ELECTRIC FENCING

This program started out as a version of 'Pong'. However, it got changed half way through its development and now only a whiff of the original idea remains. Electric fencing is a game for two players using either a pair of JY1 joysticks or the keyboard. The object of the game is to score 'hits' on the other player while dodging his attacks. Even though it's written in Basic, the game is quite fast and furious and should provide a few minutes entertaining play. The game contains some useful re-usable routines and interesting sound effects.

The listing is divided into sections to make it easier to understand and illustrate its structure. You are recommended to type in each section and then check it with the listing. Some sections can be left out initially as they not essential to the main programme.
These are:

The title screen. Lines 1490-1610. Replace this whole section with

```
1490 RETURN
```

The joystick/keyboard selection screen. Lines 280-360 Replace this section wi th

```
280 j=0:RETURN'if you only
    wish to use the keyboard
```

If you don't intend to use joysticks then you can dispense with the joystick reading routine. Lines 400-470.

Finally, the 'GAME OVER' routine. Lines 940-1030. Replace this with

```
940 STOP
```

When debugging your program remember that an error (especially `improper argument`) may not be in the line it shows up in. This is something that often confuses people. The computer can 'see' that you're not carrying your umbrella but it can't tell where you left it. If something gets printed in the wrong place, you may have left a semicolon out of the `print` statement. Don't confuse I for 1.

VARIABLES

`p1` and `p2` stand for player one and player two. Used to prefix these variables requiring a separate record for each player;

`dir` for direction `sa` for sword activated `sc` for score `wp` for window position

These other variables are used as switches to alter the flow of control through the program.

`j` for joystick or keyboard control. -1 (or true) for joysticks, 0 for keys. `pwpe` for players window positions are equal.

The string variable `a$(n)` contains the big characters 0-9. Where `n` is the required character.
`pt` for temporary pointer

THE ROUTINES

50-110 Goto the various routines that set up the program

140-240 Main loop that calls the routines to test the keys and update the screen.

280-360 Choose joystick or keyboard control

400-470 Test the joysticks

500-560 Test the keyboard

600-620 and 660-680 Update the players positions on the screen.

720-740 and 780-800 Draw players 'swords'.

840-900 Test for either player being hit and update score board.

940-1030 Game over

1070-1210 Set up string array. Change 1's and 0's into blocks and spaces.

1240-1340 Setup other constants.

1380-1450 Reset variables and screen at start of action.

1490-1610 Title screen.

When you run the program, you should get the title screen appearing and then after a pause you should see the joystick/keyboard option come up. If you want to use joysticks, press `j` and the asterisk should move to the joystick position. Press **[ENTER]** to make your choice. The title screen should come up once more and then the game starts. If you are using the keyboard the controls are:

PLAYER1	PLAYER2
up a	up 6
down z	down 3
fire x	fire 7

When either player gets a hit on the other player, his score goes up by one and this is displayed in large characters on the screen. To add to the excitement, the sound of a generator can be heard in the background and various other noises should be heard. When a player's score reaches nine, he wins and a key must be pressed to start again.

There are many ways in which you could alter the game. You could perhaps allow the players to move in any direction. Put obstacles on the screen to hide behind. Limit the amount of energy available for moving and shooting or include a high score table.

