5. Vector Race Game

General Description

This program is meant to show how vectors can be used to direct the position of a car round a race track. General instructions are seen when the program is started. The names of the players are requested, and a warning is given that the vectors are 'cumulative' and indicate 'speed' and direction. This means that next position is determined by your previous vector. The game is single key stroke entry, except when putting in your names, when the return key must be used. The 'delete' has not been allowed when selecting your vector . . . ie get it right. The vectors are displayed vertically without the enclosing brackets. You might wish to enhance this.

When playing the game the person whose go it is and his/her previous vector are displayed. Most textual messages occur at the top of the screen, except for crash and collision which occur in the centre of the screen and generate alarming sounds. The track of the 'car' is left as a white 'echo' on the screen and though it is theoretically possible, as the program is written, to crash on an echo, no third year has yet done so.

Detailed Description

Lines 10-310 The main structure of the program: nothing surprising except that I have used three flags on lines 280 and 290 when I could have squeezed it down to two. The only important point is that 'CAR = CAR + 1' must go at line 230 and not earlier or later as PROCgetvector assumes CAR values 0 and 1 rather than 1 and 2.

320-530 The 'X' and 'Y' values in line 370 will hold the previous position of the car. The array 'A$' had to be

dimensioned earlier as it also holds the names of the contestants - indeed it is used for several functions.

540-700 The race track is drawn by overlaying three text and graphics windows to give the yellow track surrounded by red. It is quicker than triangular fill.

710-890 ' UD'stands for up/down and ' LR'stands for left/ right. The moves are counted to stop someone cheating by reversing straight over the finish line (see below). Lines 790-830 over-plot the old position of the car in white. If you find the ' bug'which I suspect is there but cannot generate, then line 790 will have to become GCOL0,2 and you will lose the white track.

900-1280 Information is displayed in the text window at the top of the screen in lines 980-1030. Lines 1040-1250 actually collect the vector from the player. The vector is validated and restricted to a number between -9 and 9. The vector is collected as a single character and translated into a number subsequently.

1290-1420 If you change the start positions, then you will need to change the start message at line 690.

1430-1580 The car is checked for off limits by a colour test for yellow at line 1480. The collision check from 1490-1520 sees if the cars are within 12 pixels of each other. On the basis of experiments, this seemed acceptable, but all that needs to be changed is the value 12 in line 1520 if you are not satisfied. The victory check assumes that each car has done at least 15 moves. It is possible to get round in 15 moves so the quick reverse freak has nothing to gain. The victory check makes certain that your final position is within reasonable limits.

1590-1900 Self-evident, and looking at it again, not quite perfect as there is only one statement' sdifference between the crash and collision routines. Still, it is easily readable.

1910-2460 The instructions here are free for you to change. I have noticed that the instructions do not say that vectors are limited from -9 to 9, but when I have loaded the game for the class, I have dropped a word in their ear and they seem to have had no problems.

2470-end The error trap you may leave off, but the data is the start data for the game.

Educational Notes

I have used this to teach vectors to low ability third years. The idea is developed from an old one which we do with pencil and paper and it helps to play the game on paper first to acquaint them with the concept of vectors. I do not recommend the game as a panacea for actual teaching, but once they have mastered the simple paper track, the entertainment value of the game shows its worth.

Because we do not have enough machines, they play in groups of four a mini-league between them. It would be possible to do a single class demonstration, but then some of the messages would need to be changed.

