed.
Read(N)	Reads N lots of data, and prints it at slow speed for the instructions.

```

10 REM Five Dice
20 REM by B.Kilby
30 REM BEEBUG
40 REM VERSION P 1.0
50 :
60 MODE6
70 VDU19,0,4,0,0,0
80 ON ERROR GOTO 1890
90 B%=RND(-TIME):B%=0
100 PRINTTAB(16,2);"FIVE DICE";TAB(16,4);"by B Ki
lby"
110 IF FNyes("DO YOU WANT THE RULES ?",6,12)THEN
PROCread(5):PROCread(10)
120 PROCinit
130 PROCscreen
140 REPEAT
150 PROCroll
160 UNTIL cat%=13
170 PROCend
180 IF FNyes("New Game",28,22) THEN CLEAR:GOTO 12

```

```

190 MODE6:END
200 :
210 REM INITIALISE
220 DEF PROCinit
230 DIM dice%(5),cat%(13)
240 VDU23,1,0;0;0;0;
250 VDU15,12
260 cat%=0:call%=0:bonus%=0
270 ENDPROC
280 :
290 REM SCREEN
300 DEF PROCscreen
310 PRINTTAB(15,1)"FIVE DICE"
320 IF B%>0 THEN PRINTTAB(15,3);"Best  ";B%
330 PRINTTAB(0,4)"  VALUE"TAB(16)"CATEGORY"TAB(32)
"POINTS" '
340 FORI%=1TO6:PRINT"dice*";I%;TAB(16);I%;NEXT
350 PRINTTAB(18,6)"ONES"
360 PRINTTAB(18,7)"TWOS"
370 PRINTTAB(18,8)"THREES"
380 PRINTTAB(18,9)"FOURS"
390 PRINTTAB(18,10)"FIVES"
400 PRINTTAB(18,11)"SIXES"
410 PRINT" If 1-6 total over 62 then BONUS"
420 FORI%=3TO4:PRINT"add dice"TAB(18);I%;" OF A K
IND":NEXT
430 PRINT;25TAB(18)"FULL HOUSE"
440 PRINT;30TAB(18)"4  STRAIGHT"
450 PRINT;40TAB(18)"5  STRAIGHT"
460 PRINT;50TAB(18)"FIVE DICE"
470 PRINT"add dice"TAB(18)"CHANCE"
480 PRINTTAB(27);"TOTAL"
490 FORI%=13TO19:PRINTTAB(15,I%);I%-6:NEXT
500 PRINTTAB(0,21);"No.  1  2  3  4  5";
510 ENDPROC
520 :
530 REM MAIN SECTION
540 DEF PROCroll
550 go%=0
560 PROCthrow("12345")
570 Q=FNchoice

```

```

580 ON Q GOSUB 620,620,620,620,620,620,660,660,67
0,680,690,700,710
590 gt%=gt%+points%
600 PRINTTAB(35,5+Q DIV7+Q);points%
610 ENDPROC
620 points%=FNtest1(Q):call%=call%+1
630 bonus%=bonus%+points%
640 IF call%=6 THEN PROCbonus
650 RETURN
660 points%=ABS(FNtest7(Q-4)*FNadd):RETURN
670 points%=FNtest9:RETURN
680 points%=FNtest10:RETURN
690 points%=FNtest11:RETURN
700 points%=ABS(FNtest7(Q-7)*50):RETURN
710 points%=FNadd:RETURN
720 :
730 REM CAT. TESTS
740 REM ONES to SIXES
750 DEF FNtest1(number%)
760 LOCAL total%,I%
770 total%=0
780 FORI%=1TO5:total%=total%+(dice%(I%)=number%):
NEXT
790 =ABS(total%)*number%
800 :
810 REM 5 STRAIGHT
820 DEF FNtest11
830 LOCAL test$,I%
840 PROCsort
850 test$=""
860 FORI%=1TO5:test$=test$+STR$(dice%(I%)):NEXT
870 IF test$="12345"OR test$="23456"THEN =40 ELSE
=0
880 :
890 REM 4 STRAIGHT
900 DEF FNtest10
910 IF (FNtest1(3)>=3 AND FNtest1(4)>=4)AND((FNte
st1(2)>=2 AND(FNtest1(1)>=1 OR FNtest1(5)>=5))OR(FN
test1(5)>=5 AND FNtest1(6)>=6)) THEN =30 ELSE =0
920 :
930 REM FULL HOUSE

