FOOTU: FOO Revisited

A Game For The OSI C1P, or how we learned the true meaning of the oft used phrase "This program is easily adapted to..."

Charles M. and Michael J. De Santis

On p. 26:45 of the July 1980 issue of **MICRO,** the "small systems jornal" from Ohio Scientific listed a little race program called "FOO". It was stated that the program would run on disk based OSI machines but that "the program is easily adapted to" OSI basic-in-ROM machines. Well, maybe its easy if you're one of OSI's computer designers or software whizzes and know where all the goodies are tucked away inside all the OSI computers, but my son Mike and I had one devil of a weekend getting "FOO" to run on our diskless — C1P. However, I can't say it was a bad experience because we learned a lot about our little machine and have come up with a couple of things that should be of interest to other C1P owners as well.

A Carriage Control

For instance, did you know that SPC (0) when used in a PRINT statement causes about 15 line feeds to occur. We discovered this one while trying to figure out why the roadway on OSI's version of "FOO" would space out and break up occasionally (see their line 550).

Keyboard Control Routine

After that was corrected, our next problem was to get the vehicle in the game moving under keyboard control. We found that, for some reason that we didn't want to take a lot of time to discover, the subroutine starting at line 600 of the OSI version of the game wouldn't work on the C1P as the program was originally written.

To correct this problem, we just re-wrote the subroutine using the "more standard" format from the OSI graphics manual, i.e. POKE 57088, row #: IF PEEK (57088) = col. # THEN ...etc. However, our keyboard control software evolved into a form that we think is really useful for many other programs.

In the typical game program as in "FOO", numbers, i.e. number keys, are used to control the direction of an object on the video screen, e.g. "1" for movements to the left and "2" for movements to the right. A problem with this approach usually crops up at the end of a game if, for instance, an INPUT statement is used to query the user about continuing. If the player isn't fast enough (he's just been controlling a space ship and has crashed into a star at 30,000 mi/hr.) he enters a "1" or "2" where a "Y" or an "N" was expected, and he has to fuss around to correct the entry or restart the program if he's already hit the RETURN. The more insidious version of this problem arises when the "keyboard-control-during-program-execution" feature is turned off while you're still holding down the "1" or "2" key. This situation usually arises abrubtly because of a game rule violation of some sort. The game stops and control returns to BASIC. This happens so fast that you're still holding down one of the number control keys, and BASIC interprets this to be the entry of a program line number. If you type anything else and then hit the RETURN you've just added a new line to your program; and you won't know it until the next time you try to run it. My favorite error in this regard ends up with line 1 reading: 1 LIST. When the program is run, I get a listing.

Well here's how to fix things so that the problem never happens again. First of all, don't use numbers for control functions (obvious, right?); we've used the left and right shift keys for control for several reasons: (a) they're spaced a nice distance apart for hand control; and (b) they're both accessible using the same row number in the keyboard polling routine.

Secondly, and this is where the serendipity comes in, the SHIFT LOCK key must be released in order for the SHIFT keys to be activated since it is also accessed through the same row number. In our version of "FOO", after all of the game options are selected, we use instructions such as:

```
270 PRINT "TO START, RELEASE SHIFT LOCK"
271 POKE 57088, 254: IF PEEK (57088) = 254 THEN
    270
```

The "254" is the column number of the SHIFT LOCK key on the polled keyboard so that line 271 keeps getting repeated until the SHIFT LOCK is released. As soon as it is released, the game starts and the shift keys are active. If the game should end abruptly or unexpectedly, and keys that may have been pressed are not entered because the RETURN key is inactive while the SHIFT LOCK key is not depressed.

The SHIFT LOCK must be pressed in order for BASIC to respond. At the end of the game or at any intermediate INPUT statement, we print a reminder to "PUSH THE SHIFT LOCK" for the proper data entry or to restore normal operation. It's a great way to do it! Try it, you'll like it.

FOOTU — C1P Version

Listing 1 is our version of FOO modified to run on our C1P which has 4k of memory. Some of the scaling factors of the original program have been eliminated and the SHIFT and SHIFT LOCK features discussed in this article are employed. The display has been scaled to fit the C1P's capabilities. For other machines, lines 110, 230, 240, 290 and