Program Listing

```
 10 REM ============================
 20 REM     The Vector Race game
 30 REM   by Ian Murray written for
 40 REM   the BBC Model B, February
 50 REM   1982.
 60 REM ........................
 70 REM
 80 REM ============================
 90 REM      Main Structure
100 REM ......................
110 REM
120 CLEAR
130 MODE 1:DIM A$(6):ON ERROR GOTO 2470
140 *KEY10OLD|MRUN|M
150 PROCintroduction
160 PROCinitialise
170 PROCdraw_screen
180 PROCstart_positions
190 REPEAT
200   CAR = 0
210   REPEAT
220     PROCgetvector(CAR)
230     CAR = CAR + 1
240     PROCchecklimits(CAR,X,Y)
250     PROCdraw_car(CAR,X,Y)
260   UNTIL Crash OR Victory OR Collide OR CAR = 2
270 UNTIL Crash OR Victory OR Collide
280 IF Collide THEN PROCcollide
290 IF Crash THEN PROCcrash ELSE PROCvictory
300 PROCfinal
310 END
320 REM ==========================
330 REM       initialise
340 REM ......................
350 REM
360 DEF PROCinitialise
370 CAR = 0:Moves = 0:X1=0:Y1=0:X2=0:Y2=0
380 Crash  =  FALSE
390 Victory = FALSE
400 Collide = FALSE
410 TITLE$ = "VECTOR RACE GAME"
420 ENVELOPE 1,2,-10,5,-10,20,10,20,30,-1,30,-1,150,30
430 GCOL 0,3
```

```
 440 GCOL 0,130
 450 COLOUR 3
 460 COLOUR 129
 470 DIM Position(5),Speed(5)
 480 A$(0) = " first"
 490 A$(1) = "second"
 500 FOR Z = 1 TO 4
 510   READ Position(Z)
 520 NEXT
 530 ENDPROC
 540 REM =========================
 550 REM       draw race track
 560 REM .......................
 570 REM
 580 DEF PROCdraw_screen
 590 VDU 28,0,11,36,31:CLS
 600 VDU 24,70;70;1200;650;
 610 CLG
 620 VDU 29,50;100;
 630 VDU 28,6,25,32,15
 640 CLS
 650 PRINT TAB(6,3) TITLE$
 660 GCOL 2,1
 670 MOVE 0,140
 680 DRAW 130,140
 690 PRINT TAB(0,10) "<< start & finish"
 700 ENDPROC
 710 REM =========================
 720 REM       plot car position
 730 REM .......................
 740 REM
 750 DEF PROCdraw_car(CAR,UD,LR)
 760 Moves = Moves + 1
 770 CAR = CAR + 41
 780 VDU 29,50;100;
 790 GCOL 0,3
 800 IF CAR = 42 THEN MOVE X1,Y1 ELSE MOVE X2,Y2
 810 VDU 5,CAR
 820 MOVE UD,LR
 830 GCOL 0,0
 840 VDU 5,CAR
 850 VDU 4
 860 GCOL 0,3
 870 IF CAR = 42 THEN X1=UD:Y1=LR
 880 IF CAR = 43 THEN X2=UD:Y2=LR
 890 ENDPROC
 900 REM =========================
 910 REM       get the vector
 920 REM .......................
 930 REM
 940 DEF PROCgetvector(CART)
 950 LOCAL NUMBER,Count,CAR
 960 CAR = 2*CART
 970 Count = 0
 980 VDU 28,0,9,39,0
 990 COLOUR 128:COLOUR 3
1000 CLS
1010 PRINT "It's "A$(CART+4)"'s turn to move the car"
1020 PRINT; "Your current speed is "Speed(CAR+1)/10
1030 PRINT;TAB(22) Speed(CAR+2)/10
1040 REPEAT
1050   NUMBER = FALSE
1060   PRINT TAB(0,5+Count) "Your vector please..."A$(Count
)" number ";
1070   B$ =""
1080   REPEAT
1090     REPEAT
1100       A$ = GET$
1110     UNTIL (A$>="0" AND A$<="9") OR A$="-"
```

```
 1120      IF B$="-" AND A$="-"THEN A$=""
 1130      B$=B$+A$
 1140      IF B$ <> "-" THEN NUMBER = TRUE
 1150      PRINT A$;
 1160    UNTIL NUMBER
 1170    SOUND 2,-15,150,4
 1180    PRINT
 1190    Count = Count + 1
 1200    PRINT
 1210    A$(1+Count) = B$
 1220    B$=""
 1230    Speed(CAR+Count) = Speed(CAR+Count) + 10 * VAL(A$(1+
Count))
 1240    Position(CAR+Count) = Position(CAR+Count) + Speed(CA
R+Count)
 1250 UNTIL Count = 2
 1260 X = Position(CAR+1)
 1270 Y = Position(CAR+2)
 1280 ENDPROC
 1290 REM ==========================
 1300 REM     start positions
 1310 REM ......................
 1320 REM
 1330 DEF PROCstart_positions
 1340 LOCAL Z
