

8. Light Bikes

General Description

This game is just for fun. The user instructions are displayed when the program is run. The object of the game is to survive for as long as you possibly can. You must imagine that you are riding a light bike which produces a wall behind you as you progress around the screen. If you crash into one of these walls both you and the wall will vaporise. Good luck!

Detailed Description

Lines 11-12 Displays title and asks how many players are going to play, also this is the first part of the delete trick.

14-18 It first goes to the procedure to set up variables then to the procedure to display the game board, then to the procedure to get movement from the keyboard.

Line 16: if computer is still playing then jump to procedure to make his move.

Line 17: go to procedure to move all the players.

19-25 Procedure for displaying title, and jumps to procedure to ask user who is going to play.

If $L = 0$ then left player is playing.

If $R = 0$ then right player is playing.

If $CP = 0$ then computer is playing.

26-34 Sets up variables.

SP = speed.

LX, LY are the coordinates of left player.

RX, RY are the coordinates of right player.

CX, CY are the coordinates of the computer.

LDX, LDY are the direction of the left player.

RDX, RDY are the direction of the right player.

CDX, CDY are the direction of the computer.

LD, RD are the direction of each player (they are used to stop user turning back on himself!).

35-47 Described in Lines 19-25

48-55 Display game board.

56-66 Get user movement from keyboard.

For left player

↑
W
←A D→
X
↓

For right player

↑
P
←L :→
/
↓

67-88 Make computer choose move.

Lines 70-73: the random number Q decides which direction he will choose if he has not hit a wall.

Line 74: Jumps to procedure to see if he might hit a wall if he goes that way.

Line 75: If he hasn't hit a wall then end procedure.,

Line 76-82: Evasive manoeuvres.

84-90 Procedure to check if computer will hit wall.

91-119 Procedure to move players.

Lines 93-95: If player is still alive then add direction to his movement.

Lines 96-100: If left player is still alive check to see if he has hit an invisible trail left by another player crashing, if so, ignore it, otherwise he has hit something, set L = -1.

Lines 101-109: Similar to Lines 96-100, but for different players.

Lines 110-112: If player has just crashed e.g. L = -1 or R = -1 or CP = -1 then make their wall invisible and make a sound, also let L or R or CP = 1.

Lines 113-115: If players are still alive then put them on the screen.

Lines 116-118: Check to see if there is a winner.

120-126 If winner then say who it is.

130-240 Display instructions

250-280 Second part of delete trick.

290-360 Credits.