LISTING

```
10 REM ********************************
   *****************
20 REM *** Programs from AMSTRAD CPC464
   User Magazine ***
30 REM **********        FENCING       *
   *****************
40 REM ********************************
   *****************
50 DEFINT a-z
60 MODE 0
70 GOSUB 1070
80 GOSUB 1490
90 GOSUB 280
100 GOSUB 1490
110 GOSUB 1380
120 '
130 '
140 REM start
150 IF finished THEN GOTO 100
160 GOSUB 240
170  CALL &BD19:IF p1dir THEN GOSUB 600
    ELSE CALL &BD19:CALL &BD19
180  CALL &BD19:IF p2dir THEN GOSUB 660
    ELSE CALL &BD19:CALL &BD19
190 IF p1sa=-1 THEN GOSUB 720
200 IF p2sa=-1 THEN GOSUB 780
210 GOTO 140
220 '
230 '
240 IF j THEN  400 ELSE  500
250 '
260 '
270 '
280 CLS:PEN 6
290 PRINT:PRINT"   CHOOSE CONTROL"
300 PRINT:PRINT:PRINT:PRINT" press J K o
    r ENTER"
310 LOCATE 4,10:PRINT"JOYSTICK";TAB(5);"
    OR KEYS"
320 LOCATE 12,10:IF j THEN PRINT"*":ELSE
     PRINT" "
330 LOCATE 12,11:IF j THEN PRINT" ":ELSE
     PRINT"*"
340 IF NOT(INKEY(45)) THEN j=-1
350 IF NOT(INKEY(37)) THEN j=0
360 IF NOT(INKEY(18)) THEN RETURN ELSE 3
    20
370 '
380 '
390 '
400 p1=JOY(0):p2=JOY(1)
410 p1dir=(p1 AND 1)*-1+(p1 AND 2)*0.5
420 p2dir=(p2 AND 1)*-1+(p2 AND 2)*0.5
430 IF P1 AND 16 THEN p1sa=p1sa-1:IF p1s
    a=-1 THEN AFTER 15 GOSUB 840
440 IF P2 AND 16 THEN p2sa=p2sa-1:IF p2s
    a=-1 THEN AFTER 15 GOSUB 840
450 IF p1sa THEN p1dir=0
460 IF p2sa THEN p2dir=0
470 RETURN
480 '
490 '
500 p2dir=((INKEY(4)=0)*1)+((INKEY(5)=0)
    *-1)
510 p1dir=((INKEY(69)=0)*1)+((INKEY(71)=
    0)*-1)
520 IF INKEY(63)=0 THEN p1sa=p1sa-1:IF p
    1sa=-1 THEN AFTER 15 GOSUB 840
530 IF INKEY(10)=0 THEN p2sa=p2sa-1:IF p
    2sa=-1 THEN AFTER 15 GOSUB 840
540 IF p1sa THEN p1dir=0
550 IF p2sa THEN p2dir=0
560 RETURN
570 '
580 '
590 '
600 pt=p1wp+p1dir:IF pt>25 OR pt<6 THEN
    RETURN ELSE p1wp=pt
610 p1dir=0
620 PEN 1:LOCATE 3,p1wp:CLS #3:PRINT CHR
    $(209);:RETURN
630 '
640 '
650 '
660 pt=p2wp+p2dir:IF pt>25 OR pt<6 THEN
    RETURN ELSE p2wp=pt
670 p2dir=0
680 PEN 2:LOCATE 18,p2wp:CLS #5:PRINT CH
    R$(211);:RETURN
690 '
700 '
710 '
720 PAPER #4,4:WINDOW #4,4,17,p1wp,p1wp:
    CLS#4:CALL &BD19:CALL &BD19
730 PAPER #4,0:CLS#4
740 GOTO 600
750 '
760 '
770 '
780 PAPER #6,5:WINDOW #6,4,17,p2wp,p2wp:
    CLS#6:CALL &BD19:CALL &BD19
790 PAPER #6,0:CLS#6
800 GOTO 660
810 '
820 '
830 '
840 pwpe=(p1wp=p2wp):IF p1sa AND NOT(p2s
    a) AND pwpe THEN p1sc=p1sc+1:SOUND 1
    32,120,10,0,1,0:PRINT#1,a$(p1sc);:IF
     p1sc=9 THEN 940
850 IF p2sa AND NOT(p1sa) AND pwpe THEN
    p2sc=p2sc+1:SOUND 132,100,10,0,1,0:P
    RINT#2,a$(p2sc);:IF p2sc=9 THEN 940
860 IF p1sa THEN SOUND 132,40,70,0,1,1
870 IF p2sa THEN SOUND 132,56,70,0,1,1
880 p1sa=0
890 p2sa=0
900 RETURN
```

```
910 '
920 '
930 '
940 PEN 6
950 LOCATE 6,10:PRINT"GAME OVER"
960 IF p1sc=9 THEN INK 1,2,20:INK 2,0 EL
    SE INK 2,6,17:INK 1,0
970 SOUND 129,1000,0,12,3:SOUND 130,900,
    0,12,3
980 WHILE INKEY$<>"":WEND
990 t!=TIME:WHILE t!+2000>TIME:WEND
1000 WHILE INKEY$="":WEND
1010 CLS
1020 finished=-1
1030 RETURN
1040 '
1050 '
1060 '
1070 a$(0)="1111011011011111"
1080 a$(1)="0110010010011001"
1090 a$(2)="1110011111100111"
1100 a$(3)="1110011111001111"
1110 a$(4)="1001001011111001"
1120 a$(5)="1111001111001111"
1130 a$(6)="1111001111011111"
1140 a$(7)="1110010010100101"
1150 a$(8)="1111011111011111"
1160 a$(9)="1111011111001001"
1170 FOR n=0 TO 9
1180  howlong=LEN(a$(n))
1190  FOR n2=1 TO howlong
1200    IF MID$(a$(n),n2,1)="1"THEN MID$
     (a$(n),n2,1)=CHR$(143)ELSE MID$(a$(
     n),n2,1)=CHR$(32)
1210 NEXT n2,n
1220 '
1230 '
1240 b$="ELECTRIC FENCING"
1250 c$=CHR$(32)+CHR$(164)+" Alexander M
     artin"
1260 ENV 1,=9,2000:ENT -1,6,3,1
1270 ENV 2,127,0,0,127,0,0,127,0,0,127,0
     ,0,127,0,0
1280 ENV 3,=9,9000
1290 '
1300 '
1310 '
1320 BORDER 0
1330 PEN #4,1:PEN #6,2:PEN #1,1:PEN #2,2
     :PAPER #1,3:PAPER #2,3:PEN #0,6
1340 RETURN 'FROM SETTING UP CONSTANTS
1350 '
1360 '
1370 '
1380 INK 0,12:INK 1,2:INK 2,6:INK 3,13:I
     NK 4,20:INK 5,17:INK 6,20
1390 WINDOW #3,3,3,6,25:WINDOW #5,18,18,
     6,25
1400 WINDOW #1,3,5,1,5:WINDOW #2,16,18,1
     ,5:WINDOW #7,1,20,1,5:PAPER #7,3
1410 CLS:CLS#7:PRINT#1,a$(0);:PRINT#2,a$
     (0);:p1sc=0:p2sc=0:p1wp=5:p2wp=24:p
     1dir=1:p2dir=1
1420 GOSUB 600:GOSUB 660
1430 SOUND 1,1000,0,12,2:SOUND 2,900,0,1
     2,2
1440 p1sa=0:p2sa=0:finished=0
1450 RETURN 'FROM GAME SHEET RESTORE
1460 '
1470 '
1480 '
1490 CLS
1500 PEN 7
1510 FOR n=1 TO LEN(b$)
1520  LOCATE 2+n,10
1530  FOR n2=LEN(b$) TO n STEP-1
1540   PRINT MID$(b$,n2,1)
1550   LOCATE 2+n,10
1560   SOUND 135,20*n2,5,12,2,1
1570 NEXT n2,n
1580 SOUND 135,100,0,13,3,1,20
1590 PEN 6:PRINT:PRINT:PRINT:PRINT c$
1600 t!=TIME:WHILE t!+2500>TIME:WEND
1610 RETURN
```