

31. Topsy

General Description

This program has been around for several years. The object is to steer a drunken person home using compass directions, N, S, E, W as well as the intermediate points of the compass.

You will see the compass in the top left-hand corner of the screen without any intermediate points, two sets of red brick walls with a possible route through them to home on the right of the screen which is in green. The drunk is also coloured as a red blob, though you may wish to change this. The detailed information will tell you how. Limit checks are made as to what is a sensible move and even for adults it can be surprisingly difficult. The program uses sound well, but has been written in Mode 7 so that it will run on a Model 'A'. If you rewrite it with full graphics for a Model 'B', I shall be pleased to hear from you.

Detailed Description

Lines 60-170 Clears variables. Game operates in Mode 7. Deletes and disables cursor. Saves array spaces for four walls.

180-270 Holds the main structure of the program.

280-680 PROCdisplay: Displays compass points, four walls, plots homebase and men.

690-1030 User requested to input direction and number of goes.

1040-1120 This procedure checks to see if user's position will fit on screen. If not message displayed.

1140-1150 Prints number of attempts and the character position is updated. Will character hit wall? Lines 1350-1380 check to see if character is to the right of the wall reprinted number of moves checked.

1500-1580 User' sposition takes character out of bounds.
Display message and activate sound envelope.

1740-1820 User didn' get to base in seventy moves. Display message.

1830-1900 User has reached base. Display message.

1910-2020 User invited to replay.

Educational Notes

This program really suits the younger user more than the older, who though he/she may find it difficult, tends to consider it childish due to its simplistic representation. I have used it successfully with a whole class of visiting primary children where they have to decide collectively what the next move ought to be. Obviously it presupposes a knowledge of the compass, though you do not need to know the intermediate points of the compass to be able to succeed at the program. There is a cut off point at which you are considered to have taken too long.

You may wish to experiment with graph paper and pencil before using the program or set up a route using building blocks. Both these would act as good introductions to using the program.

Program Listing

```
10 REM *****
20 REM **
30 REM ** Written by Ian Clarke. **
40 REM ** Feb 83. **
50 REM *****
60 ON ERROR GOTO 1910
70 ENVELOPE 1,3,3,4,5,5,4,3,1,-1,2,-2,120,90
80 ENVELOPE 2,3,-3,-4,-5,5,4,3,1,-1,2,-2,120,90
90 CLEAR
100 MODE 7
110 VDU 23;8202;0;0;0;
120 PRINT
121 *FX4,1
130 DIM A(25),B(25),C(25),D(25)
140 PRINT TAB(8,20);CHR$(133);"To continue press a key."
150 *FX15
160 D=GET
170 CLS
180 PROCdisplay
190 PROCinput
200 PROCcheck
210 IF Z=7001 THEN GOTO 190
220 PROCmove
230 IF G>70 THEN PROCend:GOTO260
240 IF Y=HY AND X=1=HX THEN PROCright:GOTO 260
```

```

250 GOTO 190
260 FOR Z=1 TO 6000:NEXT
270 GOTO 1910
271 REM -----
272 REM Displays the four walls
273 REM the base and your
274 REM position on the screen.
280 DEFPROCdisplay
290 PRINT TAB(1,0);"N"
300 PRINT TAB(0);"W E"
310 PRINT TAB(1);"S"
320 FOR Y=3 TO 23
330 PRINT TAB(9,Y);CHR$(145);CHR$(47)
340 A(Y)=1
350 NEXT
360 S=RND(8)+8
370 PRINT TAB(10,S);" "
380 A(S)=0
390 C=S-(RND(6)+1)
400 Q=S+(RND(6)+1)
410 FOR Y=C TO Q
420 PRINT TAB(13,Y);CHR$(145);CHR$(47)
430 C(Y)=1
440 NEXT
450 FOR Y=3 TO 23
460 PRINT TAB(35,Y);CHR$(145);CHR$(47)
470 B(Y)=1
480 NEXT
490 S=RND(8)+8
500 PRINT TAB(36,S);" "
510 B(S)=0
520 D=S-(RND(8)+1)
530 R=S+(RND(8)+1)
540 FOR Y=D TO R
550 PRINT TAB(32,Y);CHR$(145);CHR$(47)
560 D(Y)=1
570 NEXT
580 XP=RND(6)
590 YP=RND(6)+11
600 PRINT TAB(XP,YP);CHR$(145);CHR$(172)
610 X=XP
620 Y=YP
630 XD=XP
640 YD=YP
650 HX=37
660 HY=RND(6)+11
670 PRINT TAB(HX,HY);CHR$(146);CHR$(136);CHR$(126)
679 REM .....
680 ENDPROC
681 REM -----
682 REM Input user's desired
683 REM direction and how many
684 REM times in that direction
690 DEFPROCinput
700 PRINT TAB(8,24);CHR$(134);"Which direction? ";
710 *FX15
720 C$=GET$
730 IF C$<>"N" AND C$<>"E" AND C$<>"S" AND C$<>"W" THEN G
OTO 720
740 PRINT TAB(26,24);C$;
750 *FX15
760 C1$=GET$
770 IF ASC(C1$)=13 THEN GOTO 840
780 IF C$="W" OR C$="E" THEN GOTO 760
790 IF C1$<>"W" AND C1$<>"E" THEN GOTO 760
800 PRINT C1$;
810 C$=C$+C1$
820 D=GET
830 IF D<>13 THEN GOTO 820