```

```

940 DEF FNtest9
950 PROCsort
960 LOCAL flag%
970 flag%=((dice%(1)=dice%(2))AND(dice%(2)=dice%(
3)))AND(dice%(4)=dice%(5))OR((dice%(1)=dice%(2))AND
(dice%(3)=dice%(4))AND(dice%(4)=dice%(5)))
980 IF flag%<>FNtest7(5) THEN =25 ELSE =0
990 :
1000 REM 3-4-5 OF A KIND
1010 DEF FNtest7(number%)
1020 LOCAL I%,X%,total%
1030 FORI%=1TO6
1040 total%=0
1050 FORX%=1TO5
1060 IF dice%(X%)=I% THEN total%=total%+1
1070 NEXT
1080 IF total%>=number% THEN I%=7:=TRUE
1090 NEXT:=FALSE
1100 :
1110 REM ADD DICE
1120 DEF FNadd
1130 LOCAL I%,total%
1140 FORI%=1TO5:total%=total%+dice%(I%):NEXT:=tota
l%
1150 REM SORT
1160 DEF PROCsort
1170 LOCAL Y%,Z%,T%
1180 FORY%=1TO4
1190 FORZ%=Y%+1TO5
1200 IF dice%(Y%)>dice%(Z%) T%=dice%(Y%):dice%(Y%)
=dice%(Z%):dice%(Z%)=T%
1210 NEXT:NEXT:ENDPROC
1220 :
1230 REM RE-THROW DICE
1240 DEF PROCthrow(dice$)
1250 LOCAL I%
1260 FORI%=1TO LEN(dice$)
1270 flag%=VAL(MID$(dice$,I%,1)):IF flag%>5 THEN f
lag%=0:I%=6:ENDPROC
1280 dice%(flag%)=RND(6)
1290 NEXT

```

```

1300 PROCdice:ENDPROC
1310 :
1320 REM CLEAR MESSAGE
1330 DEF PROCclear
1340 PRINTTAB(27,22);SPC(12)
1350 PRINTTAB(28,23)SPC(10)
1360 ENDPROC
1370 :
1380 REM BONUS
1390 DEF PROCbonus
1400 IF bonus%>62 THEN bonus%=30:gt%=gt%+30 ELSE b
onus%=0
1410 PRINTTAB(35,12);bonus%
1420 ENDPROC
1430 :
1440 REM YES OR NO
1450 DEF FNyes(Q$,X%,Y%)
1460 LOCAL reply$
1470 PRINTTAB(X%,Y%);Q$;
1480 REPEAT:reply$=CHR$(GET AND &DF):UNTIL reply$=
"Y" OR reply$="N"
1490 PROCclear
1500 =(reply$="Y")
1510 :
1520 REM CHOICES
1530 DEF FNchoice
1540 REPEAT:go%=go%+1
1550 SOUND1,-12,200,5
1560 IF NOT FNyes("ROLL (Y/N)?",28,22) THEN go%=2:
GOTO 1660
1570 REPEAT
1580 REPEAT
1590 SOUND1,-10,150,5
1600 PRINTTAB(27,22);"WHICH DICE?"
1610 INPUTTAB(28,23)Q$
1620 PROCclear
1630 UNTIL Q$>CHR$47 AND Q$<CHR$54
1640 PROCthrow(Q$)
1650 UNTIL flag%
1660 UNTIL go%=2
1670 REPEAT

```

```

1680 REPEAT
1690 SOUND1,-10,100,5
1700 PRINTTAB(28,22)"CATEGORY?"
1710 INPUTTAB(28,23)"Q
1720 PROCclear
1730 UNTIL Q>0 AND Q<14
1740 UNTIL cat%(Q)=0
1750 cat%(Q)=1:cat%=cat%+1:=Q
1760 :
1770 REM PRINT DICE
1780 DEF PROCdice
1790 PRINTTAB(5,23);:FORN%=1TO5:PRINT;dice%(N%)SPC
2;:NEXT
1800 TIME=0:REPEATUNTIL TIME>50
1810 ENDPROC
1820 :
1830 REM END RESULT
1840 DEF PROCend
1850 PRINTTAB(35,20);gt%;
1860 IF gt%>B% THEN B%=gt%:PRINTTAB(15,3);"Best  "
;B%
1870 ENDPROC
1880 :
1890 REM ERROR TRAP
1900 ON ERROR OFF
1910 MODE6:REPORT:PRINT" at line ";ERL
1920 END
1930 REM RULES
1940 :
1950 DATA"In the following game the object is to
score as high as possible, by skilfully placing res
ults obtained from the dice into the various categ
ories."
1960 DATA"There are FIVE DICE."
1970 DATA"You are allowed a maximum of three goes
for each category (although you can stopafter the f
irst or second roll)."
1980 DATA"You may re-throw ANY or ALL the dice as
you see fit, by typing out the relevant dice number
s (1 to 5 in any order)."
1990 DATA"When you are satisfied with the results

```

(or after the third throw whichever is first) you MUST place them in one of the 13 categories."

