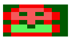
Pick a Path Red

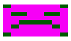
Introduction

Red has to set out on a journey, which will determine if they are successful or not depending on which Path they take, however Red may also encounter Aggressive Residents which spring out in front of you, if you try and fight them it will be fatal for Red, however if Red chooses to Dodge around, the Aggressive Resident only has 50/50 chance of striking out at Red, though Red if struck, Red can only takes a couple of blows.

Meet the Characters


Red

Red Under Attack


Aggressive Resident

Wall

Loading the Game

Written in Winape 2.0 Alpha 18, this game will work on any CPC (464,664 & 6128) with Disk Drive: Select 'File', Drive 'A', Insert Disk Image, 'Pick a Path Red.dsk' (it maybe necessary to change to the directory the dsk file is in) to open, in the emulator type RUN"disc to start the game.

This game can RUN through a number of Amstrad CPC Emulators, including CPCBox which can handle DSK image files.

Controls

Use the Cursor Keys Left, Right and Up Arrow to move Red forwards.

Making this game

I've written this game for the BASIC 10 Liners Competition for the EXTREM-256 category, the game has some DATA in it (not Machine Code), which is converted back to construct the Path ways. During the initialization phase, the Aggressive Residents are randomibly scattered around the Path Ways, which are represents by the number 2s and at the end numbers 3,4 & 5 are used to determine if Red has travelled the correct path or is simply been unlucky. I've been describing this game as a Role-Play type game as all the Luck needs to go Red to be successful.

