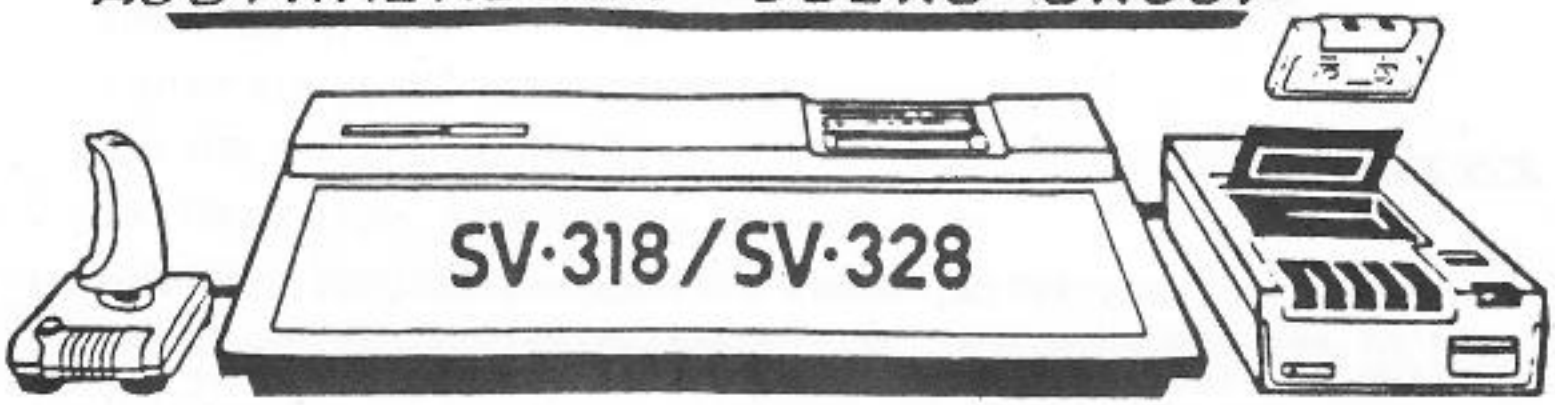
SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP



News Letter

REGISTERED BY AUSTRALIA POST PUBLICATION NO. TBH 0917 CATEGORY "B"

CONTENTS	PAGE

ISSUE No.
2 - 1
DATE
OCT - 1984

ALL CORRESPONDENCE TO:

S. A. U. G.,
P. O. BOX 191,
SOUTH LAUNCESTON,
TASMANIA, 7249.

(003) 312648

MEMBERSHIP FEES	
AUSTRALIA	$15.00
OVERSEAS	$20.00
OVERSEAS AIRMAIL ..	$25.50

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

INTRO

Happy Birthday to Us, Happy Birthday to Us , ... OH! hi there I did'nt see you come in. I was just singing, (what a noise). It's our 1st birthday , Yes it's hard to believe but The User Group Newsletter is ONE YEAR old, Personally I have aged FIVE!!. But it's all been worth it...

Many good things are in the pipeline for the comming year. So I will keep you all in suspense and not tell you about them. (It's fun being the editor!!)

This month's Newsletter is what I call a programming overdose with 4 major programs.

PUZZLE is a mathamatical game which will exercise your mind for a few hours.

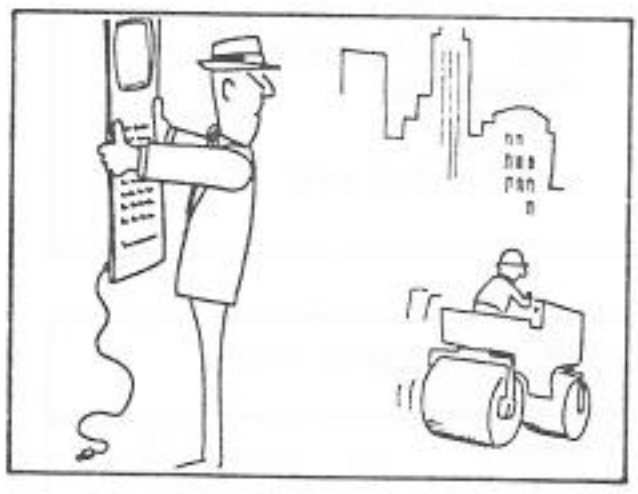
GRAPH is a program that allows you to graph data on the screen in 3 ways LINE,BAR & 3D BAR.