2000 DATA" CATEGORIES:"

2010 DATA" ONES to SIXES :every dice that has the same value as the cat. chosen is added together eg .11223=4(2+2) in TWOS"

2020 DATA" 3 OF A KIND :any 3 dice the same score value of all 5 dice"

2030 DATA" 4 OF A KIND :any 4 dice the same score value of all 5 dice"

2040 DATA" FULL HOUSE :3 of one number AND 2 of another scores 25 points"

2050 DATA" 4 STRAIGHT :either 1234 or 2345 or 3456 score 30 points (any order)"

2060 DATA" 5 STRAIGHT :either 12345 or 23456 scores 40 points (any order)"

2070 DATA" FIVE DICE :all the same ? score 50 points"

2080 DATA" CHANCE :anything goes here so score total value of dice."

2090 DATA" When asked 'CATEGORY?' enter the NUMBER of the category"

2100 REM WAIT

2110 DEF PROCwait

2120 PRINT'"press space bar."

2130 REPEAT UNTIL GET=32

2140 CLS:ENDPROC

2150 REM SLOW PRINTOUT

2160 DEF PROCread(N)

2170 LOCAL T,letter,word,word\$

2180 VDU12,10

2190 FORword=1 TO N

2200 READ word\$

2210 FORletter=1 TO LEN(word\$)

2220 PRINTMID\$(word\$,letter,1);

2230 TIME=0:REPEAT UNTIL TIME>2

2240 NEXT

2250 TIME=0:REPEAT UNTIL TIME>250

2260 PRINT:NEXT

2270 PROCwait

2280 ENDPROC

ANAGRAMS

by Mike Case



'Anagrams' requires a degree of thought on the part of the player — and, moreover, it has no 'firebutton'! The player is presented with anagrams which he must solve, each of which has at least one correct solution.

The list of words used is held in DATA statements from line 1580 onwards. The words are terminated by 999 to mark the end of the list, and new words can be inserted in fresh data statements, providing you make sure that 999 is the last item of data. The maximum number of words currently allowed is 200, set at line 70. Change vocab\$(200) to a larger value if required.

To make the game suitable for children, either remove the longer words and replace them with simpler ones, or alter the section at line 250 to filter out long words, e.g. insert 285 IF length>4 THEN GOTO 250 (you can also filter out the shorter words by a similar process). Note that some of the anagrams displayed may have more than one solution. The program will only accept ONE of these. For this reason we have edited out a considerable number of the 3 and 4 letter words initially supplied.

This program has been used with some success teaching two- and three-year-olds letter and number recognition. For this purpose, replace all the data statements in lines 1580 to 1720 with the sequence of letters or numbers from which you wish selection to be made, e.g.

1590 DATA A,B,C,D,E,F,G,H,I,J,K,L,M

1600 DATA O,P,Q,R,S,T,U,V,W,X,Y,Z,999

```

10  REM Anagrams
20  REM by Mike Case
30  REM BEEBUG
40  REM VERSION P 1.0
50  :
60  ON ERROR GOTO 1750
70  MODE 1
80  DIM CL$(9), vocab$(200), guess$(20), jumbled$(2
0)
90  REM DISABLE CURSOR
100 VDU23,1,0;0;0;0;
110 W%=0:I%=0
120 CLS
130 PROCTITLE
140 J%=INKEY(500)
150 CLS:COLOUR1
160 PRINTTAB(0,2);"SCORE IS ";W%;" OUT OF ";I%;S
PC(2);'"
170  REM Set random number generator to random st
art
180  X=RND(-TIME)
190  RESTORE 1590
200  REM READ IN THE WORDS
210  vocab=0
220  REPEAT vocab=vocab+1
230  READ vocab$(vocab)
240  UNTIL vocab$(vocab)="999":NUM=vocab-1
250  REM Pick a random word
260  random=RND(NUM)
270  phrase$=vocab$(random)
280  length=LEN(phrase$)

```

```

290  guess=1
300  right=0:wrong=0
310  PROCSHUFFLE:COLOUR2
320  PRINT""The word you have to find is ";length
h
330  PRINT"letters long, you have to guess each"
""letter in turn, ";
340  PRINT"starting at the""first one."
350  COLOUR 3
360  PRINT""PRESS THE SPACE BAR TO START";
370  REPEAT UNTIL GET=32
380  CLS
390  COLOUR2:PRINTTAB(16);"Length = ";length;
400  PRINTTAB(5,5)"Here is the jumbled up phrase"
410  COLOUR1:PRINTTAB(10,10);shuffled$
420  COLOUR3:PRINTTAB(0,16)"Now enter your guess
or letter no. ";guess
430  guess$(guess)=GET$
440  REM to prevent guess of letter not in word
450  PROCSILLY
460  IF silly=0 THEN GOTO430
470  PROCHECK
480  IF check=0 THEN PROCNOISE:wrong=wrong+1
490  IF check=1 THEN PROCYES:guess=guess+1
500  REM Set number of wrong guesses allowed to I
NT(half total letters)
510  IF (INT(length/2))-wrong=1 THEN PRINTTAB(0,2
0);"THIS IS YOUR VERY LAST CHANCE";SPC(40):GOTO350
520  PRINTTAB(0,20);"YOU HAVE ";(INT(length/2))-w
rong;" WRONG GUESSES LEFT"
530  IF check=1 THEN right=right+1
540  end$=""
550  IF wrong=INT(length/2) THEN I=INKEY(200):PRO
CANSWER
560  IF right=length THEN PROCWIN
570  IF end$="" THEN 420
580  IF end$="N" OR end$="n" THEN MODE6:PRINTTAB(
8,8);"B Y E - B Y E ":END
590  IF end$="Y" OR end$="y" THEN GOTO150
600  :
610  DEFPROCHECK

