

70-220 displaying the maze and tilt indicators  
 230 starting the clock and initialising variables  
 240-250 re-entry point for each calculation: clearing tilt pointers from screen  
 260-350 reading keys, calculating and displaying tilt  
 360-370 calculating the next position for the ball  
 380-460 finding out if the ball is free to move to this position, and what would happen if it went there  
 470 moving the ball (unless already down a hole)  
 480-490 preparing for next move  
 500-600 ball falls down a hole (sound effect and return ball to start)  
 610-650 ball falls down final hole; display time  
 1000-1230 DATA for walls of maze  
 2000-2010 DATA for positions of holes (except the green one)  
 3000-3010 subroutine to detect when a sound is finished

### The program

```

10 REM ** BALL MAZE **
20 DIM H(20,2)
30 CTLSPR 1,8: CTLSPR 2,2: CTLSPR 6,0
40 GENPAT 1,129,0,60,126,126,126,126,60,
0: GENPAT 1,151,255,255,255,255,255,255,
255,255
50 GENPAT 1,131,0,60,126,102,102,126,60,
0: GENPAT 1,132,0,126,126,126,126,126,12
6,0
60 GENPAT 3,1,24,60,126,219,153,24,24,24
: GENPAT 3,2,24,48,96,255,255,96,48,24
70 VS 4: CLS : COLOUR 0,15: COLOUR 1,4:
COLOUR 2,4: COLOUR 4,4
80 CSR 0,0: FOR J=0 TO 23
90 READ A$
100 FOR K=0 TO 31: IF K=31 AND J=23 THEN
GOTO 120
110 IF A$(K+1)="1" THEN PRINT " "; ELSE
PRINT CHR$(151);
120 NEXT : NEXT
130 COLOUR 1,1: FOR J=1 TO 20
140 READ H(J,1): READ H(J,2): CSR H(J,1)
,H(J,2): PRINT CHR$(129)

```

```

150 NEXT
160 COLOUR 1,8: CSR 1,1: PRINT CHR$(131)
170 LET X=1: LET Y=1: LET NX=1: LET NY=1
180 COLOUR 1,2: CSR 29,21: PRINT CHR$(13
2)
190 COLOUR 1,13: CSR 31,12: PRINT "*"
200 CSR 15,23: PRINT "*";
210 SPRITE 1,1,124,4,0,0,0: LET SX=124
220 SPRITE 2,2,252,92,0,0,0: LET SY=92
230 CLOCK "000000": LET AX=0: LET AY=0:
LET VX=0: LET VY=0
240 IF AX<>0 THEN ADJSR 1,1,0
250 IF AY<>0 THEN ADJSR 1,2,0
260 LET A=ASC(INKEY$)
270 IF A=25 AND AX<0 OR A=8 AND AX>0 THE
N LET AX=0
280 IF A=25 AND AX<10 THEN LET AX=AX+1
290 IF A=8 AND AX>-10 THEN LET AX=AX-1
300 IF A=10 AND AY<0 OR A=11 AND AY>0 TH
EN LET AY=0
310 IF A=10 AND AY<10 THEN LET AY=AY+1
320 IF A=11 AND AY>-10 THEN LET AY=AY-1
330 IF AX<>0 THEN ADJSR 2,1,SX+8*AX: A
DJSR 1,1,13
340 IF AY<>0 THEN ADJSR 3,2,SY-8*AY: A
DJSR 1,2,13
350 LET VX=VX+AX*2: LET VY=VY+AY*2
360 IF ABS(VX)>10 THEN LET NX=X+1*SGN(V
X): LET VX=0
370 IF ABS(VY)>10 THEN LET NY=Y+1*SGN(V
Y): LET VY=0
380 CSR NX,Y
390 LET CX=ASC(SPK$)
400 CSR X,NY: LET CY=ASC(SPK$)
410 CSR NX,NY: LET CZ=ASC(SPK$)
420 IF CX=129 OR CY=129 OR CZ=129 THEN
GOTO 500
430 IF CX=151 THEN LET VX=0: LET NX=X
440 IF CY=151 THEN LET VY=0: LET NY=Y
450 IF CZ=151 THEN LET VX=0: LET VY=0:
LET NX=X: LET NY=Y
460 IF CX=132 OR CY=132 THEN GOTO 610
470 CSR X,Y: PRINT " ": CSR NX,NY: PRINT
CHR$(131)
480 LET X=NX: LET Y=NY

```



```

490 GOTO 240
500 CSR X,Y: PRINT " "
510 IF AX<>0 THEN ADJSR 1,1,0
520 IF AY<>0 THEN ADJSR 1,2,0
530 SOUND 0,400,160,50,0,100,1
540 GOSUB 3000
550 SOUND 0,1000,240,0,-15,70,1
560 GOSUB 3000
570 SOUND 0,0,0
580 LET AX=0: LET AY=0: LET VX=0: LET VY
=0
590 LET X=1: LET Y=1: LET NX=1: LET NY=1
600 GOTO 210
610 CLS : VS 4: CLS
620 LET T=VAL(MID$(TIME$,3,2))
630 CSR 3,3: ADJSR 1,1,0: ADJSR 1,2,0
640 PRINT "You won! - in ";T;" minutes"
650 GOTO 650
1000 DATA 00000000000000000000000000000000
00
1010 DATA 01111111111111111110111010101
00
1020 DATA 00000001000000000000000101010101
01
1030 DATA 0111110111110111111111101011101
01
1040 DATA 0100010100010100000000001010101
01
1050 DATA 01110111110111110111011101010101
01
1060 DATA 00010001010100000101010101010101
01
1070 DATA 010111010101011101010101010111
01
1080 DATA 010001010101010101010100010100
01
1090 DATA 010111010100010111010101110101
01
1100 DATA 011100010101010000010101000001
01
1110 DATA 010111010101111111011111011111
01
1120 DATA 010001000100010101000100010000
01

```

```

1130 DATA 010111111101110101110101010111
01
1140 DATA 010100000001000101000001010101
01
1150 DATA 011111111111011101011101111101
01
1160 DATA 010100000000010001010101000001
01
1170 DATA 010101111111011111010101110111
01
1180 DATA 011101000001010000010101010101
01
1190 DATA 010101011111110111010101010101
01
1200 DATA 010101000000000101000100010100
01
1210 DATA 010101111111111101111111110111
01
1220 DATA 000000000000000000000000000000
01
1230 DATA 000001111111111111111111110000
00
2000 DATA 19,1,25,1,29,1,15,5,1,7,23,7,1
1,8,27,9,29,9,7,11
2010 DATA 11,11,21,13,23,13,15,15,3,17,2
9,18,7,19,19,23,19,1,21
3000 IF PEEK(64082)<>PEEK(64086) THEN G
OTO 3000
3010 RETURN

```

Variations

A new maze can be designed on a sheet of squared paper, 32 squares across and 24 squares deep. This includes the wall around the maze and the tilt indicator area. Shade in squares to define where the dividing walls are to be. The DATA consists of 24 groups of 32 digits, one group for each row of the design. Where there is a shaded square (wall), the digit is a '1'; where there is a blank square (path) there is a '0'. Replace the listed lines 1000 to 1230 with lines containing details of your own maze. Alter lines 2000 and 2010 to give the positions of the holes in your maze. These are given as pairs