Graphics

3-D Graphics

These programs (variations of the same program, using different data) allow you to create and rotate three-dimensional figures. To manipulate the image, proceed as follows:

The cursor keys move the object in the direction indicated by the arrows on the keys.

f5)
f6) used to modify the perspective of the image

f7 moves the image so that the object is parallel with your line of sight (this will be clear when you run the program)

f9 returns the object to starting position

The programs were written by Andrew Herron. The first version shows the letters REW, on a grid, being rotated

```
10 REM     3D GRAPHICS PACKAGE
20
30 REM     COPYRIGHT  A.HERRON
35
36 REM    Real Exhausting Work !
40
50 ON ERROR GOTO 1020
60
70 GOTO 370:REM JUMP TO MAIN PROGRAM
80
90 REM **SUBROUTINES TO MANIPULATE**
       **IMAGE SIZE, ORIENTATION, **
```

```
          **ETC.
  100
  110 NR=NR-1:IF NR<0 THEN NR=120+NR
  120 RETURN
  130 NR=NR+1:IF NR>119 THEN NR=NR-120
  140 RETURN
  150 ZO=ZO*.95:RETURN
  160 ZO=ZO/.95:RETURN
  170 S=S*.7:RETURN
  180 S=S/.7:RETURN
  190 S=0:RETURN
  200 S=1:RETURN
  210 ZO=20:NR=0:S=1:S2=5000
  220 RETURN
  230
  240 REM COMPUTE & STORE IMAGE DATA
  250
  260 FOR I%=1 TO 78
  270   READ Q(I%,0,D),X,Y,Z
  280   P=(ZO+S*(X*T(NR,0)-Z*T(NR,1)))/S
2
  290   Q(I%,1,D)=(X*T(NR,1)+Z*T(NR,0))/
P
  300   Q(I%,2,D)=Y/P
  310   NEXT I%
  320 RESTORE
  330 RETURN
  340
  350 REM **START OF MAIN PROGRAM**
  360
  370 *TV255
  380 *FX 4 2
  390 *KEY12"1"
  400 *KEY13"2"
  410 *KEY14"3"
  420 *KEY15"4"
  430 *KEY5"5"
  440 *KEY6"6"
  450 *KEY7"7"
```

```
  460 *KEY8"8"
  470 *KEY9"9"
  480 *FX 11 0
  490 DIM T(119,1),Q(100,2,2)
  500 start=TRUE
  510 MODE 5:VDU29,640;512;
  520 VDU 23,0,10,32,0;0;0;
  530
  540 REM LOOK UP TABLE FOR SIN & COS
  550
  560 FOR I%=0 TO 119:T(I%,0)=SIN RAD(I%
*3)
  570   T(I%,1)=COS RAD(I%*3)
  580   NEXT I%
  590
  600 ALWAYS=TRUE:C=2:D=1
  610 GOSUB 210
  620
  630 REPEAT
  640
  650   REM **READ CONTROL KEYS**
  660
  670   PRINT TAB(0,0);"?";
  680   *FX 15 1
  690   IF start THEN start=FALSE:GOTO 7
10
  700   V%=VAL(GET$)
  710   PRINT TAB(0,0);" ";
  720   ON V%+1 GOSUB 120,110,130,150,16
0,170,180,190,200,210
  730
  740   REM **COMPUTE IMAGE DATA**