```

```

620  check=0
630  IF guess$(guess)=MID$(phrase$,guess,1) THEN
check=1
640  IF check=1 AND guess=1 THEN PRINTTAB(1,13);S
PC(length+5);
650  IF check=1 THEN PRINTTAB(3,13);LEFT$(phrase$
,guess):PRINTTAB(4,9)SPC(length+3):PROCDELETE
660  IF check=0 THEN PROCUNDERLINE
670  ENDPROC
680  :
690  DEFPROCNOISE
700  SOUND 3,-15,100,5
710  SOUND 2,-15,103,5
720  SOUND 0,-12,20,5
730  ENDPROC
740  :
750  DEFPROCYES
760  ENVELOPE 1,1,2,-2,5,30,25,15,126,0,0,-126,12
6,126
770  SOUND 1,1,80,5
780  SOUND 1,1,100,5
790  ENDPROC
800  :
810  DEFPROCANSWER
820  I%=I%+length:W%=W%+right
830  shuffled$=""
840  FOR times=1 TO length
850  shuffled$=shuffled$+jumbled$(times)
860  NEXTtimes
870  CLS
880  PRINTTAB(0,2)"The jumbled phrase  was:-"
890  PRINTTAB(5,5);shuffled$
900  PRINTTAB(0,7)"The original phrase was :-"
910  PRINTTAB(5,10);phrase$
920  PRINT' 'SPC(4);"Another go                (Y or N)
?"
930  end$=GET$
940  IF end$<>"y"AND end$<>"Y" AND end$<>"n"AND e
nd$<>"N" THEN 930
950  ENDPROC
960  :

```

```

970  DEFPROCTITLE
980  FORJ%=3 TO 23 STEP 4
990  FORK%=2 TO 22 STEP 20
1000 COLOUR RND(3)
1010 PRINTTAB(K%,J%);"A N A G R A M S"
1020 NEXT: NEXT
1030 ENDPROC
1040 :
1050 DEFPROC SHUFFLE
1060 IF LEFT$(phrase$,1)=CHR$32 THEN phrase$=RIGHT
T$(phrase$, (LENphrase$-1)): length=length-1
1070 IF LEFT$(phrase$,1)=CHR$32 THEN 1060
1080 space=0
1090 new$=phrase$
1100 FOR times=1 TO length
1110 IF times=length X=1: GOTO 1130
1120 X=RND(LEN(new$))
1130 jumbled$(times)=MID$(new$,X,1)
1140 new$=LEFT$(new$, (X-1))+RIGHT$(new$, (LEN new$
-X))
1150 NEXT times
1160 shuffled$=""
1170 FOR times=1 TO length
1180 shuffled$=shuffled$+jumbled$(times)
1190 IF jumbled$(times)=CHR$32 THEN space=space+
1
1200 NEXT times
1210 ENDPROC
1220 :
1230 DEFPROC WIN
1240 PROCYES: PROCYES
1250 W%=W%+right: I%=I%+length
1260 I=INKEY(200): CLS
1270 IF wrong>1 THEN PRINTTAB(0,5);"YOU WIN WITH
ONLY ";wrong;" WRONG GUESSES"
1280 IF wrong=1 THEN PRINTTAB(0,5);"YOU WIN IN ON
LY ONE WRONG GUESS"
1290 IF wrong=0 THEN PRINTTAB(0,5);"WELL DONE !!
NO WRONG GUESSES"
1300 PRINT''' "DO YOU WANT ANOTHER GO ?"
1310 end$=GET$

```

```

1320 IF end$<>"Y"AND end$<>"N" AND end$<>"n" AND
end$<>"y" THEN 1310
1330 ENDPROC
1340 :
1350 DEFPROCUNDERLINE
1360 find=0
1370 REPEAT find=find+1
1380 UNTIL guess$(guess)=MID$(shuffled$,find,1)
1390 PRINTTAB(4,9);SPC(20)
1400 PRINTTAB((9+find),11);CHR$94
1410 ENDPROC
1420 :
1430 DEFPROCDELETE
1440 find=0
1450 REPEAT find=find+1
1460 UNTIL guess$(guess)=MID$(shuffled$,find,1)
1470 shuffled$=LEFT$(shuffled$,find-1)+" "+RIGHT$
(shuffled$, (LENshuffled$-find))
1480 COLOUR1:PRINTTAB(0,10);shuffled$;
1490 ENDPROC
1500 :
1510 DEFPROCSILLY
1520 silly=0
1530 FOR find=1 TO length
1540 IF guess$(guess)=MID$(shuffled$,find,1) THEN
silly=1
1550 NEXTfind
1560 ENDPROC
1570 :
1580 REM Vocabulary
1590 DATA HOUSE,COMPLIANT,OFFENCE,PROTRUDE,PROPOR
TIONS
1600 DATA ACCESSION,ADJACENT,ADEQUATE,ADHESIVE,BE
NEVOLENT,CONVERT,GRAIN
1610 DATA GROOM,PESSIMISM,PERFECT,HOLD,BRACELET,C
ONVENIENCE
1620 DATA DATE,CREOSOTE,CRATER,CRAVE,CRAB
1630 DATA TREASURE,TRADE,DART,CORNET,CRANE,CARROT
,LAPEL,CRADLE
1640 DATA CONTAIN,TRIP,PLANE,KNEEL,CABINET
1650 DATA DIVIDE,DRIED,IDEA