Problems Encountered

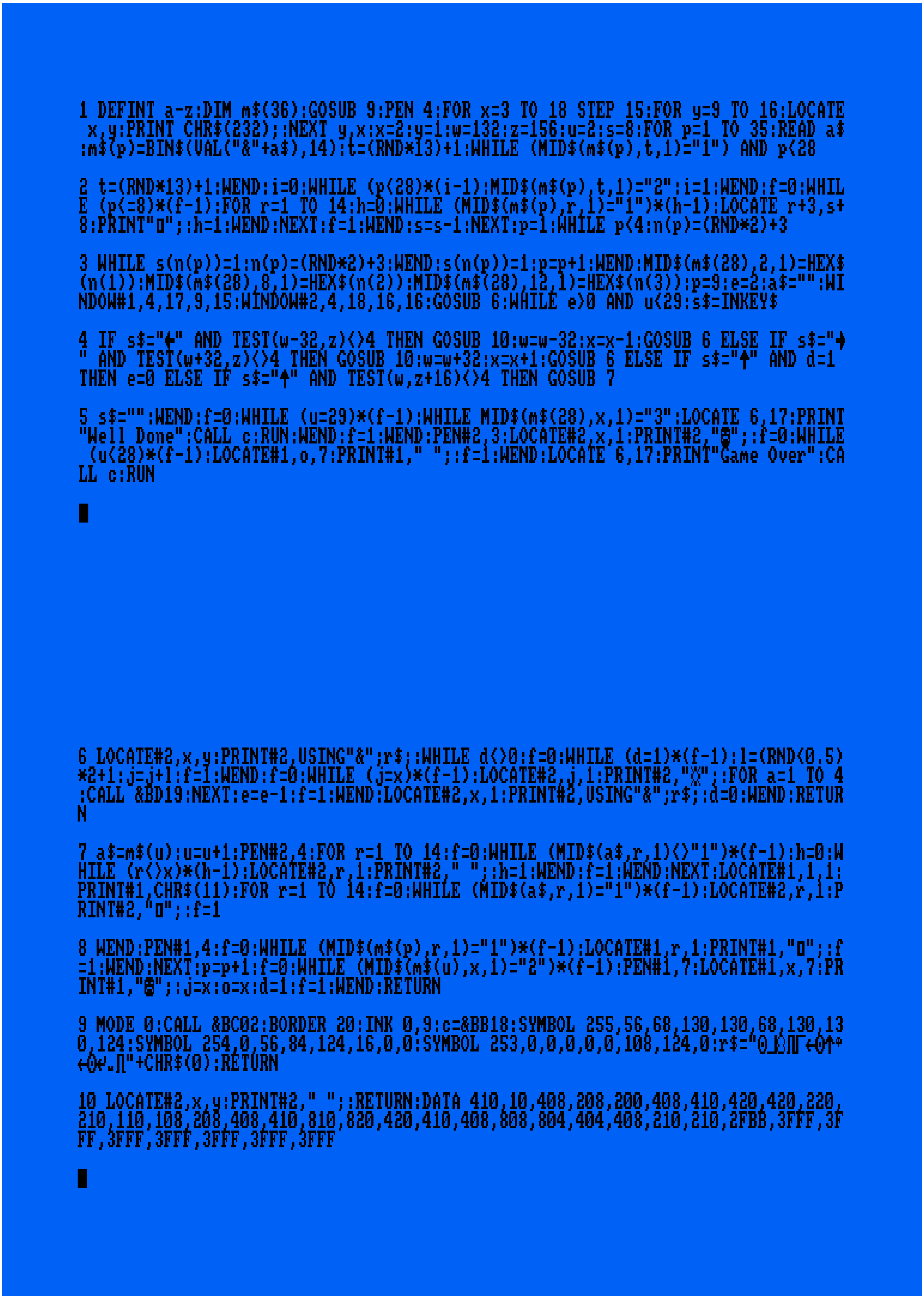
Thanks to the Atariage forum, I could improve on some of my BASIC coding habits and save some keystrokes, which has allowed me to simplify a number of WHILE..WEND statements to act the same way as IF..THEN statements.
I also needed a routine to select 1s or -1s, so when an Aggressive Resident pops up, if the player presses Left or Right has the Aggressive Resident decided to attack where Red decides to move and if so, to take action. Thanks to the Atariage forum again for finding some code which works on my Amstrad CPC.
Another problem I had relates to the 3,4 & 5 numbers I have to select at the end of the Path. Each number is Randomibly selected and then placed on 3 places, however numbers can only appear once and after looking at a similiar routine I'd used on another game I made, worked out a way, so numbers are only selected once, this is critical in changing the value at the end of each path and where the luck comes in as part of the game.
Update: After initially releasing the game to the BASIC 10-Liners Contest, the movement of the scenery I felt was a bit odd because the bottom window was updated before the other scenery was moved. I made some alterations, so old scenery is removed, scenery is moved down, followed by updating the scenery along the bottom, then the top. I think this looks better, but had to insert another FOR loopin the process. I also noticed my game was too easy at the end as no Aggressive Residents would appear, I then realized I was bumping into the more distant ones instead of the ones from the next line up, which was easy enough to fix.

Example of the game

11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
14111113111511
00001000010200
00001000012000
00210000001000
00010020000100
00100000200100
00100000201000
00010000201000
00012000010000
00010002100000
00100000100002
00100000010020
00012000010000
00010020001000
00201000001000
00000100001200
02000100010000
00021000010000
00001020100000
00010000100200
20010000100000
00210000010000
02010000001000
00001000002000
00021000001000
00010020001000
00000000210000
00010020010000