  750   GOSUB 260
  760
  770   REM **DRAW IMAGE**
  780   GCOL 1,D
  790   PROCDRAW(D)
  800
  810   REM *CHANGE DISPLAYED PAGE**
```

```
  820    VDU19,D,7;0;19,C,0;0;19,3,7;0;
  830
  840    REM **ERASE IMAGE**
  850    GCOL 2,D
  860    PROCDRAW(C)
  870
  880    E=C:C=D:D=E
  890    UNTIL ALWAYS=FALSE
  900  END
  910
  920  DEF PROCDRAW(H)
  930  REM DRAW IMAGES
  940
  950  FOR I%=1 TO 78
  960    PLOT Q(I%,0,H),Q(I%,1,H),Q(I%,2,
H)
  970    NEXT I%
  980  ENDPROC
  990
 1000  REM Handles ON ERROR GOTO ...
 1010
 1020  *FX 12
 1030  *FX 4
 1040  MODE 7
 1050  PRINT
 1060  REPORT
 1070  PRINT" at line ";ERL
 1080  END
 1090
 1100  REM DATA FOR IMAGE POINTS
 1110
 1120  REM **DATA FOR LOGO**
 1130  REM **FRONT FACE**
 1140  DATA 4,-1.4,-.5,0,5,-1.4,.5,0
 1150  DATA 5,-.7,.5,0,5,-.6,.4,0
 1160  DATA 5,-.6,0,0,5,-.7,-.05,0
 1170  DATA 5,-.6,-.1,0,5,-.6,-.5,0
 1180  DATA 5,-.8,-.5,0,5,-.8,-.1,0
 1190  DATA 5,-1.2,-.1,0,5,-1.2,-.5,0
```

```
1200 DATA 5,-1.4,-.5,0
1210 DATA 4,-1.2,.1,0,5,-1.2,.3,0
1220 DATA 5,-.8,.3,0,5,-.8,.1,0
1230 DATA 5,-1.2,.1,0
1240 REM LETTER E
1250 DATA 4,-.4,-.5,-.2,5,-.4,.5,-.2
1260 DATA 5,.4,.5,-.2,5,.4,.3,-.2
1270 DATA 5,-.2,.3,-.2,5,-.2,.1,-.2
1280 DATA 5,.4,.1,-.2,5,.4,-.1,-.2
1290 DATA 5,-.2,-.1,-.2,5,-.2,-.3,-.2
1300 DATA 5,.4,-.3,-.2,5,.4,-.5,-.2
1310 DATA 5,-.4,-.5,-.2
1320 REM LETTER W
1330 DATA 4,.7,-.5,-.4,5,.6,-.4,-.4
1340 DATA 5,.6,.5,-.4,5,.8,.5,-.4
1350 DATA 5,.8,-.3,-.4,5,.9,-.3,-.4
1360 DATA 5,.9,.1,-.4,5,1.1,.1,-.4
1370 DATA 5,1.1,-.3,-.4,5,1.2,-.3,-.4
1380 DATA 5,1.2,.5,-.4,5,1.4,.5,-.4
1390 DATA 5,1.4,-.4,-.4,5,1.3,-.5,-.4
1400 DATA 5,.7,-.5,-.4
1410 REM GRID
1420 DATA 4,-1.2,-.5,5,5,-1.2,-.5,-5
1430 DATA 4,-1.4,-.5,5,5,-1.4,-.5,-5
1440 DATA 4,-1.2,-.5,5,5,-1.2,-.5,-5
1450 DATA 4,-1.0,-.5,5,5,-1.0,-.5,-5
1460 DATA 4,-.8,-.5,5,5,-.8,-.5,-5
1470 DATA 4,-.6,-.5,5,5,-.6,-.5,-5
1480 DATA 4,-.4,-.5,5,5,-.4,-.5,-5
1490 DATA 4,-.2,-.5,5,5,-.2,-.5,-5
1500 DATA 4,0,-.5,5,5,0,-.5,-5
1510 DATA 4,.2,-.5,5,5,.2,-.5,-5
1520 DATA 4,.4,-.5,5,5,.4,-.5,-5
1530 DATA 4,.6,-.5,5,5,.6,-.5,-5
1540 DATA 4,.8,-.5,5,5,.8,-.5,-5
1550 DATA 4,1.0,-.5,5,5,1.0,-.5,-5
1560 DATA 4,1.2,-.5,5,5,1.2,-.5,-5
1570 DATA 4,1.4,-.5,5,5,1.4,-.5,-5
1580 DATA 4,1.6,-.5,5,5,1.6,-.5,-5
```