520 may have to be modified. Also lines 600 – 660 will have to be modified for C2 and C4 computers. Just remember .... "This program is easily adapted to ..." Good Luck!

```
FOOTU
 100 POKE 530, 1
 110 KY=57088:SM=2:MS=1:RN=0
 115 BS=54051
 117 ML=0:SN=255
 120 LP=5
 130 PL=2/LP
 155 KP=0
 160 IF A$="Y" THEN ME=EM:WI=WF:GU=UG:
     GOTO 270
 170 FOR I=1 TO 30:PRINT:NEXT I
 180 PRINT "FOOTU"
 190 PRINT:PRINT"RACEWAY"
 200 PRINT:PRINT"YOU RUN AT YOUR OWN
     RISK!"
 210 PRINT:PRINT"LEFT=LEFT SHIFT RIGHT=
     RT SHIFT"
 215 PRINT:PRINT"OVERDRIVE=CTRL"
 230 PRINT: INPUT"INITIAL WIDTH (1-20)";WI
 240 PRINT:INPUT"DELAY(1-15)";ME
 241 EM=ME
 245 PRINT
 250 GU=0:INPUT"PEDESTRIANS (Y/N)";X$:
     IF LEFT$(X$,1)="Y"THEN GU=.3
 260 KP=0: INPUT "KILLER FOO (Y/N)"; X$:
     IF LEFT$(X$,1)="Y" THEN PK=1
 270 PRINT: PRINT "TO START PRESS SHIFT
     LOCK"
 271 POKE KY, 254: IF PEEK (KY)=254 THEN 271
 280 FOR I=1 TO 30: PRINT:NEXT I
 290 WD=WI:WF=WI: WI=(12-WI) /2
 291 ME=54060-ME*32
 300 FOR M=1 TO LP; GOSUB 600: GOSUB 500:
     ML=ML+1: NEXT M
 350 WI=WI-1
 370 IF WI 4 THEN 300
 400 FOR M=1 to LP: GOSUB 600: GOSUB 500:
     ML=ML+1: NEXT M
 450 WI=WI+1
 470 IF WI<WD THEN 400
 490 GOTO 300
 500 RN=RN+SM*RND (1)–MS
 510 WT=WT+SGN(RN)
 520 IF WI+WT>20 THEN WT=WT-1: RN=0
 530 IF WT<0 OR WT=0 THEN WT=WT-1: RN=0
 540 IF WI<0 THEN WI=2
 545 IF WI>8 AND RND (1)<GU THEN POKE
      BS+WT+1+INT (WI*RND(1) ),  240
 550 PRINT SPC (WT); "XX"; SPC (WI); "XX"
 560 RETURN
 600 POKE Y, 254
 610 IF PEEK (KY) =251 THEN ME=ME–1KK=–1
 620 IF PEEK (KY) =253 THEN ME=ME+1:KK=1
 630 IF PEEK (KY) =191 THEN ME=ME+KK
 640 IF PEEK (ME)<>32 THEN 700
 650 POKE ME, C
 660 RETURN
 700 IF PEEK (ME) =240 THEN GY=240
 705 IF GY=240 AND PK THEN KP=KP+1:
     GY=0: GOTO 650
 720 PRINT "YOU BLEW IT!"
 725 PRINT
 730 MI=ML*PL
 750 PRINT "AFTER"; MI;"MILES"
 755 IF PK THEN PRINT "AND"; KP; "KILLS"
 757 PRINT: PRINT "TOTAL POINTS:";
     INT(MI+4*(1–PK) *MI + 100*KP)
 760 GOSUB 1000
 810 GOTO 5000
1000 IF PK THEN WD=KP: GOTO 1030
1010 WD=MI/WF
1030 PRINT: PRINT "CONGRATULATIONS"
1040 PRINT "YOU MAY NOW CALL"
1045 PRINT "YOURSELF"
1050 PRINT: PRINT "       "
1060 IF WD<3 THEN PRINT "LITTLE"; GOTO 1200
1070 IF WD<5 THEN PRINT "TENDER";:
     GOTO 1200
1080 IF WD<12.5 THEN PRINT "MEDIOCRE";:
     GOTO 1200
1099 IF WD<25 THEN PRINT "BIG";: GOTO 1200
1100 IF WD<38 THEN PRINT "MASTER";:
     GOTO 1200
1110 IF WD<50 THEN PRINT "GRAND";:
     GOTO 1200
1120 PRINT "CHEATER"
1200 PRINT "FOO"
1210 IF GY=240 THEN PRINT "KILLER!"
1220 PRINT "!"
1230 RETURN
5000 PRINT: PRINT: PRINT "PRESS SHIFT LOCK"
5001 PRINT: INPUT "AGAIN"; A$: A$= LEFT$
     (A$,1)
5010 IF A$<>"Y" THEN 6000
5020 INPUT "SAME"; A$:A$= LEFT$ (A$,1)
5025 IF A$<>"Y" THEN CLEAR
5030 GOTO 100
6000 END
```

©