# Key '0' = Space, '1' = Wall, '2' Aggressive Resident, '3' Safe Exit, '4' Unsafe Exit, '5' Unsafe Exit #2

10 Liner Source Code (from screenshots)



```
1 DEFINT a-z:DIM m$(36):GOSUB 9:PEN 4:FOR x=3 TO 18 STEP 15:FOR y=9 TO 16:LOCATE
 x,y:PRINT CHR$(232);:NEXT y,x:x=2:y=1:w=132:z=156:u=2:s=8:FOR p=1 TO 35:READ a$
:m$(p)=BIN$(VAL("&"+a$),14):t=(RND*13)+1:WHILE (MID$(m$(p),t,1)="1") AND p<28

2 t=(RND*13)+1:WEND:i=0:WHILE (p<28)*(i-1):MID$(m$(p),t,1)="2":i=1:WEND:f=0:WHIL
E (p<=8)*(f-1):FOR r=1 TO 14:h=0:WHILE (MID$(m$(p),r,1)="1")*(h-1):LOCATE r+3,s+
8:PRINT"□";:h=1:WEND:NEXT:f=1:WEND:s=s-1:NEXT:p=1:WHILE p<4:n(p)=(RND*2)+3

3 WHILE s(n(p))=1:n(p)=(RND*2)+3:WEND:s(n(p))=1:p=p+1:WEND:MID$(m$(28),2,1)=HEX$
(n(1)):MID$(m$(28),8,1)=HEX$(n(2)):MID$(m$(28),12,1)=HEX$(n(3)):p=9:e=2:a$="":WI
NDOW#1,4,17,9,15:WINDOW#2,4,18,16,16:GOSUB 6:WHILE e>0 AND u<29:s$=INKEY$

4 IF s$="←" AND TEST(w-32,z)<>4 THEN GOSUB 10:w=w-32:x=x-1:GOSUB 6 ELSE IF s$="→
" AND TEST(w+32,z)<>4 THEN GOSUB 10:w=w+32:x=x+1:GOSUB 6 ELSE IF s$="↑" AND d=1
THEN e=0 ELSE IF s$="↑" AND TEST(w,z+16)<>4 THEN GOSUB 7

5 s$="":WEND:f=0:WHILE (u=29)*(f-1):WHILE MID$(m$(28),x,1)="3":LOCATE 6,17:PRINT
"Well Done":CALL c:RUN:WEND:f=1:WEND:PEN#2,3:LOCATE#2,x,1:PRINT#2,"☻";:f=0:WHILE
 (u<28)*(f-1):LOCATE#1,o,7:PRINT#1," ";:f=1:WEND:LOCATE 6,17:PRINT"Game Over":CA
LL c:RUN

█

6 LOCATE#2,x,y:PRINT#2,USING"&";r$;:WHILE d<>0:f=0:WHILE (d=1)*(f-1):l=(RND<0.5)
*2+1:j=j+l:f=1:WEND:f=0:WHILE (j=x)*(f-1):LOCATE#2,j,1:PRINT#2,"※";:FOR a=1 TO 4
:CALL &BD19:NEXT:e=e-1:f=1:WEND:LOCATE#2,x,1:PRINT#2,USING"&";r$;:d=0:WEND:RETUR
N

7 a$=m$(u):u=u+1:PEN#2,4:FOR r=1 TO 14:f=0:WHILE (MID$(a$,r,1)<>"1")*(f-1):h=0:W
HILE (r<>x)*(h-1):LOCATE#2,r,1:PRINT#2," ";:h=1:WEND:f=1:WEND:NEXT:LOCATE#1,1,1:
PRINT#1,CHR$(11):FOR r=1 TO 14:f=0:WHILE (MID$(a$,r,1)="1")*(f-1):LOCATE#2,r,1:P
RINT#2,"□";:f=1

8 WEND:PEN#1,4:f=0:WHILE (MID$(m$(p),r,1)="1")*(f-1):LOCATE#1,r,1:PRINT#1,"□";:f
=1:WEND:NEXT:p=p+1:f=0:WHILE (MID$(m$(u),x,1)="2")*(f-1):PEN#1,7:LOCATE#1,x,7:PR
INT#1,"☻";:j=x:o=x:d=1:f=1:WEND:RETURN

9 MODE 0:CALL &BC02:BORDER 20:INK 0,9:c=&BB18:SYMBOL 255,56,68,130,130,68,130,13
0,124:SYMBOL 254,0,56,84,124,16,0,0:SYMBOL 253,0,0,0,0,0,108,124,0:r$="⊙⊥⌂⊓┌←⊙↑ᵠ
←⊙↵.⊓"+CHR$(0):RETURN

10 LOCATE#2,x,y:PRINT#2," ";:RETURN:DATA 410,10,408,208,200,408,410,420,420,220,
210,110,108,208,408,410,810,820,420,410,408,808,804,404,408,210,210,2FBB,3FFF,3F
FF,3FFF,3FFF,3FFF,3FFF,3FFF

█
```