```

```

1660 DATA PARTITION, AVOIDANCE, ENDURANCE, ELEPHANT,
SLIPPER, SLIDE
1670 DATA GRAPES, PRIDE, BOTHER, BRAIN, BRAID, BEARD
1680 DATA AGGRIEVED, VERTIGO, ELEVATOR, EXCAVATOR, AV
IATOR, UMBRELLA, UNDER
1690 DATA SUGAR, STARVE, STERN, IF, IMAGE, INFECT
1700 DATA UNDERSTAND, LOVERS, SLAM, APPLICATION, EVEN
TUALITY
1710 DATA QUEEN, HOT, FEAR, PRAM, REAM, CROCODILE, APPL
E
1720 DATA EXACERBATE, EXCOMMUNICATION, DRONE, GALE, F
LAG
1730 DATA HEAP, EPIC, THING, 999
1740 :
1750 ON ERROR OFF
1760 MODE 6
1770 IF ERR=17 END
1780 REPORT:PRINT" at line ";ERL
1790 END

```

LIFE

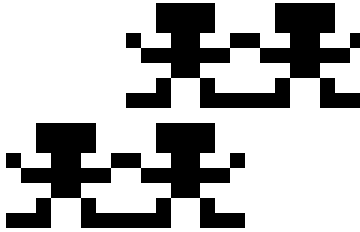
Program by Jack Hardie
Text by Sheridan Williams

This uses Assembler routines to give reasonable speed, and the screen presentation is good. It also has an effective start-up routine. On-screen instructions may be called up with a help comand.

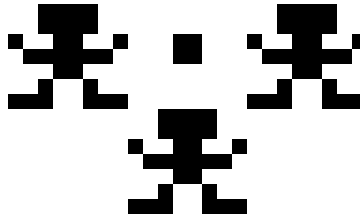
Life is a game that was developed by Professor John Conway at the University of Cambridge. As far as I am aware it was first published in the 'Mathematical Games' feature in 'Scientific American' magazine by Martin Gardner in October 1970.

The game is played on a grid, where each cell may be alive or dead. Each cell has 8 neighbours. The population of the cells changes by a set of predetermined rules which are as follows:

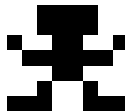
1. If a live cell has 2 or 3 live neighbours it will live on to the next generation.



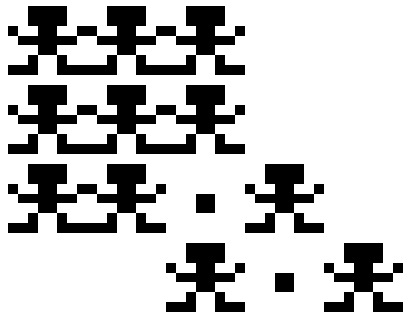
2. If an empty (dead) cell has EXACTLY 3 neighbours the cell will be born in the next generation.



3. If a cell has either none or 1 neighbour it will die of loneliness.



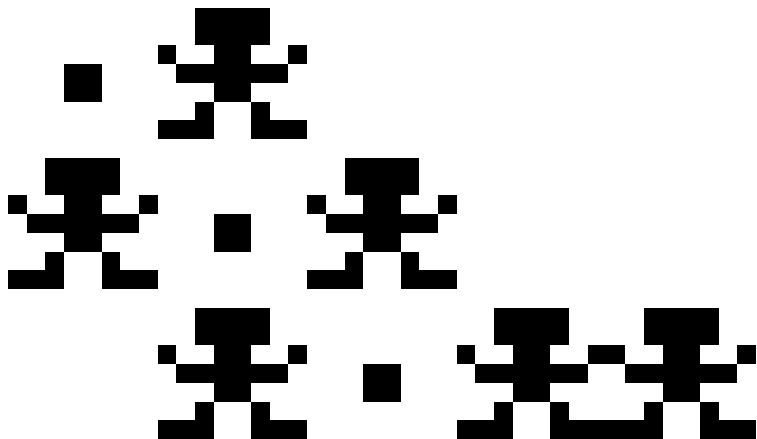
4. If a cell has 4 or more neighbours it will die from overcrowding.



The study of the original ' Life'game has become a cult and there are numerous references to it in all sorts of magazines. Here are some references for further reading:

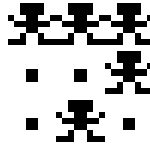
BYTE magazine Sept 75, Oct 75, Dec 75, Jan 76, Dec 78, Jan 79;

SCIENTIFIC AMERICAN Oct 70, Nov 70, Jan 71, Feb 71, Mar 71, Apr 71, Nov 71, Jan 82. There must be other references too in British magazines.

