

Brain Games

Simple Simon

Now you can improve your memory and have fun at the same time with this program which emulates the 'Simon' hand-held toys.

The BBC Micro will generate an ever-growing sequence of tones and colours which you must imitate. You use the cursor keys to repeat the sequence which the computer has just played. It will then add a further tone/colour to the sequence, and you must repeat it from the beginning.

```
10 REM SIMPLE SIMON
20
30 MODE 2
40 ON ERROR GOTO 1680
50 PROC_INITIALISE
60
70 REPEAT
80   PROC_SHOWOFF
90   PROC_NEWGAME
100  REPEAT
110    PROC_GENERATESEQUENCE
120    PROC_FOLLOWSEQUENCE
130    UNTIL WRONG OR CHEAT
140    IF WRONG THEN PROC_SHOWSEQUENCE
150    UNTIL NOT FN_ANOTHER
160
170 MODE 7
180 PROC_RESTORE
190 END
200
210 DEF PROC_SHOWOFF
220 *FX 15
230 pressed=FALSE
```

```

240 REPEAT
250   FOR actual=0 TO 7
260     FOR logical=3 TO 9 STEP 2
270       VDU 19,logical,(actual+(logical/2)) MOD 8,0,0,0
280     NEXT logical
290     PROC_BUZZ(1+(RND(40)MOD4),25)
300     IF NOT INKEY(0) THEN pressed=TRUE
310     IF pressed THEN actual=7
320     PROC_DELAY(25)
330     NEXT actual
340   UNTIL pressed
350 VDU 19,3,1,0,0;19,5,2,0,0;19,7,3,0,0;1,9,4,0,0;
360 ENDPROC
370
380 DEF PROC_DRAW
390 VDU 19,0,7,0,0;19,1,0;0;19,3,0;0;19,5,0;0;19,7,0;0;19,9,0;0;
400 VDU 29,512;512;
410 PX%=0:PY%=0:CX%=50:CY%=50
420 FOR A%=0 TO 90 STEP 9
430   sin=SINRAD(A%):cos=COSRAD(A%)
440   X%=500*sin:Y%=500*cos
450   GCOL 1,1
460   MOVE 0,0:MOVE PX%,PY%:PLOT 85,X%,Y%
470   MOVE 0,0:MOVE -PX%,PY%:PLOT 85,-X%,Y%
480   MOVE 0,0:MOVE PX%,-PY%:PLOT 85,X%,-Y%
490   MOVE 0,0:MOVE -PX%,-PY%:PLOT 85,-X%,-Y%
500   PX%=X%:PY%=Y%:X%=50+400*sin:Y%=50+400*cos
510   GCOL 0,3:MOVE -50,50:MOVE -CX%,CY%:PLOT 85,-X%,Y%
520   GCOL 0,5:MOVE 50,50:MOVE CX%,CY%

```

```

:PLOT 85,X%,Y%
  530   GCOL 0,7:MOVE 50,-50:MOVE CX%,-C
Y%:PLOT 85,X%,-Y%
  540   GCOL 0,9:MOVE -50,-50:MOVE -CX%,
-CY%:PLOT 85,-X%,-Y%
  550   CX%=X%:CY%=Y%
  560   NEXT A%
  570 VDU 29,0;0;5
  580 GCOL 0,6
  590 MOVE 320,544:PRINT"SIMPLE"
  600 MOVE 352,500:PRINT"SIMON"
  610 VDU 4,19,3,1,0,0;19,5,2,0,0;19,7,3
,0,0;19,9,4,0,0;
  620 ENDPROC
  630
  640 DEF PROC_INITIALISE
  650 VDU 23,0,10,32,0;0;0;
  660 vol=-10:REM sound volume
  670 PROC_BUZZ(0,500)
  680 PROC_DRAW
  690 max=19
  700 DIM colour%(max)
  710 *FX 4 1
  720 *FX 11 0
  730 ENDPROC
  740
  750 DEF PROC_DELAY(delay)
  760 LOCAL finishtime
  770 finishtime=TIME+delay
  780 REPEAT UNTIL TIME>=finishtime
  790 ENDPROC
  800
  810 DEF PROC_NEWGAME
  820 WRONG=FALSE:CHEAT=FALSE
  830 end=0
  840 PROC_BUZZ(0,1275)
  850 COLOUR 6:COLOUR 129
  860 PRINT TAB(13,1);"READY!"
  870 REPEAT UNTIL GET$="Y"