And now this version rotates the house image:

```
   10 REM    3D GRAPHICS PACKAGE
   20
   30 REM    COPYRIGHT  A.HERRON
   40 REM           HOUSE
   50
   60 ON ERROR GOTO 1030
   70
   80 GOTO 380:REM JUMP TO MAIN PROGRAM
   90
  100 REM **SUBROUTINES TO MANIPULATE**
         **IMAGE SIZE, ORIENTATION, **
         **ETC.
  110
  120 NR=NR-1:IF NR<0 THEN NR=120+NR
  130 RETURN
  140 NR=NR+1:IF NR>119 THEN NR=NR-120
  150 RETURN
  160 ZO=ZO*.95:RETURN
  170 ZO=ZO/.95:RETURN
  180 S=S*.7:RETURN
  190 S=S/.7:RETURN
  200 S=0:RETURN
  210 S=1:RETURN
  220 ZO=20:NR=0:S=1:S2=5000
  230 RETURN
  240
  250 REM COMPUTE & STORE IMAGE DATA
  260
  270 FOR I%=1 TO 36
  280   READ Q(I%,0,D),X,Y,Z
  290   P=(ZO+S*(X*T(NR,0)-Z*T(NR,1)))/S
2
```

```
  300    Q(I%,1,D)=(X*T(NR,1)+Z*T(NR,0))/
P
  310    Q(I%,2,D)=Y/P
  320    NEXT I%
  330 RESTORE
  340 RETURN
  350
  360 REM **START OF MAIN PROGRAM**
  370
  380 *TV255
  390 *FX 4 2
  400 *KEY12"1"
  410 *KEY13"2"
  420 *KEY14"3"
  430 *KEY15"4"
  440 *KEY5"5"
  450 *KEY6"6"
  460 *KEY7"7"
  470 *KEY8"8"
  480 *KEY9"9"
  490 *FX 11 0
  500 DIM T(119,1),Q(100,2,2)
  510 start=TRUE
  520 MODE 5:VDU29,640;512;
  530 VDU 23,0,10,32,0;0;0;
  540
  550 REM LOOK UP TABLE FOR SIN & COS
  560
  570 FOR I%=0 TO 119:T(I%,0)=SIN RAD(I%
*3)
  580    T(I%,1)=COS RAD(I%*3)
  590    NEXT I%
  600
  610 ALWAYS=TRUE:C=2:D=1
  620 GOSUB 220
  630
  640 REPEAT
  650
  660    REM **READ CONTROL KEYS**
```

```
  670
  680   PRINT TAB(0,0);"?";
  690   *FX 15 1
  700   IF start THEN start=FALSE:GOTO 7
20
  710   V%=VAL(GET$)
  720   PRINT TAB(0,0);" ";
  730   ON V%+1 GOSUB 130,120,140,160,17
0,180,190,200,210,220
  740
  750   REM **COMPUTE IMAGE DATA**
  760   GOSUB 270
  770
  780   REM **DRAW IMAGE**
  790   GCOL 1,D
  800   PROCDRAW(D)
  810
  820   REM *CHANGE DISPLAYED PAGE**
  830   VDU19,D,7;0;19,C,0;0;19,3,7;0;
  840
  850   REM **ERASE IMAGE**
  860   GCOL 2,D
  870   PROCDRAW(C)
  880
  890   E=C:C=D:D=E
  900   UNTIL ALWAYS=FALSE
  910 END
  920
  930 DEF PROCDRAW(H)
  940 REM DRAW IMAGES
  950
  960 FOR I%=1 TO 78
  970   PLOT Q(I%,0,H),Q(I%,1,H),Q(I%,2,
H)
  980   NEXT I%
  990 ENDPROC
 1000
 1010 REM Handles ON ERROR GOTO ...
 1020
```

```
1030 *FX 12
1040 *FX 4
1050 MODE 7
1060 PRINT
1070 REPORT
1080 PRINT" at line ";ERL
1090 END
1100
1110 REM DATA FOR IMAGE POINTS
1120
1130
1140 REM HOUSE
1150 REM WALLS
1160 DATA 4,1,1,1,5,1,-1,1,5,-1,-1,1
1170 DATA 5,-1,1,1,4,1,1,-1,5,1,-1,-1
1180 DATA 5,-1,-1,-1,5,-1,1,-1
1190 DATA 4,1,-1,-1,5,1,-1,1
1200 DATA 4,-1,-1,-1,5,-1,-1,1
1210
1220 REM DOOR
1230 DATA 4,-.2,-1,1.3,4,-.2,0,1.3
1240 DATA 85,.2,0,1.3,4,.2,-1,1.3
1250 DATA 85,-.2,-1,1.3
1260
1270 REM ROOF
1280 DATA 4,-1.1,1,1.1,5,1.1,1,1.1
1290 DATA 5,1.1,1,-1.1,5,-1.1,1,-1.1
1300 DATA 5,-1.1,1,1.1,5,-.8,1.5,0
1310 DATA 5,.8,1.5,0,5,1.1,1,1.1
1320 DATA 4,-.8,1.5,0,5,-1.1,1,-1.1
1330 DATA 4,.8,1.5,0,5,1.1,1,-1.1
1340
1350 REM WINDOW
1360 DATA 4,1,0,.5,5,1,.5,.5
1370 DATA 5,1,.5,-.5,5,1,0,-.5
1380 DATA 5,1,0,.5,4,1,0,0
1390 DATA 5,1,.5,0
1400
1410 REM  Data held as
```