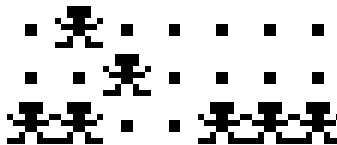


There is even a set of jargon built up around LIFE. For example various shapes can be divided into classes like Constellation, Still Life, Oscillator, Methuselah, Spaceship, Uniform Propagators, Eaters etc.

Here are two examples, the first is of a Spaceship called a ' glider' the second is one of the record holders for a long lived "Methuselah" system:



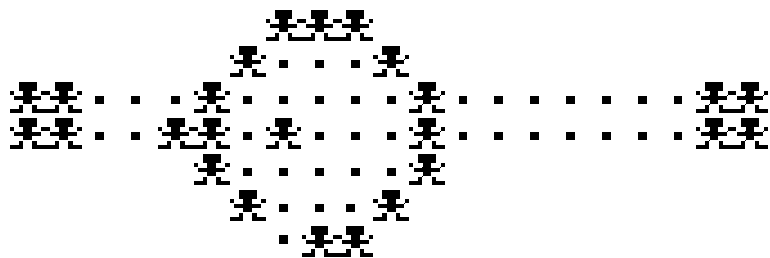
GLIDER



METHUSELAH

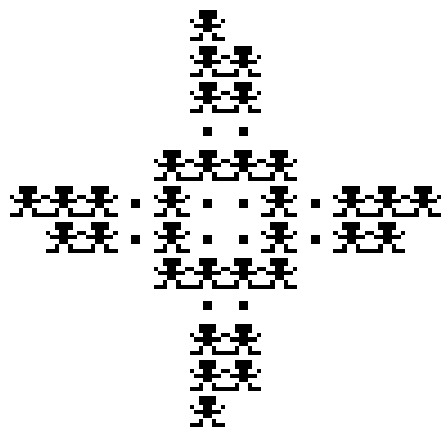
The glider will move slowly across the screen. The Methuselah will just expand, for ever?

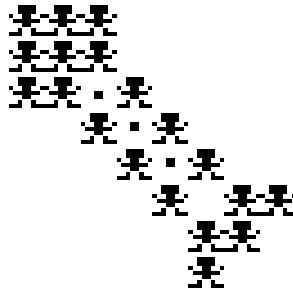
There are also ' puffertrains' , ' gliderguns' and many others. Try and invent some of your own. As a further example here is a basic shuttle. The four blobs at either end are ' eaters' and are used to clear up the debris. The central shape will change, but it will move regularly between the left and right ' eaters' for ever.



Basic Shuttle

When the BBC microcomputer version of this program was first published in BEEBUG magazine, we ran a competition for the most interesting start patterns. The two winning entries are given below. They give good sequences, and you may like to try running these shapes.





The particular implementation of 'Lifelisted here combines Basic and Assembler in an intelligent way. The most time-critical operations are in Assembler, and the other parts are in Basic for ease of writing. The program also makes good use of procedures. Note how the three main Assembler routines are used with the statements: CALL zero, CALL breed and CALL screen in lines 160, 230 and 250 respectively.

Be very careful when typing in lines 800-2420; and make sure that you save a copy before actually running the program because an error in the Assembler part can disable the system. Note that all text to the right of the slash character (\) are comments, and may omitted from the program.

```

10 REM Life
20 REM by J.Hardie
30 REM BEEBUG
40 REM VERSION P 1.0
50 :
60 ON ERROR GOTO 3500
70 MODE 4
80 HIMEM=HIMEM-&500
90 DIM note(8)
100 PROCASSEMBLE
110 PROCVDU
120 PROCLOGO
130 REPEAT

```

THOUGHT GAMES

```

140 PROCVDU
150 gen=-1
160 CALL zero
170 PROCPATTERN
180 VDU 23,1,0;0;0;0;:REM cursor off
190 CLS
200 PRINT TAB(19,31);"Q'-quit, 'R'-repeat";TAB(1
,31);"Generation ";
210 REPEAT
220 gen=gen+1
230 CALL breed
240 PRINT TAB(12,31);gen;
250 CALL screen
260 key=INKEY(0) AND &DF
270 UNTIL key=&51 OR key=&52
280 VDU22,4:REM Enable cursor
290 UNTIL key=&51
300 MODE 6
310 PRINT TAB(15,10);"Thats all."
320 END
330 :
340 DEF PROCVDU
350 VDU19,0,4,0,0,0:REM Blue ground
360 VDU19,1,7,0,0,0:REM White f'ground
370 VDU 23,224,8,&5D,&3E,&1C,&49,&55,&22,0:REM fr
og character
380 ENDPROC
390 :
400 DEF PROCPATTERN:REM Enter pattern.
410 LOCAL key
420 REM Disable editing
430 *FX4,1
440 CLS:PROCPROMPTS
450 REPEAT
460 key=GET
470 IF key<&8C AND key>&87 THEN PROCMOVE(key)
480 IF key=&7F THEN PROCUNFROG
490 IF key=&20 THEN PROCFROG
500 IF (key AND &DF)=&48 THEN PROCHELP:CALL zero:
PROCPROMPTS
510 UNTIL key=&0D