```

```

880 PROC_BUZZ(0,0)
890 COLOUR 128
900 PRINT TAB(13,1); "[spc6]"
910 ENDPROC
920
930 DEF PROC_GENERATESEQUENCE
940 current=0
950 REPEAT
960 IF current=end THEN colour%(curr
ent)=1+(RND(40) MOD 4)
970 PROC_FLASH(colour%(current),40)
980 current=current+1
990 UNTIL current>end
1000 end=current
1010 IF current>max THEN CHEAT=TRUE
1020 COLOUR 6:COLOUR 129
1030 PRINT TAB(17,1);end
1040 ENDPROC
1050
1060 DEF PROC_FOLLOWSEQUENCE
1070 *FX 15 1
1080 FOR digit=0 TO end-1
1090 REPEAT
1100 key=INKEY(500)
1110 UNTIL (key>=136 AND key<=139)
OR key=TRUE
1120 col=key-135
1130 IF col<>colour%(digit) THEN WRON
G=TRUE
1140 IF WRONG THEN digit=end-1
1150 IF NOT WRONG THEN PROC_FLASH(col
,30)
1160 NEXT digit
1170 IF WRONG THEN PROC_BUZZ(0,250):PRO
C_DELAY(300)
1180 PROC_DELAY(50)
1190 ENDPROC
1200
1210 DEF PROC_SHOWSEQUENCE

```

```

1220 FOR current=0 TO end-1
1230     PROC_FLASH(colour%(current),30)
1240     NEXT
1250 PROC_DELAY(150)
1260 ENDPROC
1270
1280 DEF PROC_BUZZ(col,dur)
1290 IF col<1 THEN SOUND &110,vol,3,dur
/5:SOUND &111,0,101,dur/5:ENDPROC
1300 SOUND &12,vol,48+(12*col),dur/5
1310 ENDPROC
1320
1330 DEF PROC_FLASH(colour,delay)
1340 VDU 19,1+(2*colour),0;0;
1350 PROC_BUZZ(colour,delay)
1360 PROC_DELAY(delay)
1370 VDU 19,1+(2*colour),colour,0,0;
1380 PROC_DELAY(delay)
1390 ENDPROC
1400
1410 DEF FN_ANOTHER
1420 COLOUR 6
1430 FOR out=1 TO 2
1440     IF CHEAT THEN 1500
1450     PRINT TAB(15,0);"You"
1460     PRINT TAB(12,1);"finished"
1470     PRINT TAB(13,2);end-1;" run";
1480     IF end-1<>1 THEN PRINT"s"
1490     GOTO 1510
1500     PRINT TAB(13,0);"CHEAT!"
1510     PRINT TAB(14,27);"Would"
1520     PRINT TAB(15,28);"you"
1530     PRINT TAB(14,29);"like"
1540     PRINT TAB(12,30);"to play"
1550     PRINT TAB(12,31);"again ?";
1560     *FX 15 1
1570     IF out=1 THEN REPEAT key$=GET$:U
NTIL key$="Y" OR key$="N"
1580     COLOUR 128:COLOUR 0

```

```
1590     NEXT
1600 IF key$="Y" THEN =TRUE ELSE =FALSE
1610
1620 DEF PROC_RESTORE
1630 *FX 4
1640 *FX 12
1650 *FX 15 0
1660 ENDPROC
1670
1680 MODE 7
1690 PROC_RESTORE
1700 REPORT:PRINT" AT LINE ";ERL
```


Mastercode

This clever computer implementation of the ever-popular 'coloured pegs in holes, deduce the code' game is now ready to challenge your wits. Full instructions are within the program.

```
10 REM Mastercode
20
30 MODE 7
40 PROC_init
50 REPEAT
60     PROC_new_game
70     REPEAT
80         PROC_guess
90         PROC_display
100        UNTIL correct OR guess%=10
110    UNTIL NOT FN_another
120 *FX 4
130 END
140
150 DEF PROC_init
160 VDU 23,0,10,32,0;0;0;
170 *FX 4 1
180 DIM play%(9),comp%(9),temp%(9)
190 ENDPROC
200
210 DEF PROC_new_game
220 FOR A%=1 TO 9
230     play%(A%)=-1:comp%(A%)=-1
240     NEXT
250 CLS
260 PRINT TAB(12,0);CHR$(141);" [fs1]Ma
stercode[fs7]"
270 PRINT TAB(12,1);CHR$(141);" [fs1]Ma
```

```

stercode[fs7]"
  280 PRINT'CHR$(141);"[fs3]Instructions
:"
  290 PRINT CHR$(141);"[fs3]Instructions
:"
  300 PRINT"[fs6][spc3][  [fs3]Moves cur
sor to the left"
  310 PRINT"[fs6][spc3]]  [fs3]Moves cur
sor to the right"
  320 PRINT"[fs6] ^[fs3]&[fs6]v[fs3]Chan
ges colour"
  330 PRINT"[fs6]RETURN[fs3]Enters the g
uess"
  340 PRINT'"'[fs2]How many holes (3-9)
? ";
  350 REPEAT H$=GET$
  360   H%=VAL(H$)
  370   UNTIL H%>2 AND H%<10
  380 PRINT H$
  390 FOR A%=1 TO H%
  400   comp%(A%)=RND(8)-1
  410   NEXT
  420 guess%=0
  430 CLS
  440 PRINT TAB(7,0);"[fs3]Mastercode";T
AB(25,1);"[fs1]STATUS"
  450 FOR A%=1 TO H%
  460   PRINT TAB(2+2*A%,2);A%;
  470   NEXT
  480 ENDPROC
  490
  500 DEF PROC_guess
  510 guess%=guess%+1
  520 PRINT TAB(0,23);"[fs1]Move ";guess
%;"[fs3]Your guess ?";
  530 PRINT TAB(0,1+2*guess%);"[fs1]";gu
ess%;"[fs7]";
  540 P%=1
  550 FOR A%=1 TO H%