HAND is a drawing program, I have received a few drawing programs but this one stood out from the rest.

The last program this month is FREEFALL and comes from the book THE MISSILE & OTHER PROGRAMS. See the AD in this edition.

I hope you get something from this months issue and before I stop let me thank everyone again for their help this month.

..... Happy birthday to us ...



SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

Spectravideo Memory Map

By ???

The following addresses work O.K. on my SV-328. But I have not tried them on a 318.

These addresses are only the tip of the ice berg but at least it is a start.

* Indicates a 40 Column Screen Only.

	5D6	Error List
*	F543	Reduced line size to n columns
	F7F2/F7F3	Basic program end
	F7F4/F7F5	Basic program start
	F8F1	Trace command status (Ø=off , 175 = on)
	FAØ2	Keyclick (non-zero = on)
*	FAØ3	Cursor Y
*	FAØ4	X
*	FAØ5	on/off (non-zero = on)
*	FAØ6	Bottom up protect (Ø = unprotect whole screen or 256-n protects n bottom lines of screen)
*	FAØA	Write colour
*	FAØB	Screen colour
*	FAØC	Border colour
	FA1A/FA1B	Type ahead buffer pointer (start)
	FA1C/FA1D	(end)
	FA1E	Function key 1 (15 bytes + Ø)
	FA2E	Function key 2
	....	
	....	
	FABE	Function key 1Ø
	FE2E/FE2F	System time clock
	FE35	Reverse video
	FE38	Caps lock (writing does'nt affect light)
	FE3A	Screen mode
	FE3B	Sprite mode
	FE78	Causes input to be ignored for n/5Ø secs