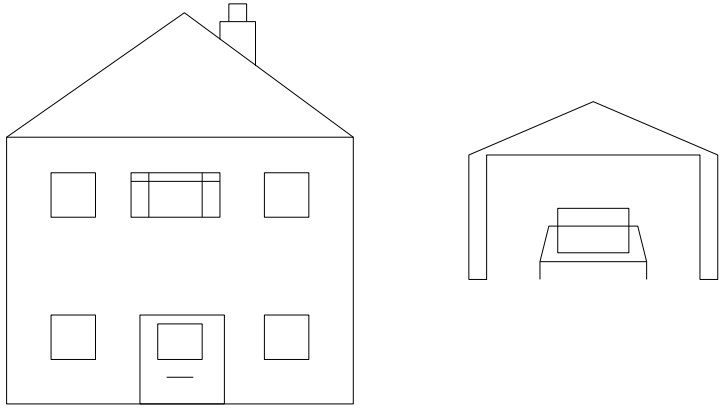
```
1420
1430 REM plot type & X,Y,Z co-ords
```

Draw 40

This program allows you to create complicated pictures on the TV screen, and then dump them to the MCP-40 Colour Printer/Plotter. There are many commands you can use:

G clears screen and variables
s slows the cursor movement down
f speeds it up
d medium speed
^ plots a point at the cursor position
- (minus sign) draws a line from the last point plotted
y selects yellow
a red
b black
w white
g changes the colour displayed on the screen (keep pressing the g until the colour you want is found)
| toggles dotted/solid lines
$ deletes last point plotted
% dumps output to plotter
h calls a help routine. This routine needs to be written by you, and is inserted from line 690 onwards.