```

```

560     play%(A%)=0
570     VDU 31,1+2*A%,1+2*guess%,32,32
580     NEXT
590 REPEAT
600     PRINT TAB(2+2*P%,2+2*guess%);" ^"
;
610     K%=GET
620     PRINT TAB(2+2*P%,2+2*guess%);"  "
630     IF K%=136 THEN P%=P%-1
640     IF K%=137 THEN P%=P%+1
650     IF P%=0 THEN P%=H%
660     IF P%=H%+1 THEN P%=1
670     IF K%=138 THEN play%(P%)=play%(P
%)-1
680     IF K%=139 THEN play%(P%)=play%(P
%)+1
690     IF play%(P%)=-1 THEN play%(P%)=7

700     IF play%(P%)=8 THEN play%(P%)=0
710     PRINT TAB(1+2*P%,1+2*guess%);
720     IF play%(P%)=0 THEN VDU 32,32 EL
SE VDU 144+play%(P%),255
730     UNTIL K%=13
740 PRINT TAB(0,2+2*guess%);SPC(39);
750 PRINT TAB(0,23);SPC(39);
760 ENDPROC
770
780 DEF PROC_display
790 W%=0:V%=0
800 FOR A%=1 TO H%
810     temp%(A%)=comp%(A%)
820     NEXT
830 FOR S%=1 TO H%
840     IF play%(S%)=comp%(S%) THEN V%=V
%+1:play%(S%)=100:temp%(S%)=50
850     NEXT
860 FOR S%=1 TO H%
870     FOR T%=1 TO H%
880         IF play%(S%)=temp%(T%) AND S%<

```

```

>T% THEN W%=W%+1:temp%(T%)=89:play%(T%)=
99
    890         NEXT T%
    900     NEXT S%
    910 PRINT TAB(22,1+2*guess%);STRING$(V
%, "[fs2]O");STRING$(W%, "[fs3]O");
    920 IF V%=H% THEN correct=TRUE ELSE co
rrect=FALSE
    930 ENDPROC
    940
    950 DEF FN_another
    960 PRINT TAB(0,22);
    970 IF correct THEN PRINT"[fs1]You gue
ssed it in ";guess%;" go(es)." ELSE PRIN
T"[fs1]Sorry. No more moves allowed."
    980 PRINT'"Another Go ?";
    990 REPEAT K$=GET$
1000     UNTIL K$="Y" OR K$="N"
1010 CLS
1020 =(K$="Y")

```

Barrage

In this fine two-player game you' ll find yourself facing a great screen display of a barrage in the form of a mountain. You' re told the wind speed and direction, and the aim of the game is, of course, to lob shells over the mountain to try and destroy each other' s tanks.

```
10 REM BARRAGE
20 MODE 1
30 PROC_initialise
40 REPEAT
50     PROC_initialiseplay
60     PROC_drawlandscape
70     REPEAT
80         REPEAT
90             PROC_play
100             UNTIL hit
110             PROC_scores
120             UNTIL destroyed
130     UNTIL NOT FN_anothergame
140 END
150
160 REM PROCedures
170
180 DEF PROC_initialise
190 DIM hght%(1),kill%(1),skill%(1),tank%(1),name$(1)
200 VDU 19,0,4,0,0,0
210 VDU 19,2,0,0,0,0
220 VDU 19,3,2,0,0,0
230 VDU 23,128,0,7,15,15,255,255,127,3
1, 23,129,4,200,240,240,254,255,252,248
240 VDU 23,130,32,19,15,15,127,255,63,
```

```

31, 23,131,0,224,240,240,255,255,252,248
250 VDU 23,132,&40,&A0,&40,0,0,0,0,0
260 VDU 23,133,-1,-1,-1,-1,-1,-1,-1,-1
270 VDU 23,0,10,32,0;0;0;
280 *FX 9 2
290 *FX 10 2
300 rnd=RND(-TIME)
310 g=9.81
320 COLOUR 3
330 PRINT'"Skill level's"
340 RESTORE 1980
350 FOR level=1 TO 5
360     READ SKILL$
370     PRINT level;") ";SKILL$
380     NEXT
390 FOR player%=0 TO 1
400     COLOUR player%+1
410     PRINT'"Player ";player%+1;"'s na
me :";
420     INPUT"name$(player%):name$(play
er%)=LEFT$(name$(player%),10)
430     IF name$(player%)="" THEN name$(
player%)="PLAYER "+STR$(player%+1)
440     PRINT'name$(player%);"'s skill l
evel 1-5"'(novice to expert) :";
450     REPEAT KEY$=GET$
460         UNTIL KEY$>"0" AND KEY$<"6"
470     RESTORE 1980
480     FOR level=1 TO VAL(KEY$)
490         READ SKILL$
500         NEXT
510     PRINT SKILL$
520     skill%(player%)=VAL(KEY$)
530     NEXT
540 PRINT'"Press SPACE BAR to play";
550 REPEAT UNTIL GET=32
560 COLOUR 128:COLOUR 1
570 ENDPROC
580 DEF PROC_drawlandscape