PUZZLE

by : J. Van Staveren

```
 10  ' * * * * * * * * * * * * * * *
 20  '       MATH-PUZZLE
 30  ' * * * * * * * * * * * * * * *
 40  '
 50  '     21 APRIL 1979
 60  '
 70  ' WRITTEN BY;
 80  '           J VAN STAVERN
 90  '           30 WAVENEY ST
100  '          LAUNCESTON
110  '          TASMANIA   7250
120  '          AUSTRALIA
130  '
140  ' * * * * * * * * * * * * * * *
150  '
160 N=RND(-TIME)
170 SCREEN 0,0
180 CLS
190  PRINT TAB(7);"<<<<<   MATHPUZZLE   >>>>>"
200  PRINT :PRINT TAB(13);"A GAME OF SKILL"
210  PRINT:PRINT:PRINT:PRINT
220  PRINT"DO YOU NEED INSTRUCTIONS? YES/NO ";
230  A$=INPUT$(1)
240  IF A$<>"Y" THEN 390
250  CLS
260  PRINT:PRINT:PRINT
270  '       INSTURCTIONS
280  PRINT "'MATHPUZZLE' is a multiplication of 3"
290  PRINT "figures  times  2 figures,  where all"
300  PRINT "figures   have   been   replaced   by"
310  PRINT "Alphabetical characters."
320  PRINT
330  PRINT "If you  guess  the value  of a letter"
340  PRINT "correctly    all    the    characters"
350  PRINT "representing  that   value  will   be"
360  PRINT "replaced by that value."
370  PRINT:PRINT:PRINT
380  INPUT "TYPE ENTER TO COMMENCE PLAY ";Y$
390  '   DIMENSION MATRIX AND STRINGS
400      DIM A(5,5),B(5,5),L(10)
410  '
420  CLS
430  '   SET MAT A TO -1 AND MAT B TO 32 (BLANK)
440     FOR I%= 1 TO 5
450             FOR J%= 1 TO 5
460                     A(I%,J%)=-1: B(I%,J%)=32
470             NEXT
480     NEXT
490     '         ARRAY L=0
500     FOR I%= 1 TO 10
510             L(I%)=0
```

```
520     NEXT
530     '          SET VALUES
540     A(1,3)=INT(9*RND(1)+1)
550     A(1,4)=INT(10*RND(1))
560     A(1,5)=INT(10*RND(1))
570     A(2,4)=INT(9*RND(1)+1)
580     A(2,5)=INT(9*RND(1)+1)
590     '          SET ANSWERS
600     A1=(A(1,3)*100)+(A(1,4)*10)+A(1,5)
610     C=A(2,5)*A1
620     D=A(2,4)*A1
630     E=C+(D*10)
640     '          PLACE ANSWERS IN STRINGS
650     C$=STR$(C)
660     D$=STR$(D)
670     E$=STR$(E)
680     '          PLACE NUMBERS IN MAT A
690     C=5
700     FOR J%= LEN(C$) TO 2 STEP -1
710             A(3,C)=VAL(MID$(C$,J%,1))
720             C=C-1
730     NEXT
740     '
750     C=4
760     FOR J%= LEN(D$) TO 2 STEP -1
770             A(4,C)=VAL(MID$(D$,J%,1))
780             C=C-1
790     NEXT
800     '
810     C=5
820     FOR J%= LEN(E$) TO 2 STEP -1
830             A(5,C)=VAL(MID$(E$,J%,1))
840             C=C-1
850     NEXT
860     '          PLACE RANDOM LETTERS IN ARRAY L
870     FOR I%= 0 TO 9
880             V=INT(26*RND(1)+65)
890             FOR J%= 0 TO I%
900                     IF L(J%)=V THEN 880
910             NEXT J%
920             L(I%)=V
930     NEXT I%
940     '          PLACE LETTERS IN MAT B FOR PRINTING
950     FOR I%=1 TO 5
960             FOR J%= 1 TO 5
970                     IF A(I%,J%)<>-1 THEN B(I%,J%)=L(A(I%,J%))
980             NEXT J%
990     NEXT I%
1000    '          SCREEN PRINT-OUT
1010     C=1:GOTO 1070
1020     C=C+1
1030    CLS
1040    IF Q=1 THEN Q=0:PRINT"WRONG ";LEFT$(G$,1);" <>";G;:GOTO 1070
1050    '
1060    PRINT "CORRECT ";LEFT$(G$,1);" =";G;
1070    PRINT "GUESS NO; ";C
1080    PRINT : PRINT : F=0
1090    FOR I%= 1 TO 5
```

'First, I'll tell you the good part. See the little plug on the end of this cord? It's OK.'

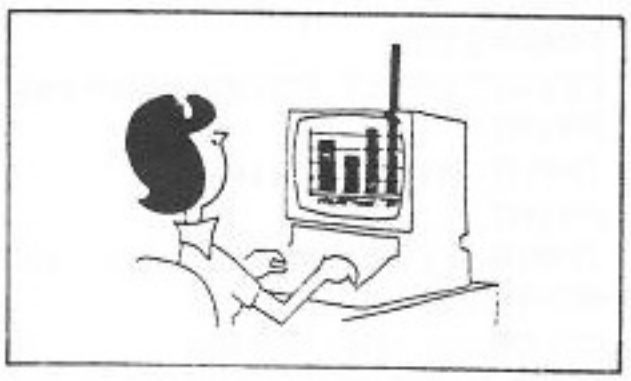
```
1100            FOR J%= 1 TO 5
1110                    PRINT TAB(J%*4+10);CHR$(B(I%,J%));
1120            NEXT J%
1130            PRINT
1140            IF I%=2 THEN PRINT TAB(18);"--------------"
1150            IF I%=4 THEN PRINT TAB(14);"------------------"
1160    NEXT I%
1170    PRINT : PRINT : PRINT
1180    IF W=1 THEN 1350          'W = WINflag
1190    INPUT "PICK A LETTER ";G$
1200    IF G$="" THEN 1030
1210    IF ASC(LEFT$(G$,1))<65 THEN 1030
1220    PRINT
1230    INPUT "WHAT'S IT'S VALUE ";G
1240    IF L(G)=ASC(LEFT$(G$,1)) THEN L(G)=G: GOTO 1260
1250    Q=1: GOTO 1020
1260    FOR I%= 1 TO 5
1270            FOR J%= 1 TO 5
1280               IF A(I%,J%)=G THEN B(I%,J%)=A(I%,J%)+48:A(I%,J%)=-1
1290               IF A(I%,J%)<>-1 THEN F=1
1300            NEXT J%
1310    NEXT I%
1320    IF F=1 THEN F=0: GOTO 1020
1330    '       F=0 IF YOU'VE WON
1340    W=1: GOTO 1030  '     print the board once more
1350    PRINT: W=0
1360    PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1370    PRINT "YOU SOLVED THE PUZZLE IN ";C;"GUESSES"
1380    PRINT
1390    PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1400    PRINT
1410    S=INT(G+(ASC(G$)/6))
1420    FOR I= 1 TO S:Q=RND(1):NEXT I
1430    INPUT "Play again?  Y/N  ";Y$
1440    IF Y$="" THEN 420
1450    IF LEFT$(Y$,1)="Y" THEN 420
1460    IF LEFT$(Y$,1)="y" THEN 420
1470    PRINT "END"
1480    END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

GRAPH

by : A. Kellner