Program Listing

```
10 MODE7
11 PROC_TI:ONERRORGOTO13
12 GOTO130
13 PROC_SE:MODE5
14 PROC_GB
15 PROC_KEY
16 IFCP=0PROC_CO
17 PROC_move
18 GOTO15
19 DEFPROC_TI
20 TITLE$="LIGHT BIKES"
21 PRINTTAB(10,10);CHR$141;CHR$130;TITLE$
22 PRINTTAB(10);CHR$141;CHR$130;TITLE$
23 A$=INKEY$(300)
24 PROC_ins
25 ENDPROC
26 DEFPROC_SE
27 SP=8:CRASH=0:LD=4:RD=3
28 LX=120:LY=500
29 RX=1100:RL=LY
30 CX=600:CY=150
31 LDX=SP:LDY=0
32 RDX=-SP:RDY=0
33 CDX=0:CDY=SP
34 ENDPROC
35 DEFPROC_ins
36 CLS
37 CLS
38 PRINTTAB(1,10)"Are two players going to play (Y/N)";
39 A$=GET$:IFA$<>"Y"ANDA$<>"N"THEN39
40 IFA$="Y"THENR=0:L=0 ELSE L=0:R=1
41 IFR=1THENC=0:GOTO45
42 PRINTTAB(1,12)"Do you want me to play (Y/N)";
43 A$=GET$:IFA$<>"Y"ANDA$<>"N"THEN43
44 IFA$="Y"THENC=0ELSECP=1
45 PRINTTAB(5,15);"Press any key to begin"
46 A$=GET$
47 ENDPROC
48 DEFPROC_GB
49 GCOL0,1
50 MOVE100,100
51 DRAW100,923
52 DRAW1179,923
53 DRAW1179,100
54 DRAW100,100
55 ENDPROC
56 DEFPROC_KEY
57 A$=INKEY$(0)
58 IFA$="W"ANDLD<>2LDX=0:LDY=SP:LD=1
59 IFA$="X"ANDLD>1LDX=0:LDY=-SP:LD=2
60 IFA$="A"ANDLD<4LDX=-SP:LDY=0:LD=3
61 IFA$="D"ANDLD<>3LDX=SP:LDY=0:LD=4
62 IFA$="P"ANDRD<>2RDX=0:RDY=SP:RD=1
63 IFA$="/"ANDRD>1RDX=0:RDY=-SP:RD=2
64 IFA$="L"ANDRD<4RDX=-SP:RDY=0:RD=3
65 IFA$=":"ANDRD<>3RDX=SP:RDY=0:RD=4
66 ENDPROC
67 DEFPROC_CO
68 CRASH=0
69 Q=RND(30)
70 IFQ=2CDX=0:CDY=SP
71 IFQ=4CDX=0:CDY=-SP
72 IFQ=6CDX=SP:CDY=0
73 IFQ=8CDX=-SP:CDY=0
74 PROC_TEST
```

```

75 IFCRASH=0THEN83
76 CDX=-SP:CDY=0:PROC_TEST
77 IFCRASH=0THEN83
78 CDX=SP:CDY=0:PROC_TEST
79 IFCRASH=0THEN83
80 CDX=0:CDY=SP:PROC_TEST
81 IFCRASH=0THEN83
82 CDX=0:CDY=-SP:PROC_TEST
83 ENDPROC
84 DEFPROC_TEST
85 C=POINT(CX+CDX,CY+CDY)
86 CRASH=0
87 IFC=0ENDPROC
88 IFC=2ANDL=1THENCRAH=0:ENDPROC
89 IFC=3ANDR=1THENCRAH=0:ENDPROC
90 CRASH=1:ENDPROC
91 DEFPROC_move
92 SOUND0,-6,6,1
93 IFL=0THENLX=LX+LDX:LY=LY+LDY
94 IFR=0THENRX=RX+RDX:RY=RY+RDY
95 IFCP=0THENCX=CX+CDX:CY=CY+CDY
96 IFL=1THEN101
97 C=POINT(LX,LY)
98 IFR=1ANDC=3THEN101
99 IFCP=1ANDC=1THEN101
100 IFC<>0THENL=-1
101 IFR=1THEN106
102 C=POINT(RX,RY)
103 IFL=1ANDC=2THEN106
104 IFCP=1ANDC=1THEN106
105 IFC<>0THENR=-1
106 IFCP=1THEN110
107 C=POINT(CX,CY)
108 IF(L=1ANDC=2)OR(R=1ANDC=3)THEN110
109 IFC<>0THENC=-1
110 IFL=-1THENVDU19,2,0,0,0,0:L=1:SOUND1,-15,123,2
111 IFR=-1THENVDU19,3,0,0,0,0:R=1:SOUND1,-15,150,2
112 IFCP=-1THENVDU19,1,0,0,0,0:CP=1:SOUND1,-15,200,2
113 IFL=0THENGCOL0,2:PLOT69,LX,LY:DRAWLX-LDX,LY-LDY
114 IFR=0THENGCOL0,3:PLOT69,RX,RY:DRAWRX-RDX,RY-RDY
115 IFCP=0THENGCOL0,1:PLOT69,CX,CY:DRAWCX-CDX,CY-CDY
116 IFL=1ANDR=1 E=1:GOTO120
117 IFCP=1ANDR=1 E=2:GOTO120
118 IFCP=1ANDL=1 E=3:GOTO120
119 ENDPROC
120 AS=INKEY$(100)
121 CLS
122 PRINTCHR$20''''
123 IFE=1PRINT"Computer has won."
124 IFE=2PRINT"Left player has won."
125 IFE=3PRINT"Right player has won."
126 END
130 MODE7
140 PRINT'" The idea of this game is to survive."
150 PRINT'"You can survive by not letting your"
160 PRINT"light-bike crash into a wall."
170 PRINT'"As you move, you leave a wall behind"
180 PRINT"you."
190 PRINT'"If you hit a wall, then the wall you"
200 PRINT"have made will vapourize, along with you."
210 PRINT"Controls are:-"
220 PRINT" W P "
230 PRINT" A D L : "
240 PRINT" X / "
250 PRINT'"Press Function key 'f0'"
260 *KEY0DELETE130,400|MRUN|M
270 IFINKEY(-33)=0THEN270
280 END
290 *****

```

```
300 *           LIGHT BIKES           *
310 *   DESIGNED AND WRITTEN BY       *
320 *           Andrew Pusey.         *
330 *           FEB 1983               *
340 *****
350 DO NOT RENUMBER THIS PROGRAM
360
```