```

```

590 VDU 26,16
600 VDU 24,0;0;1279;1023;29,0;128;
610 GCOL 0,3
620 hght%(0)=RND(100)
630 height=hght%(0)
640 maxh=hght%(0)
650 step=30
660 FOR xcord=0 TO 1279 STEP step
670     MOVE xcord,-128
680     MOVE xcord+step,-128
690     PLOT 85,xcord,height
700     IF xcord>minx AND xcord<midx THE
N height=height+RND(40)
710     IF xcord>midx AND xcord<maxx THE
N height=height-RND(40)
720     IF height<0 THEN height=0
730     PLOT 85,xcord+step,height
740     IF height>maxh THEN maxh=height
750     NEXT xcord
760 IF height>0 THEN hght%(1)=height E
LSE hght%(1)=0
770 VDU 5,18,0,1,25,4,tank%(0);hght%(0
)+32;128,129
780 VDU 18,0,2,25,4,tank%(1);hght%(1)+
32;130,131,4
790 GCOL 0,2
800 ENDPROC
810 DEF PROC_initialiseplay
820 destroyed=FALSE
830 minx=100+RND(200)
840 midx=540+RND(200)
850 maxx=1179-RND(200)
860 tank%(0)=RND(minx-64)
870 tank%(1)=maxx+RND(1215-maxx)
880 kill%(0)=0
890 kill%(1)=0
900 first=RND(2)-1
910 ENDPROC
920 DEF PROC_play

```

```

930 COLOUR 131
940 hit=FALSE
950 wind=RND(30)-15
960 COLOUR 0:VDU 28,0,31,39,28
970 PRINT TAB(9,0);"WIND SPEED ";ABS(w
ind);" Kmh ";
980 IF wind<0 THEN PRINT"WEST  "
990 IF wind>0 THEN PRINT"EAST  "
1000 IF wind=0 THEN PRINT"[spc6]"
1010 VDU 28,0,31,39,29
1020 FOR player%=first TO 1
1030     first=0
1040     REPEAT
1050         CLS:COLOUR player%+1
1060         VDU 5,18,0,1,25,4,tank%(0);hgh
t%(0)+32;128,129,18,0,2,25,4,tank%(1);hg
ht%(1)+32;130,131,4
1070         PRINT name$(player%);" 's go."
1080         INPUT"Angle of elevation 10[fs
4]-69[fs4]  ":"ANG
1090         UNTIL ANG>9 AND ANG<70
1100         ANG=RAD(ANG)
1110         VDU 28,0,31,39,31
1120         REPEAT
1130             INPUT"Velocity 1-19 kmh  ":"VEL
1140             UNTIL VEL>0 AND VEL<20
1150             PRINT"Velocity 1-19 kmh  ":"VEL;
1160             VDU 28,0,31,39,29
1170             VEL=VEL*10
1180             X1%=FNX(ANG,VEL,.1,player%)
1190             Y1%=FNY(ANG,VEL,.1,player%)
1200             offset=0
1210             GCOL 0,1
1220             TIME=10
1230             REPEAT
1240                 X%=FNX(ANG,VEL,TIME/50,player%
)
1250                 Y%=FNY(ANG,VEL,TIME/50,player%
)