Listing

```
100 ' Pick a Path Red (Listing Version with Comments)
110 ' Set variables to Integer, Setup a String Array m$ and Start Initializing Screen
Colours and Sprite
120 DEFINT a-z
130 DIM m$(36)
140 GOSUB 1150
150 ' Commence drawing screen sides, draw only the first 8 lines.
160 PEN 4
170 FOR x=3 TO 18 STEP 15
180   FOR y=9 TO 16
190     LOCATE x,y
200     PRINT CHR$(232);
210 NEXT y,x
220 x=2
230 y=1
240 gx=132
250 gy=156
260 bx=2
270 p3=8
280 ' Read the rest of the screen data and use BIN$ to store 0s & 1s into array
290 FOR p=1 TO 35
300   READ a$
310   m$(p)=BIN$(VAL("&"+a$),14)
320     ' Start positioning Aggressive Resident, but don't position if a wall
330     ' is found (while check @ 290), or if p is greater than 28.
340     p4=(RND*13)+1
350     WHILE (MID$(m$(p),p4,1)="1") AND p<28
360       p4=(RND*13)+1
370     WEND
380     f3=0
390     ' This positions Aggressive Resident into final place.
400     WHILE (p<28)*(f3-1)
410       MID$(m$(p),p4,1)="2"
420       f3=1
430     WEND
440   f=0
450   ' Begin Drawing in Final Pathways, which is 14x8 in size.
460   WHILE (p<=8)*(f-1)
470     FOR p2=1 TO 14
480       f2=0
490       WHILE (MID$(m$(p),p2,1)="1")*(f2-1)
500         LOCATE p2+3,p3+8
510         PRINT CHR$(232);
520         f2=1
530       WEND
540     NEXT p2
550     f=1
560   WEND
570   p3=p3-1
580 NEXT p
590 ' Select 3 random numbers 3,4 & 5, but do not complete until all 3 numbers
600 ' are used. The s array is setup here to make that possible.
610 p=1
620 WHILE p<4
630   n(p)=(RND*2)+3
640   WHILE s(n(p))=1
650     n(p)=(RND*2)+3
660   WEND
670   s(n(p))=1
680   p=p+1
690 WEND
700 ' Position random numbers to their final positions at the end of the path
710 MID$(m$(28),2,1)=HEX$(n(1))
720 MID$(m$(28),8,1)=HEX$(n(2))
730 MID$(m$(28),12,1)=HEX$(n(3))
740 p=9
750 e=2
760 a$=""
770 ' Setup Game Screen
```

```
780 WINDOW#1,4,17,9,15
790 WINDOW#2,4,18,16,16
800 PEN#2,3
810 GOSUB 1260
820 ' Main Game Loop
830 WHILE e>0 AND bx<29
840   s$=INKEY$
850   IF s$=CHR$(242) AND TEST(gx-32,gy)<>4 THEN GOSUB 1240:gx=gx-32:x=x-1:GOSUB 1260
860   IF s$=CHR$(243) AND TEST(gx+32,gy)<>4 THEN GOSUB 1240:gx=gx+32:x=x+1:GOSUB 1260
870   IF s$=CHR$(240) AND d=1 THEN e=0 ELSE IF s$=CHR$(240) AND TEST(gx,gy+16)<>4 THEN
GOSUB 1340
880   s$=""
890 WEND
900 ' The game has ended one way or another.
910 f=0
920 WHILE (bx=29)*(f-1)
930   WHILE MID$(m$(28),x,1)="3"
940     LOCATE 6,17
950     PRINT"Well Done"
960     CALL &BB18