 1350 FOR Z = 0 TO 1
 1360   Z1 = 2*Z +1
 1370   Z2 = 2*Z +2
 1380   X = Position(Z1)
 1390   Y = Position(Z2)
 1400   PROCdraw_car(Z+1,X,Y)
 1410 NEXT
 1420 ENDPROC

 1430 REM ==========================
 1440 REM      check limits
 1450 REM ......................
 1460 REM
 1470 DEF PROCchecklimits(CAR,X,Y)
 1480 IF POINT(X,Y) <> 2 THEN Crash = TRUE
 1490 LOCAL A,B
 1500 IF CAR=1 THEN A = ABS(X-X2) ELSE A= ABS(X-X1)
 1510 IF CAR=2 THEN B = ABS(Y-Y2) ELSE B= ABS(Y-Y1)
 1520 IF A<12 AND B <12 THEN Collide = TRUE
 1530 IF Moves > 30 THEN PROCvictory_check
 1540 ENDPROC
 1550 REM ......................
 1560 DEF PROCvictory_check
 1570 IF X>0 AND X<125 AND Y>159 AND Y<259 THEN Victory = TR
UE:Winner = CAR
 1580 ENDPROC
 1590 REM ==========================
 1600 REM  victories and crashes
 1610 REM ......................
 1620 REM
 1630 DEF PROCcollide
 1640 SOUND 1,1,10,150
 1650 VDU 28,6,25,32,15
 1660 PRINT TAB(6,5) "*****************"
 1670 PRINT TAB(6,6) "< CARS COLLIDE >"
 1680 PRINT TAB(6,7) "*****************"
 1690 IF CAR = 1 THEN Winner = 2 ELSE Winner = 1
 1700 PROCvictory
 1710 ENDPROC
 1720 REM ......................
 1730 DEF PROCcrash
 1740 SOUND 1,1,10,150
 1750 VDU 28,6,25,32,15
 1760 PRINT TAB(6,5) "*****************"
```

```
1770 PRINT TAB(6,6) "< YOU  CRASHED >"
1780 PRINT TAB(6,7) "****************"
1790 IF CAR = 1 THEN Winner = 2 ELSE Winner = 1
1800 PROCvictory
1810 ENDPROC
1820 REM ........................
1830 DEF PROCvictory
1840 VDU 28,0,9,39,0
1850 CLS
1860 PRINT TAB(10,2) "WELL DONE!"
1870 PRINT:PRINT
1880 PRINT; "Car "Winner" has won this race on the track."
1890 PRINT "Congratulations to "A$(Winner+3)
1900 ENDPROC
1910 REM =========================
1920 REM      final
1930 REM ........................
1940 REM
1950 DEF PROCfinal
1960 LOCAL Finish
1970 Finish = TIME + 1000
1980 REPEAT
1990 UNTIL TIME > Finish
2000 VDU 26
2010 COLOUR 128
2020 CLS
2030 PRINT TAB(10,10) "Do you want another go ?";
2040 REPEAT
2050   REPLY$ = GET$
2060   *FX 15,0
2070 UNTIL REPLY$="Y" OR REPLY$="N"
2080 PRINT REPLY$
2090 IF REPLY$="N" THEN SOUND 2,-15,30,8:END ELSE RUN
2100 ENDPROC
2120 REM =========================
2130 REM     introduction
2140 REM ........................
2150 REM
2160 DEF PROCintroduction
2170 CLS
2180 COLOUR 1
2190 PRINT TAB(10,3) "VECTOR  RACE  GAME!"
2200 COLOUR 3
2210 PRINT TAB(5,5) "This is a game for two players."
2220 PRINT TAB(5) "You will each control a vector"
2230 PRINT TAB(5) "car  ... car 1 is a star and"
2240 PRINT TAB(5) "car 2 is a plus sign."
2250 PRINT:PRINT
2260 PRINT TAB(5) "The speed of the car is given"
2270 PRINT TAB(5) "by a vector !!"
2280 PRINT TAB(5) "FOR EXAMPLE .. speed vector 3"
2290 PRINT TAB(32) "-3"
2300 PRINT TAB(5) "will move the car to a new"
2310 PRINT TAB(5) "position 3 to the right"
2320 PRINT TAB(5) "and 3 down."
2330 COLOUR 1:PRINT
2340 PRINT "WARNING";:COLOUR 3
2350 PRINT "  speeds are cumulative!"
2360 COLOUR 2
2370 PRINT:PRINT "Your track will be left in white"
2380 PRINT"on the screen."
2390 PRINT:PRINT
2400 INPUT"First player's name please "A$(4)
2410 INPUT"Second player's name please "A$(5)
2420 COLOUR 3:PRINT
2430 PRINT"WHEN YOU ARE READY TOUCH ANY KEY"
2440 *FX15,0
2450 REP$=GET$
2460 ENDPROC
```

```
2470 IF ERR=17 GOTO 2000
2480 MODE1
2490 PRINT; "Programmer error "ERR" at line "ERL

2500 PRINT"My apologies .. please tell me how the"
2510 PRINT"error occurred !!"
2520 DATA 25,160,85,160
```