```



```

1260      IF SGN(midx-X%)=SGN(wind) OR Y
%>=maxh THEN offset=offset+(wind/10)
1270      X%=X%+offset
1280      SOUND &11,-10,Y%DIV4,5
1290      PLOT 71,X1%,Y1%
1300      X1%=X%:Y1%=Y%
1310      C%=POINT(X%,Y%)
1320      PLOT 69,X%,Y%
1330      UNTIL X%<0 OR X%>1280 OR (C%<>
0 AND Y%<1200)
1340      GCOL 0,3:PLOT 69,X%,Y%
1350      SOUND &10,-10,6,10
1360      FOR Y1%=Y% TO maxh STEP 4
1370          C%=POINT(X%,Y1%)
1380          IF C%=1 OR C%=2 THEN hit=TRUE:
Y1%=maxh
1390          IF C%=0 THEN Y1%=maxh
1400          NEXT
1410      IF (C%=1 AND player%=1) OR (C%=2
AND player%=0) THEN hit=TRUE
1420      IF (X%<midx AND Y%<=hght%(0)) OR
(X%>midx AND Y%<=hght%(1)) OR hit THEN
1540
1430      VDU 18,0,0,5
1440      REPEAT CY%=POINT(X%,Y%)
1450          REPEAT CX%=POINT(X%,Y%)
1460              MOVE X%-16,Y%+28:VDU 133
1470              X%=X%+(26*SGN(X%-midx))
1480              UNTIL CX%<>3
1490              X%=X1%
1500              Y%=Y%+32
1510              UNTIL CY%<>3
1520      VDU4
1530      temp=player%
1540      IF hit THEN player%=1
1550      NEXT
1560      player%=temp
1570      ENDPROC
1580      DEF PROC_scores

```

```

1590 VDU 19,1+((player%+1) MOD 2),12,0,
0,0
1600 S%=1:V%=0
1610 REPEAT V%=V%+S%
1620     SOUND 0,-V%,6,2
1630     IF V%=15 THEN S%=-1
1640     UNTIL V%=0
1650 kill%(player%)=kill%(player%)+1
1660 IF kill%(player%)=skill%(player%)
THEN destroyed=TRUE
1670 IF player%=0 THEN VDU 19,2,0,0,0,0
ELSE VDU 19,1,1,0,0,0
1680 CLS:COLOUR player%+1
1690 PRINT name$(player%); " has scored
a hit.";
1700 FOR person%=0 TO 1
1710     COLOUR person%+1
1720     PRINT 'name$(person); ": "; TAB(11
);kill%(person%); "/" ;skill%(person%);
1730     NEXT
1740 *FX 15 1
1750 delay=GET
1760 ENDPROC
1770 DEF FN_anothergame
1780 IF skill%(player%)<5 THEN skill%(p
layer%)=skill%(player%)~1
1790 CLS
1800 PRINT "Do you wish start another
battle : ";
1810 *FX 15 1
1820 REPEAT
1830     KEY$=GET$
1840     UNTIL KEY$="Y" OR KEY$="N"
1850 PRINT KEY$
1860 IF KEY$="N" THEN =FALSE
1870 PRINTname$(player%); "'s skill leve
l is now ";skill%(player%)'ie ";
1880 RESTORE 1980
1890 FOR level=1 TO skill%(player%)

```

```

1900    READ SKILL$
1910    NEXT
1920 PRINT SKILL$
1930 A=GET
1940 =TRUE
1950 DEF FNX(ang,vel,time,player)
1960 IF player=0 THEN =tank%(player%)+5
6+vel*time*COS(ang) ELSE =tank%(player%)
-(vel*time*COS(ang))+8
1970 DEF FNY(ang,vel,time,player)=hght%
(player)+28+(vel*time*SIN(ang)-(0.5*g*time^2))
1980 DATA Rubbish,Passable,Average,Good
,Brilliant