Note that the use of upper and lower case for these commands is vital. That is, you must use upper or lower case as shown here. Once you become familiar with this program, you'll be pleased to see how effective the results of using it can be, as this sample indicates:



```
   10 REM DRAW40 - For MCP-40 Plotter
   20
   30 REM TESTED ON BASIC2 AND OS1.20
   40
   50   REM Data stored as !Variable sta
rting at t%. 64 Words stored.
   60   REM Bits 31-24=ctrl%; Bits 23-12
=xcoord%; Bits 11-0=ycoord%
   70   REM ctrl% Bits 31-30: 00=Mark, 0
1=Draw, 10=New Format, 11=Draw Dotted
   80   REM ctrl% Bits 29-28=Logical Col
our,Bits 27-24=Actual Colour
   90 *TV0,1
  100 MODE1
  110 DIMacol%(3):REM Actual colour for
Logical Colour
  120 recln%=255:DIMt% recln%:p%=0:mp%=0
:  REM Set Start of Data Store, p%=Point
er, recln%=Record Length
  130 modescl%=4
  140  c2%=2^12:c3%=2^24:c4%=2^4:c5%=2^6
:  REM Set Constants
  150 VDU23;8202;0;0;0;:  REM Switch off
 Cursor
  160 VDU24,0;0;1279;919;:VDU28,0,2,39,0
```

```
:CLS:CLG:  REM Set Windows
  170 VDU23,255,0,16,16,16,254,16,16,16:
  REM Define Sprite
  180 PROCinit:  REM Initialize
  190 cs$="sdf^-Gbaywg|#$%h":  REM Comma
nd String
  200
  210   REM **Control Loop**
  220 REPEAT PROCcommand:VDU4:CLS:PRINT
'xx%,yy%,"  Lcol=";fcolor%;:COLOURfcolor
%:PRINT"* ";:COLOUR0:PRINT"Acol=";acol%(
fcolor%);"  Pcol=";pcol%;:IFp%>recln% PR
OCsave
  230     VDU5:UNTIL FALSE:END
  240
  250
  260   REM **Control Loop PROC.**
  270 DEF PROCcommand
  280 LOCAL I%,J%
  290 get%=GET-&87
  300 ON get% GOSUB410,420,430,440 ELSE
GOTO330
  310 PROCcross(cmx%*scale%,cmy%*scale%)
:ENDPROC
  320   REM Decode Command
  330 in$=CHR$(get%+&87)
  340 FOR I%=1TO LEN(cs$)
  350   IFin$=MID$(cs$,I%,1)THEN J%=I%:I
%=&FF
  360   NEXT
  370 ON J% GOSUB450,460,470,480,490,500
,510,520,530,540,550,560,570,580,590,600
 ELSE GOSUB600
  380 ENDPROC
  390
  400   REM **Subroutines used in PROCco
mmand**
  410 cmx%=-1:cmy%=0:RETURN
  420 cmx%=1:cmy%=0:RETURN
```

```
  430 cmx%=0:cmy%=-1:RETURN
  440 cmx%=0:cmy%=1:RETURN
  450 scale%=modescl%:RETURN
  460 scale%=modescl%*4:RETURN
  470 scale%=modescl%*16:RETURN
  480 PROCmd(0,xx%,yy%):PROCscrn(1):RETU
RN
  490 PROCmd(1,xx%,yy%):PROCscrn(1):RETU
RN
  500 PROCinit:RETURN
  510 PROCcfg(0):RETURN
  520 PROCcfg(1):RETURN
  530 PROCcfg(2):RETURN
  540 PROCcfg(3):RETURN
  550 PROCclc:RETURN
  560 dot%=dot%EOR-1:RETURN
  570 PROCss:RETURN
  580 PROCera:RETURN
  590 PROCplt(0):RETURN
  600 PROCh:RETURN
  610
  620   REM**Initialize PROC.**
  630 DEF PROCinit
  640 LOCAL I%:FORI%=0TO3:acol%(I%)=2^I%
-1:NEXT
  650 CLG:ctrl%=0:lxx%=0:lyy%=0:xx%=636:
yy%=508:dot%=0:VDU5:fcolor%=3:@%=5:scale
%=modescl%*16:PROCpc:p%=0:pcol%=3:comwd%
=&37000000:VDU20:COLOUR0:COLOUR131:GCOL4
,0:*FX4,1
  660 ENDPROC
  670
  680   REM**'HELP' PROC.**
  690 DEF PROCh:ENDPROC
  700
  710   REM**Move Cross PROC.**
  720 DEF PROCcross(X%,Y%)
  730 PROCpc
  740 xx%=xx%+X%:yy%=yy%+Y%
```

```
  750 IFxx%<0 OR xx%>=c2% xx%=xx%-X%
  760 IFyy%<0 OR yy%>=c2% yy%=yy%-Y%
  770 PROCpc
  780 ENDPROC
  790
  800   REM**Print Cross PROC.**
  810 DEF PROCpc
  820 MOVExx%-12,yy%+16:VDU255
  830 ENDPROC
  840
  850   REM **Pack Instruction**
  860   REM X%,Y% Integers in Range 0 to