```

```

840 PRINT TAB(9,24);CHR$(133);"How many times ";C$;"? ";
850 T$=""
860 *FX15
870 D$=GET$
880 IF ASC(D$)=13 THEN GOTO 930
890 IF D$<"0" OR D$>"9" THEN GOTO 870
891 IF LEN(T$)=3 THEN GOTO 870
900 T$=T$+D$
910 PRINT TAB(28,24);T$;
920 GOTO 870
930 T=VAL(T$)
940 FOR Z=1 TO 500:NEXT
950 IF T=0 THEN GOTO 700
960 IF LEFT$(C$,1)="N" THEN X2=0:Y2=-1
970 IF C$="E" THEN X2=1:Y2=0
980 IF LEFT$(C$,1)="S" THEN X2=0:Y2=1
990 IF C$="W" THEN X2=-1:Y2=0
1000 IF LEN(C$)=1 THEN GOTO 1030
1010 IF C1$="E" THEN X2=1
1020 IF C1$="W" THEN X2=-1
1029 REM .....
1030 ENDPROC
1031 REM -----
1032 REM Check that user's move
1033 REM are in bounds.
1040 DEFPROCcheck
1050 Z=0
1060 REPEAT
1070 Z=Z+1
1080 XD=XD+X2
1090 YD=YD+Y2
1100 IF XD<0 OR XD>38 THEN UNTIL Z>0:PROCoutofbounds:GOTO
1130
1110 IF YD<3 OR YD>23 THEN UNTIL Z>0:PROCoutofbounds:GOTO
1130
1120 UNTIL Z=T
1129 REM -----
1130 ENDPROC
1131 REM Move the position of man
1132 REM to user's desired point.
1140 DEFPROCmove
1150 Z=0
1160 REPEAT
1170 Z=Z+1
1180 G=G+1
1200 SOUND 2,-12,80,1
1210 PRINT TAB(20,0);CHR$(141);CHR$(130);"Tries=";G
1220 PRINT TAB(20);CHR$(141);CHR$(130);"Tries=";G
1230 X=X+X2
1240 Y=Y+Y2
1250 PRINT TAB(X-X2+1,Y-Y2);CHR$(32)
1260 PRINT TAB(HX,HY);CHR$(146);CHR$(136);CHR$(126)
1270 IF X<>9 THEN GOTO 1290
1280 IF A(Y)=1 THEN PROCbump:GOTO 1350
1290 IF X<>35 THEN GOTO 1310
1300 IF B(Y)=1 THEN PROCbump:GOTO 1350
1310 IF X<>13 THEN GOTO 1330
1320 IF C(Y)=1 THEN PROCbump:GOTO 1350
1330 IF X<>32 THEN GOTO 1350
1340 IF D(Y)=1 THEN PROCbump
1350 IF X=10 AND A(Y)=1 THEN PRINT TAB(9,Y);CHR$(145);CH
R$(47);CHR$(172):GOTO 1410
1360 IF X=36 AND B(Y)=1 THEN PRINT TAB(35,Y);CHR$(145);C
HR$(47);CHR$(172):GOTO 1410
1370 IF X=14 AND C(Y)=1 THEN PRINT TAB(13,Y);CHR$(145);C
HR$(47);CHR$(172):GOTO 1410
1380 IF X=33 AND D(Y)=1 THEN PRINT TAB(32,Y);CHR$(145);C
HR$(47);CHR$(172):GOTO 1410
1390 IF G=71 THEN UNTIL Z>0:GOTO 1450