```

Hangman

Now you can play the ' guess the word I' m thinking of, before you get hanged' game against the BBC Micro. The computer chooses a word from its DATA store (from line 890) then challenges you to guess it before a complete picture of a hanged man is drawn using PROC_HANG (from line 660). Once you' v played a few hundred games with this program, and you know the words, change them to a vocabulary of your own choice. Make sure you list of words ends with ' _999' as you can see in line 980.

```
10 REM HANG MAN
20
30 MODE 7
40 PROC_INIT
50 REPEAT
60   WORD$=FN_WORD(RND(W%))
70   PROC_DISP(WORD$)
80   REPEAT
90     PROC_GUESS
100    UNTIL GOT OR HUNG
110    PROC_RESULTS
120    UNTIL NOT FN_ANOTHER
130 END
140 DEF PROC_INIT
150 PRINT TAB(15,2); "[fs1]HANGMAN[fs7]
" ' '
160 PRINT ' ' "[fs8][fs2]COUNTING WORDS[f
s9] ";
170 W%=0
180 REPEAT READ WORD$
190   IF WORD$<>"-999" THEN W%=W%+1
200   Z=POS:PRINT ;W%;STRING$(POS-Z,CH
```

```

R$(8));
210 UNTIL WORD$="-999"
220 ENDPROC
230 DEF FN_WORD(P%)
240 LOCAL C%,CHOSEN$
250 RESTORE 890
260 FOR C%=1 TO P%
270 READ CHOSEN$
280 NEXT
290 ALP$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
300 =CHOSEN$
310 DEF PROC_DISP(OUT$)
320 LOCAL P%,C$
330 CLS
340 PRINT TAB(0,0);"[fs3]Not used:[fs2
]";ALP$
350 PRINT TAB(0,2);"[fs3]Guesses :[fs2
]0"
360 PRINT TAB(0,4);"[fs3]Wrong[spc3]:[
fs2]0"
370 FOR P%=1 TO LEN(OUT$)
380 PRINT TAB(14-(LEN(OUT$) DIV 2)+P
%,20);
390 IF MID$(OUT$,P%,1)="" THEN PRINT
" " ELSE PRINT"- "
400 NEXT
410 PRINT TAB(9,23);"[fs6]Your guess:[
fs3]";
420 GUESS$=STRING$(LEN(OUT$)," ")
430 wrong=0:GUESSES=0
440 ENDPROC
450 DEF PROC_GUESS
460 LOCAL KEY$,P%,WRONG
470 WRONG=TRUE
480 PRINT TAB(22,23);
490 REPEAT KEY$=GET$
500 IF KEY$>="a" AND KEY$<="z" THEN
KEY$=CHR$(ASC(KEY$)-32)
510 UNTIL INSTR(ALP$,KEY$)<>0 AND KE

```

```

Y$<>" "
  520 PRINT KEY$;
  530 FOR P%=1 TO LEN(WORD$)
  540   IF KEY$=MID$(WORD$,P%,1) THEN PR
INT TAB(14-(LEN(WORD$) DIV 2)+P%,20);KEY
$:GUES$=LEFT$(GUESS$,P%-1)+KEY$+MID$(GUE
SS$,P%+1):WRONG=FALSE
  550   NEXT
  560 ALP$=LEFT$(ALP$,INSTR(ALP$,KEY$)-1
)+ " "+MID$(ALP$,INSTR(ALP$,KEY$)+1)
  570 PRINT TAB(11,0);ALP$
  580 GOT=(WORD$=GUESS$)
  590 wrong=wrong-WRONG
  600 GUESSES=GUESSES+1
  610 HUNG=(wrong=10)
  620 PRINT TAB(11,2);GUESSES
  630 PRINT TAB(11,4);wrong
  640 IF WRONG THEN PROC_HANG
  650 ENDPROC
  660 DEF PROC_HANG
  670 ON wrong GOTO 680,690,700,710,720,
730,740,750,760,770
  680 PRINT TAB(10,18);"=====":ENDPROC
  690 PRINT TAB(10,17);STRING$(5,"|"+CHR
$(8)+CHR$(11));:ENDPROC
  700 PRINT TAB(10,12);"=====":ENDPROC
  710 PRINT TAB(14,13);"|":ENDPROC
  720 PRINT TAB(14,14);"O":ENDPROC
  730 PRINT TAB(14,15);"!":ENDPROC
  740 PRINT TAB(13,15);"/":ENDPROC
  750 PRINT TAB(15,15);"_":ENDPROC
  760 PRINT TAB(13,16);"/":ENDPROC
  770 PRINT TAB(15,16);"_":ENDPROC
  780 DEF PROC_RESULTS
  790 VDU 31,0,23,10,10,10,10,10,10,30,1
1,11,11,11,11
  800 IF GOT THEN PRINT"[fs1]You got it
in ";GUESSES;" tries.":ENDPROC
  810 PRINT"[fs1]You did not get the wor

