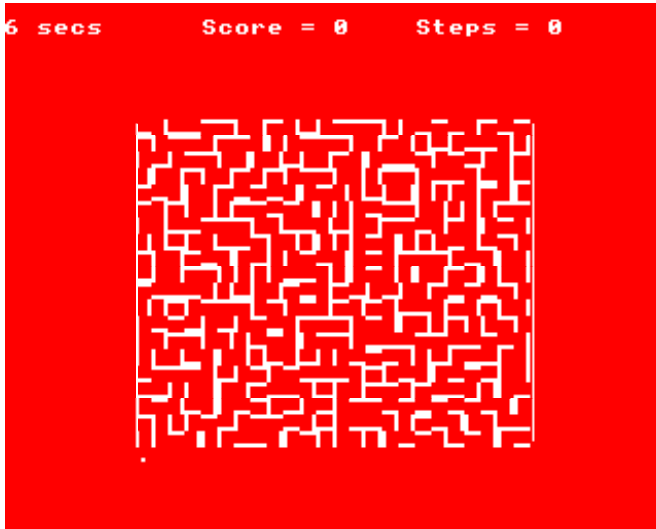


A-MAZE-ING



YOU are in the middle of a complicated maze, and your objective is to reach the outside in the quickest possible time, but also with the fewest number of moves possible.

Don't rush headlong into this one as a little forward planning can save you time and points.

Every step counts as a point and every time you try to cross a barrier counts as a point.

You must aim for the top right hand corner or the exit nearest to that one.

How to play

You are represented by a dot in the lower left hand side of the maze, and you move by using the ARROW cursor keys.

Your score will be displayed at the top of the screen. The number of steps taken has a greater effect on your final score than the time factor.

You may, of course, retrace your steps and begin again from any point you wish to. When you have reached the outside, or you wish to give up, press the ESCAPE key.

We hope you make it, as there are plenty more 'Brainteasers' waiting for you on the outside.

Programming hints

The maze is created of cells, each of which have one side blocked off. The cell shapes are drawn using VDU 23 to create user-defined graphics.

You could increase the size of the maze by changing 20 and 24 in lines 230 and 250. The maze shown is 24 columns wide by 20 rows. The maze shown is 24 columns wide by 20 rows. The maze array M% must be DIMensioned 1 column and 2 rows larger than the actual array to allow for checking the (X+1)th column and (Y+1)th row. If you want the array in the centre of the screen you should change the PRINT statements in lines 220 and 240.

The lines drawn down the side of the maze in lines 300 and 310 would have to be changed, so would the start

position of the dot in lines 320 and 330. In a 20 row maze, the 21st row is the row the dot starts on, so special conditions apply to this row in lines 370 to 430. If a different number of rows is chosen this 21 must be changed.

```

10 REM A-MAZE-ING
20 REM COPYRIGHT (C) G.LUDINSKI 1983
22 ON ERROR RUN
30 MODE4:VDU23;8202;0;0;0;
40 DIM M%(25,22)
50 CLS:COLOUR1:COLOUR128
52 VDU 19,0,1,0,0,0
54 VDU 23;8202;0;0;0;
60 LA$=CHR$(136):RA$=CHR$(137):DA$=CH
R$(138):UA$=CHR$(139)
70 GOTO 110
80 DEF FNB(N$)
90 TF=0
100 FOR L=0 TO 7
110 TF=TF+(2^L)*VAL(MID$(N$,8-L,1))
120 NEXT
130 =TF
140 VDU23,224,3,3,3,3,3,3,3,3
150 VDU23,225,0,0,0,0,0,0,255,255
160 VDU23,226,192,192,192,192,192,192,192,
192,192
170 VDU23,227,255,255,0,0,0,0,0,0
180 VDU23,228,3,3,3,FNB("00011011"),FN
B("00011011"),3,3,3
190 VDU23,229,0,0,0,FNB("00011000"),FN
B("00011000"),0,255,255
200 VDU23,230,192,192,192,FNB("1101100
0"),FNB("11011000"),192,192,192
210 VDU23,231,255,255,0,FNB("00011000"
),FNB("00011000"),0,0,0
220 PRINT:PRINT:PRINT:PRINT:PRINT:PRIN
T
230 FOR I=1 TO 20
240 PRINT:PRINT " ";
250 FOR J=1 TO 24
260 M%(J,I)=INT(RND(1)*4)
270 PRINT CHR$(224+M%(J,I));
280 NEXT:NEXT
290 MOVE256,174:DRAW256,790
310 MOVE1028,174:DRAW1028,790
320 ST=0:X=1:Y=21:X1=0:Y1=0:TIME=0
330 PRINT TAB(X+7,Y+6)" ";CHR$(8);
340 *FX4,1
350 SC=ST+INT(TIME/500)
360 I$=INKEY$(0):IF I$="" OR I$ < CHR$
(136) OR I$ > CHR$(139) THEN PRINT TAB(0
,1);INT(TIME/100);" secs"TAB(12,1)"Score
=";SC;TAB(25,1)"Steps=";ST:GOTO 340
370 IF (X=1 AND I$=LA$) OR (X=24 AND I
$=RA$) OR (Y=1 AND I$=UA$) OR (Y=21 AND
I$=DA$) THEN GOTO 330
380 IF I$ = LA$ AND ((M%(X-1,Y) <> 0 A
ND M%(X,Y) <> 2) OR Y=21) THEN X=X-1
390 IF I$ = RA$ AND ((M%(X+1,Y) <> 2 A
ND M%(X,Y) <> 0) OR Y=21) THEN X=X+1
400 IF I$ = DA$ AND ((M%(X,Y+1) <> 3 A
ND M%(X,Y) <> 1) OR Y=21) THEN Y=Y+1
410 IF I$ = UA$ AND ((M%(X,Y-1) <> 1 A
ND M%(X,Y) <> 3) OR Y=21) THEN Y=Y-1

```

```

420 ST=ST+1
430 IF Y=21 THEN PRINT TAB(X+7,27) ". ";
:GOTO 430
440 PRINT TAB(X+7,Y+6);CHR$(M%(X,Y)+22
8);CHR$(8);
450 IF X=X1 AND Y=Y1 THEN SOUND 1,-15,
53,10
460 X1=X:Y1=Y
470 GOTO 330

```