970     RUN
980   WEND
990   f=1
1000 WEND
1010 PEN#2,3
1020 LOCATE#2,x,1
1030 PRINT#2,CHR$(225);
1040 f=0
1050 WHILE (bx<28)*(f-1)
1060   LOCATE#1,o,7
1070   PRINT#1," ";
1080   f=1
1090 WEND
1100 LOCATE 6,17
1110 PRINT"Game Over"
1120 CALL &BB18
1130 RUN
1140 ' Initial Setup, Screen Mode, Border Colour, Background Colour & Sprites
1150 MODE 0
1160 CALL &BC02
1170 BORDER 20
1180 INK 0,9
1190 SYMBOL 255,56,68,130,130,68,130,130,124
1200 SYMBOL 254,0,56,84,124,16,0,0
1210 SYMBOL 253,0,0,0,0,0,108,124
1220
r$=CHR$(15)+CHR$(3)+CHR$(255)+CHR$(22)+CHR$(1)+CHR$(8)+CHR$(15)+CHR$(11)+CHR$(254)+CHR$(8
)+CHR$(15)+CHR$(13)+CHR$(253)+CHR$(22)+CHR$(0)
1230 RETURN
1240 LOCATE#2,x,y:PRINT#2," ";:RETURN
1250 ' Draw Red and depending on what d value holds because of an aggressive resident
being found, line 1140 works out the direction of attack and carry out attack in line
1150 if Red has gone the attack direction.
1260 LOCATE#2,x,y:PRINT#2,USING"&";r$;
1270 WHILE d<>0
1280 f=0:WHILE (d=1)*(f-1):l=(RND<0.5)*2+1:x2=x2+l:f=1:WEND
1290 f=0:WHILE (x2=x)*(f-1):LOCATE#2,x2,1:PRINT#2,CHR$(238);:FOR a=1 TO 4:CALL
&BD19:NEXT:e=e-1:f=1:WEND:LOCATE#2,x,1:PRINT#2,USING"&";r$;
1300 d=0
1310 WEND
1320 RETURN
1330 ' Move the Scenery, this works by updating the scenery in the 2 windows defined. Red
is in Window2, which needs to have the scenery moved into it as Red moves forward,
where's Window1 can have the scenery moved down (CHR$(11)) as the game moves along
1340 a$=m$(bx)
1350 bx=bx+1
1360 PEN#2,4
1370 FOR p2=1 TO 14
1380   f=0
1390   WHILE (MID$(a$,p2,1)<>"1")*(f-1)
1400     f2=0
```

```
1410     WHILE (p2<>x)*(f2-1)
1420       LOCATE#2,p2,1
1430       PRINT#2," ";
1440       f2=1
1450     WEND
1460     f=1
1470   WEND
1480 NEXT
1490 LOCATE#1,1,1
1500 PRINT#1,CHR$(11)
1510 FOR p2=1 TO 14
1520   f=0
1530   WHILE (MID$(a$,p2,1)="1")*(f-1)
1540     LOCATE#2,p2,1
1550     PRINT#2,CHR$(232);
1560     f=1
1570   WEND
1580   PEN#1,4
1590   f=0
1600   WHILE (MID$(m$(p),p2,1)="1")*(f-1)
1610     LOCATE#1,p2,1
1620     PRINT#1,CHR$(232);
1630     f=1
1640   WEND
1650 NEXT p2
1660 p=p+1
1670 f=0
1680 WHILE (MID$(m$(bx),x,1)="2")*(f-1)
1690   PEN#1,7
1700   LOCATE#1,x,7
1710   PRINT#1,CHR$(225);
1720   x2=x
1730   o=x
1740   d=1
1750   f=1
1760 WEND
1770 RETURN
1780 ' Data for the Pathways.
1790 DATA 410,10,408,208,200,408,410,420,420,220,210,110,108,208,408
1800 DATA 410,810,820,420,410,408,808,804,404,408,210,210,2fbb,3fff
1810 DATA 3fff,3fff,3fff,3fff,3fff,3fff
```