```

```

1400 PRINT TAB(X,Y);CHR$(145);CHR$(172)
1410 UNTIL Z=1
1420 XD=X
1430 YD=Y
1440 Z=T
1449 REM .....
1450 ENDPROC
1460 REM -----
1470 REM The user's position takes
1480 REM the man out of bounds.
1490 REM Display message.
1500 DEFPROCoutofbounds
1510 SOUND 1,2,120,50
1520 PRINT TAB(0,24);CHR$(130);CHR$(136);"This position is
out of bounds.";
1530 FOR Z=1 TO 7000:NEXT
1540 PRINT TAB(1,24);"
"
1550 XD=X
1560 YD=Y
1570 REM .....
1580 ENDPROC
1590 REM -----
1600 REM The user has just bumped
1610 REM into a wall. Display message.
1620 DEFPROCbump
1630 SOUND 1,1,85,60
1640 G=G+(T-Z)
1650 PRINT TAB(20,0);CHR$(141);CHR$(130);"Tries=";G
1660 PRINT TAB(20);CHR$(141);CHR$(130);"Tries=";G
1670 PRINT TAB(0,24);CHR$(131);CHR$(136);"You have just bu
mped into the wall.";
1680 FOR Z=1 TO 8000:NEXT
1690 PRINT TAB(1,24);"
"
1700 X=X-X2
1710 Y=Y-Y2
1720 Z=T
1729 REM .....
1730 ENDPROC
1731 REM -----
1732 REM The user's 70 moves
1733 REM are over.
1740 DEFPROCend
1750 FOR Z=1 TO 2000:NEXT
1760 CLS
1770 FOR Z=0 TO 1
1780 PRINT TAB(5,8+Z);CHR$(141);CHR$(131);"Sorry but you
didn't get"
1790 PRINT TAB(5,10+Z);CHR$(141);CHR$(131);"to the base
in 70 moves."
1800 NEXT
1810 REM .....
1820 ENDPROC
1821 REM -----
1822 REM The user has found
1823 REM the base.
1830 DEFPROCright
1840 FOR Z=1 TO 2000:NEXT
1850 CLS
1860 FOR Z=0 TO 1
1870 PRINT TAB(5,8+Z);CHR$(141);CHR$(131);"You got to th
e base in ";G;" moves."
1880 NEXT
1890 REM .....
1900 ENDPROC
1910 CLS
1920 PRINT TAB(5,15);"Do you want another game?"
1930 *FX15

```

```
1940 YN$=GET$
1950 CLS
1960 IF YN$="Y" THEN RUN
1970 PRINT TAB(10,10);CHR$(141);CHR$(131);"BYE!!!"
1980 PRINT TAB(10);CHR$(141);CHR$(131);"BYE!!!"
1990 FOR X=1 TO 3000:NEXT
2000 *FX4
2010 MODE 7
2020 END
```