 4095 -- Z% in Range 0 to 255
  870 DEF FNpack(Z%,X%,Y%):LOCALW%
  880 IF(Z%AND&80) W%=&80000000 ELSE W%=
0
  890 =(Z%AND&7F)*c3%ORX%*c2%ORY%ORW%
  900
  910   REM **Unpack Instruction**
  920   REM ctrl% in Range 0 to 255 -- x
coord%, ycoord% in Range 0 to 4095
  930 DEF PROCunpack(X%):LOCALW%
  940 IFX%<0 W%=1 ELSE W%=0
  950 X%=X%AND&7FFFFFFF
  960 ctrl%=X%DIVc3%
  970 IFW% ctrl%=ctrl%OR&80
  980 xcoord%=(X%MODc3%)DIVc2%
  990 ycoord%=X%MODc2%
 1000 ENDPROC
 1010
 1020   REM ** Set up MARK/DRAW and Coor
dinates in comwd% & Store it in t%!p% **
 1030 DEF PROCmd(B%,X%,Y%):LOCAL Z%
 1040 IF B%ANDdot% B%=3
 1050 PROCunpack(comwd%)
 1060 Z%=ctrl%AND&3F OR (B%*c5%)
 1070 comwd%=FNpack(Z%,X%,Y%)
 1080 t%!p%=comwd%:mp%=p%:p%=p%+4
 1090 ENDPROC
```

```
 1100
 1110   REM ** Reset Colour in comwd% **
 1120 DEF PROCrsc(X%):LOCALZ%
 1130 PROCunpack(comwd%)
 1140 Z%=ctrl%AND&C0ORX%*c4%ORacol%(X%)
 1150 comwd%=FNpack(Z%,xcoord%,ycoord%)
 1160 ENDPROC
 1170
 1180   REM ** Change Logical Colour PRO
C.**
 1190   REM * <MARK>(^) to Leave *
 1200 DEF PROCclc:LOCALI%
 1210 FORI%=0TO15:VDU19,fcolor%,I%;0;:ac
ol%(fcolor%)=I%
 1220   get%=GET:IF get%=&67 NEXT:GOTO12
10 ELSE I%=16:NEXT
 1230 IFget%=94 PROCrsc(fcolor%):PROCmd(
0,xx%,yy%):ENDPROC ELSE VDU7,4:PRINT'"
 <MARK> (^) PLEASE":VDU5:get%=GET:GOTO12
30
 1240
 1250   REM ** Save Picture to Disc PROC
.**
 1260 DEF PROCsave p%=0:ENDPROC
 1270
 1280   REM ** Change Foreground Colour
**
 1290 DEF PROCcfg(X%):fcolor%=X%:PROCrsc
(X%):ENDPROC
 1300
 1310   REM ** Puts MARK or Draws Line o
n Screen from comwd% instruction **
 1320 DEF PROCscrn(V%):LOCALZ%,W%
 1330 PROCunpack(comwd%)
 1340 Z%=(ctrl%AND&C0)/c5%
 1350 dot%=0:IF Z%=0 W%=69 ELSE IF Z%=1
W%=5 ELSE W%=21:dot%=-1
 1360 fcolor%=(ctrl%AND&30)/c4%
 1370 acol%(fcolor%)=ctrl%AND&F
```

```
 1380 VDU19,fcolor%,acol%(fcolor%);0;
 1390 PROCpc:MOVElxx%,lyy%
 1400 PROCwrt(V%,W%)
 1410 lxx%=xcoord%:lyy%=ycoord%
 1420 ENDPROC
 1430
 1440   REM ** Single Step Proc.- Use #
to single step **
 1450 DEF PROCss
 1460 IFp%>mp% VDU7:ENDPROC
 1470 comwd%=t%!p%:p%=p%+4
 1480 IFctrl%AND&C0=&80 ENDPROC
 1490 PROCscrn(1)
 1500 ENDPROC
 1510
 1520   REM ** Write to Screen **
 1530 DEF PROCwrt(Z%,W%):LOCALV%:IFZ%=0
V%=0 ELSE V%=fcolor%
 1540 GCOL0,V%:PLOTW%,xcoord%,ycoord%:GC
OL4,0:PROCpc
 1550 ENDPROC
 1560
 1570   REM ** Erase Proc. - Use $ to Er
ase **
 1580 DEF PROCera
 1590 IFp%<8 PROCinit:GOTO220
 1600 p%=p%-4:mp%=mp%-4
 1610 comwd%=(t%!p%AND&FF000000)OR(t%!mp
%AND&FFFFFF)
 1620 PROCscrn(0)
 1630 comwd%=t%!mp%
 1640 ENDPROC
 1650
 1660   REM ** Plotter Driver for MCP-40

 1670 DEF PROCplt(SP%):LOCALS$,Z%,SCL: R
EM SP% gives Starting Value
 1680 S$=STRING$(4,""):S$=""
 1690 VDU4:CLS:PRINT"Plotter Type MCP-40
```

```
";:VDU5
 1700 SCL=2.6:*FX5,1
 1710 VDU2,21,1,18,72: REM Set up Plotte
r
 1720 p%=SP%:pcol%=-1
 1730 REPEAT
 1740   PROCunpack(t%!p%):p%=p%+4
 1750   ON(ctrl%AND&C0)/c5%+1 GOSUB1820,
1830,1840,1850
 1760   IFS$="0" GOTO1820
 1770   Z%=(ctrl%AND&30)/c4%
 1780   Z%=0 S$="M"
 1790   IF Z%<>pcol% PRINT"C"4-Z%:pcol%=
Z%
 1800   PRINTS$;xcoord%/SCL;","ycoord%/S
CL
 1810   UNTILp%>mp%:PRINT"H,C1":VDU6,3:E
NDPROC
 1820 S$="M":RETURN
 1830 S$="L0,D":RETURN
 1840 S$="0":RETURN
 1850 S$="L4,D":RETURN
```