```

```

520 REM Enable editing.
530 *FX4,0
540 ENDPROC
550 :
560 DEF PROCMOVE(k)
570 VDU(k AND &F)
580 ENDPROC
590 :
600 DEF PROCFROG
610 ?(topleft-POS-40*VPOS)=1
620 VDU 224,8
630 ENDPROC
640 :
650 DEF PROCUNFROG
660 VDU 9,&7F
670 ?(topleft-POS-40*VPOS)=0
680 ENDPROC
690 :
700 DEF PROCASSEMBLE
710 oswrch=&FFEE
720 pointer=&80
730 count=&82
740 topleft=HIMEM+&500-40
750 topleft=topleft AND &FF
760 tophi=(topleft AND &FF00)/&FF
770 DIM Q% 250
780 FOR pass=0 TO 2 STEP 2
790 P%=Q%
800 [OPTpass
810 .screen
820 \ Print LIFE pattern on screen
830 \ and shift each memory loc'n
840 \ right so that next generation
850 \ becomes bit zero.
860
870 \ Memory is read downwards from
880 \ 'topleft' down to first page
890 \ boundary, then in 4 blocks of
900 \ one page each.
910 :
920 LDA #30 \ Cursor home

```

```

930 JSR oswrch
940 LDA #0
950 STA pointer
960 LDA #tophi
970 STA pointer+1
980 LDX #5           \ Block number
990 LDY #toplo
1000 .nextchar
1010 LDA (pointer),Y
1020 LSR A
1030 STA (pointer),Y
1040 JSR writechar
1050 DEY
1060 BNE nextchar
1070 LDA (pointer),Y \ Y=0 is a
1080 LSR A           \ special case
1090 STA (pointer),Y
1100 JSR writechar
1110 DEY
1120 .nextblock
1130 DEC pointer+1   \ Block ptr
1140 DEX             \ Block count.
1150 BNE nextchar
1160 LDA #30         \ Cursor home.
1170 JSR oswrch
1180 RTS
1190 :
1200 .writechar
1210 BCS frog
1220 LDA #&20        \ ASCII space.
1230 JSR oswrch
1240 RTS
1250 .frog
1260 LDA #224        \ ASCII frog.
1270 JSR oswrch
1280 RTS
1290 :
1300 .breed
1310 \ Apply LIFE rules, starting
1320 \ at screen row 3, column 2.
1330 \ Each of 8 neighbours is

```

```

1340 \ located by an offset from
1350 \ south-east neighbour.
1360 :
1370 CLD
1380 LDA #toplo-82      \ Start loc'n.
1390 STA pointer
1400 LDA #tophi
1410 STA pointer+1
1420 LDX #5             \ 5 blocks
1430 .nextcell
1440 JSR counter        \ Count n'bours
1450 JSR check          \ Alive ?
1460 DEC pointer
1470 BNE nextcell
1480 JSR counter        \ Zero case.
1490 JSR check
1500 DEC pointer
1510 DEC pointer+1     \ Next block.
1520 DEX
1530 BNE nextcell
1540 RTS
1550 :
1560 .counter
1570 CLC
1580 LDY #82           \ Offset for
1590 :
1600 LDA #1
1610 AND (pointer),Y
1620 STA count
1630 DEY              \ North cell.
1640 LDA #1
1650 AND (pointer),Y
1660 ADC count
1670 STA count
1680 DEY              \ N-East cell.
1690 LDA #1
1700 AND (pointer),Y
1710 ADC count
1720 STA count
1730 LDY #42          \ West cell
1740 LDA #1

```

THOUGHT GAMES

```

1750 AND (pointer),Y
1760 ADC count
1770 STA count
1780 DEY:DEY          \ East cell.
1790 LDA #1
1800 AND (pointer),Y
1810 ADC count
1820 STA count
1830 LDY #2          \ S-West cell.
1840 LDA #1
1850 AND (pointer),Y
1860 ADC count
1870 STA count
1880 DEY             \ South cell.
1890 LDA #1
1900 AND (pointer),Y
1910 ADC count
1920 STA count
1930 DEY             \ S-East cell.
1940 LDA #1
1950 AND (pointer),Y
1960 ADC count
1970 STA count
1980 RTS
1990 :
2000 .check          \ Alive or dead?
2010 LDY #41         \ Offset for
2020 :
2030 LDA #3
2040 CMP count
2050 BEQ live        \ 3 neighbours.
2060 LDA #2
2070 CMP count
2080 BEQ same        \ 2 neighbours.
2090 .dead           \ NOT 2 OR 3.
2100 LDA #1
2110 AND (pointer),Y
2120 STA (pointer),Y
2130 RTS
2140 .same LDA #1
2150 AND (pointer),Y \ Test bit 0.