```

```

d";:IF INSTR(WORD$," ")<>0 THEN PRINT"s.
They were" ELSE PRINT".It was"
    820 PRINT"[fs3]";WORD$;"[fs2]You made
";GUESSES;" guesses."
    830 ENDPROC
    840 DEF FN_ANOTHER
    850 PRINT'"[fs6]Would you like another
try:";
    860 REPEAT KEY$=GET$
    870     UNTIL KEY$="Y" OR KEY$="y" OR KE
Y$="N" OR KEY$="n"
    880 IF KEY$="Y" OR KEY$="y" THEN =TRUE
ELSE =FALSE
    890 DATA ACCUMULATOR,ADDRESS,ALGORITHM
,ALPHANUMERIC,APL,APPLICATION,SOFTWARE,A
SCII,ASSEMBLER
    900 DATA BASIC,BATCH,BAUD,BENCHMARK,BI
NARY,BOOLEAN,ALGEBRA,BOOTSTRAP,BUFFER,BU
S,BYTE
    910 DATA CPU,CHARACTER,COBOL,COMPILER,
CONCATENATE,CPM
    920 DATA DATA,DATABASE,DEBUG,DISC,DOCU
MENTATION,DOS,DOT MATRIX,DOUBLE DENSITY,
DYNAMIC
    930 DATA EPROM,ERROR,FIELD,FILE,FIRMA
RE,FLAG,FLOPPY DISC,FLOWCHART,FORTTRAN
    940 DATA GATE,GIGO,GLOBAL,GRAPHICS,HAR
D COPY,HARDWARE,HEXADECIMAL,HEX PAD,HIGH
LEVEL
    950 DATA INPUT,INPUT,INSTRUCTION,INTER
RUPT,INTERPRETER,JOYSTICK,KILOBYTE,LINE
PRINTER,LOW LEVEL
    960 DATA MACHINE LANGUAGE,MACHINE CODE
,MEMORY,MICROPROCESSOR,MODEM,MONITOR,MOT
HERBORAD,MOUSE
    970 DATA NETWORK,NUMERIC PAD,OCTAL,ON
LINE,OPERATING SYSTEM,OUTPUT,PASCAL,PORT
,PROCEDURE,PROGRAM,PILOT,RAM,REAL TIME,R
ANDOM,REFRESH,REGISTER,RESET,ROM

```

980 DATA SNOBOL, SOFTWARE, STACK, SUBROUT
INE, SYNTAX, SYSTEMS, SOFTWARE, THERMAL PRIN
TER, TIME SHARING, TURNKEY SYSTEM, WINCHEST
ER, UNIX, VMS, VIRTUAL, VOLATILE, "-999"

Solitaire Maze

Here's a game for the very patient and skilled. Written initially by Mike O' Neill and modified by Ian Hutt, Solitaire Maze challenges you to a game of strategy.

You control between one and five circular pieces. There is an equal number of diamond-shaped pieces which do not move during the game. The object of the exercise is to move your pieces around the maze so that ultimately each one covers a different diamond.

A timer runs while the program is underway. This tells you how long it took to complete the latest game, and the fastest time so far.

```
10 REM Solitaire Maze
20
30 MODE 1
40 PROC_init
50 REPEAT
60     PROC_new_game
70     PROC_maze
80     PROC_setup
90     REPEAT
100         PROC_play
110         UNTIL got=circ
120     UNTIL NOT FN_another
130 PROC_normal
140 END
150
160 DEF PROC_init
170 VDU 23,0,10,32,0;0;0;
180 VDU 23,128,&00,&3E,&7F,&7F,&7F,&7F
,&7F,&3E
190 VDU 23,129,&00,&08,&1C,&3E,&7F,&3E
```

```

, &1C, &08
  200 VDU 23,130,-1,129,129,129,129,129,
129,-1
  210 DIM X%(4),Y%(4)
  220 *FX 4 1
  230 high=-1
  240 COLOUR 1:COLOUR 130
  250 PRINT TAB(13,1);"Solitaire Maze"
  260 ENDPROC
  270
  280 DEF PROC_new_game
  290 COLOUR 1:COLOUR 128
  300 PRINT TAB(0,27);"How many pieces d
o you want (1-5) ?";
  310 REPEAT K$=GET$
  320   UNTIL K$>"0" AND K$<"6"
  330   circ=VAL(K$)
  340 ENDPROC
  350
  360 DEF PROC_maze
  370 CLS
  380 RESTORE
  390 COLOUR 2
  400 PRINT TAB(9,1);"Solitaire Maze"''
  410 FOR Y%=0 TO 21
  420   READ S$
  430   FOR X%=1 TO 32
  440     IF MID$(S$,X%,1)=" " THEN VDU
32 ELSE VDU 130
  450     NEXT X%
  460   PRINT
  470   NEXT Y%
  480 COLOUR 1
  490 FOR C%=1 TO circ
  500   REPEAT
  510     X%=RND(30):Y%=4+RND(20)
  520     UNTIL FN_ch(X%,Y%)=" "
  530     PRINT TAB(X%,Y%);"[fs1]";
  540     SOUND 1,-3*C%,53+4*C%,1

```

```

550     PROC_delay(20)
560     NEXT
570 COLOUR 3
580 FOR C%=0 TO circ-1
590     REPEAT
600         X%(C%)=RND(30):Y%(C%)=4+RND(20
)
610         UNTIL FN_ch(X%(C%),Y%(C%))=" "
620     PRINT TAB(X%(C%),Y%(C%));"[fs0]"
;
630     SOUND 1,-3*C%,101+4*C%,1
640     PROC_delay(20)
650     NEXT
660 PRINT TAB(33,6);"TIME"
670 PRINT TAB(33,8);"0"
680 IF high=-1 THEN 710
690 PRINT TAB(32,11);"Fastest"
700 PRINT TAB(33,13);high DIV 10 /10
710 ENDPROC
720
730 DEF PROC_setup
740 COLOUR 1
750 PRINT TAB(0,29);"Press SPACE BAR t
o start"
760 FOR C%=10 TO 0 STEP -1
770     PRINT TAB(30,29);C%;" "
780     SOUND 1,C%-4,101,5
790     TIME=0
800     *FX 15 1
810     K$=INKEY$(100)
820     IF K$=" " THEN C%=0 ELSE IF K$<>
" " THEN K$=INKEY$(100-TIME):GOTO 820
830     NEXT
840 PRINT TAB(0,29);SPC(39);
850 got=0
860 COLOUR 3
870 TIME=0
880 ENDPROC
890