```



```

2160 BEQ dead
2170 .live
2180 LDA #2
2190 ORA (pointer),Y \ Reset bit 1.
2200 STA (pointer),Y
2210 RTS
2220 :
2230 .zero
2240 \ Routine to set memory buffer
2250 \ to zero.
2260 LDA #tophi
2270 STA pointer+1
2280 LDA #0
2290 STA pointer
2300 LDY #0
2310 LDX #5
2320 .nextzero
2330 STA (pointer),Y
2340 DEY
2350 BNE nextzero
2360 STA (pointer),Y
2370 DEY
2380 DEC pointer+1
2390 DEX
2400 BNE nextzero
2410 RTS
2420 ]
2430 NEXT pass
2440 ENDPROC
2450 :
2460 DEF PROCHELP
2470 LOCAL key
2480 REM Disable auto key repeat.
2490 *FX11,0
2500 CLS
2510 PRINT"                LIFE"
2520 PRINT"                ----"
2530 PRINT
2540 PRINT"  The world of LIFE is populated with "
2550 PRINT"small frog-like creatures which only"
2560 PRINT"show up on the VDU screen if they are"

```

```

2570 PRINT "'alive'. Each position on the screen"
2580 PRINT "lives or dies according to how many "
2590 PRINT "live squares are next to it, so if you"
2600 PRINT "start with a pattern of live creatures"
2610 PRINT "the pattern will (usually) change after
"
2620 PRINT "each 'generation' and the pattern of"
2630 PRINT "life will 'evolve'."
2640 PRINT
2650 PRINT "Here are the rules of life, they apply"
2660 PRINT "to each character position on"
2670 PRINT "screen:"
2680 PRINT
2690 PRINT "    NUMBER"
2700 PRINT "        of", "                RESULT                "
2710 PRINT "NEIGHBOURS"
2720 PRINT STRING$(39, CHR$95)
2730 PRINT
2740 PRINT "    0 or 1            Death from loneliness"
2750 PRINT "        2            No change"
2760 PRINT "        3            Breed a new frog"
2770 PRINT "    4 to 8            Death from overcrowding"
2780 PRINT
2790 PRINT " Press spacebar to continue"
2800 PROCWAIT(200)
2810 REM Flush input buffer
2820 *FX15,1
2830 key=GET
2840 CLS:PRINT "                ENTERING A PATTERN"
2850 PRINT TAB(11); STRING$(18, CHR$95)
2860 PRINT
2870 PRINT " You must enter a pattern of live frogs
"
2880 PRINT "on the next screen."
2890 PRINT:PRINT
2900 PRINT " To produce a frog press the spacebar"
2910 PRINT "then move the cursor using the arrow"
2920 PRINT "keys and press the spacebar again "
2930 PRINT "for your next frog. To delete a "
2940 PRINT "frog use the 'DELETE' key."
2950 PRINT:PRINT

```

```

2960 PRINT"  Start with a simple pattern, such as"
2970 PRINT"a line, square or diamond."
2980 PRINT:PRINT
2990 PRINT"  When you're happy with your pattern"
3000 PRINT"then press the 'RETURN' key."
3010 PRINT:PRINT:PRINT:PRINT:PRINT
3020 PRINT"Press spacebar to continue"
3030 key=GET
3040 CLS
3050 REM Enable auto key repeat.
3060 *FX12,0
3070 ENDPROC
3080 :
3090 DEF PROC PROMPTS
3100 CLS
3110 PRINT TAB(0,0);"Enter pattern (spacebar & arrow keys).";
3120 PRINT TAB(0,31);"Press H for help, RETURN key to RUN.";
3130 PRINT TAB(20,15);
3140 ENDPROC
3150 :
3160 DEF PROC LOGO
3170 LOCAL x%,y%
3180 FOR i%=1TO8:READ note(i%)
3190 NEXT i%
3200 FOR i%=1TO30:READ x%,y%
3210 PROC BLOCK(x%,y%)
3220 SOUND 1,-15,note(RND(8)),4
3230 PROC WAIT(20)
3240 NEXT i%
3250 PROC WAIT(20)
3260 SOUND 1,-15,101,40
3270 REMSOUND 2,-15,117,40
3280 REMSOUND 3,-15,129,40
3290 PROC WAIT(200)
3300 DATA 5,13,21,25,33,41,49,53
3310 DATA 21,11,19,9,7,9,7,11,15,7
3320 DATA 15,11,29,11,27,15,19,15,15,15
3330 DATA 11,15,19,7,19,13,31,7,29,7
3340 DATA 27,7,27,11,29,15,31,15,15,9

```

THOUGHT GAMES

```
3350 DATA 7,13,9,15,23,7,15,13,27,9
3360 DATA 7,7,21,7,7,15,27,13,19,11
3370 ENDPROC
3380 :
3390 DEF PROCBLOCK(x%,y%)
3400 VDU 31,x%,y%
3410 VDU 224;224;8,8,10,224;224;
3420 ENDPROC
3430 :
3440 DEF PROCWAIT(t%)
3450 TIME=0
3460 REPEAT
3470 UNTIL TIME=t%
3480 ENDPROC
3490 :
3500 ON ERROR OFF:*FX4
3510 *FX12
3520 MODE6:IFERR=17 END
3530 REPORT:PRINT" at line ";ERL
3540 END
```