```

```

900 DEF PROC_play
910 PRINT TAB(33,8);TIME DIV10 /10
920 K%=INKEY(0)
930 *FX 15 1
940 IF K%<136 THEN ENDPROC
950 DX%=0:DY%=0
960 IF K%=139 THEN DY%=DY%-1
970 IF K%=138 THEN DY%=DY%+1
980 IF K%=136 THEN DX%=DX%-1
990 IF K%=137 THEN DX%=DX%+1
1000 IF DX%=0 AND DY%=0 THEN ENDPROC
1010 FOR C%=0 TO circ-1
1020     C$=FN_ch(X%(C%)+DX%,Y%(C%)+DY%)
1030     IF C$="[fs2]" OR C$="[fs0]" THEN
1080
1040     PRINT TAB(X%(C%),Y%(C%));" ";
1050     X%(C%)=X%(C%)+DX%;Y%(C%)=Y%(C%)+
DY%
1060     PRINT TAB(X%(C%),Y%(C%));"[fs0]"
;
1070     IF C$="[fs1]" THEN COLOUR 2:PRIN
T TAB(X%(C%),Y%(C%));"[fs0]":COLOUR 3:X%
(C%)=-1:Y%(C%)=-1:got=got+1
1080     NEXT
1090 ENDPROC
1100
1110 DEF FN_another
1120 T%=TIME
1130 PRINT TAB(0,27);"Well done.You com
pleted the course in"
1140 PRINT;T% DIV10 /10;" seconds.";
1150 IF T%<high OR high<0 THEN high=T%:
PRINT"The fastest time!"
1160 *FX 15 1
1170 PRINT TAB(0,30);"Would you like an
other go ?";
1180 REPEAT K$=GET$
1190     UNTIL K$="Y" OR K$="N"
1200 VDU 28,0,31,39,26,12,26

```

```

1210 =(K$="Y")
1220
1230 DEF PROC_normal
1240 *FX 4
1250 *FX 15 1
1260 CLS
1270 ENDPROC
1280
1290 DEF FN_ch(X%,Y%)
1300 IF X%<1 OR Y%<5 THEN ="[fs2]"
1310 VDU 31,X%,Y%
1320 A%=135
1330 =CHR$((USR(&FFF4)AND&FFFF)DIV&100)

1340
1350 DEF PROC_delay(T%)
1360 t%=TIME+T%
1370 REPEAT UNTIL TIME>=t%
1380 ENDPROC
1390 DATA "#####
####"
1400 DATA "#[spc17]#[spc3]#[spc8]#"
1410 DATA "# ##### [spc3]####
### #"
1420 DATA "# [spc9]#[spc3]# [spc6]#[s
pc3]# #"
1430 DATA "#[spc3]##### # #####
# # #"
1440 DATA "# # [spc8]# [spc6]# # [s
pc3]#"
1450 DATA "# # # ## # ##### [spc3]
#[spc3]# #"
1460 DATA "# # # [spc4]# [spc7]# [sp
c3]# # #"
1470 DATA "# [spc3]# #### # # [spc3
]# # # #"
1480 DATA "# # # # [spc7]# # # # #
###"
1490 DATA "# # [spc3]##### # [spc3]#"

```

```

# #[spc3]# #"
1500 DATA "# ### #[spc7]#[spc4]# # # ##
#[spc3]#"
1510 DATA "# # # # # ##### # ## #[spc5]
# # #"
1520 DATA "# # # #[spc9]# # # # # ##
#"
1530 DATA "##[spc7]##### ## ## ### # #
#"
1540 DATA "# ######[spc6]#[spc4]#[spc3]#
###[spc3]##"
1550 DATA "#####[spc3]# ### ## # # #[spc
4]# # #"
1560 DATA "# # #[spc3]#[spc3]# # #[spc3
]## ##### # ##"
1570 DATA "# #[spc3]#[spc3]#[spc5]# #[s
pc9]# #"
1580 DATA "# ### # ### ###[spc3]#####
#### #"
1590 DATA "##[spc5]# #[spc7]#[spc5]#[spc
5]# #"
1600 DATA "#####
#####"

```