```
 10 CLS:COLOR12,1,1
 20 K$="V9T255S1404CGF"
 30 SCREEN2,0
 40 LOCATE50,60
 50 PRINT" POINT"
 60 PRINT"  GRAPHER"
 70 C=RND(-TIME)
 80 C=INT(RND(1)*14+2)
 90 COLORC
100 D=D+1
110 IFD<4THENPLAY K$
120 IFD>10THEN130ELSE40
130 SCREEN0,0
140 CLEAR200:DEFINT A-Z
150 '
160 CLS:COLOR1,12
170 LOCATE0,7:PRINT"***** DO YOU WISH INSTRUCTIONS ? *****"
180 LOCATE4,9:PRINT"PRESS SPACEBAR FOR INSTRUCTIONS"
190 LOCATE2,11,0:PRINT"PRESS RETURN TO ENTER PROGRAM DIRECT"
200 V$=INKEY$
210 IFV$=CHR$(32)THEN1790
220 IFV$=CHR$(13)THEN240
230 GOTO200
240 CLS:COLOR12,1,1
250 LOCATE0,8
260 PRINT"  DO YOU WISH A DEMO SET OF POINTS ?"
270 PRINT:PRINT"                ANSWER Y OR N"
280 I$=INKEY$
290 IFI$="Y"ORI$="y"THEN1620
300 IFI$="N"ORI$="n"THEN320
310 GOTO280
320 CLEAR:SCREEN0,0:COLOR4:CLS:LOCATE 2,8:PRINT"DO YOU WISH TO LOAD A SET
    OF POINTS!"
330 PRINT:PRINT"                ANSWER Y OR N"
340 G$=INKEY$
350 IFG$="Y"ORG$="y"THENGOTO1510
360 IFG$="N"ORG$="n"THEN380
370 GOTO340
380 COLOR12:CLS:LOCATE 0,8:PRINT"HOW MANY POINTS TO PLOT:"
390 PRINT:PRINT" ENTER AT LEAST (1) FOR BAR GRAPH"
400 PRINT:PRINT" ENTER AT LEAST (2) FOR LINE GRAPH"
410 LOCATE10,15:INPUT B
420 IFB<1THEN380
430 DIM M(B,2)
440 LOCATE8,11
450 COLOR4:CLS:LOCATE 0,8:PRINT"    INPUT MAXIMUM X AND Y VALUES:"
460 PRINT
470 PRINT"          MAX.*** 1000 ***"
480 PRINT:PRINT"     ENTER VALUES LIKE THIS X,Y "
490 PRINT:LOCATE12,14
500 INPUT Q,U
```

```
510 IFQ>1000 ORU>1000 THEN CLS:GOTO450
520 PRINT
530 FORA=1TOB
540 PRINT"INPUT POINT NO.";A;" LIKE THIS X,Y "
550 PRINT
560 INPUT M(A,1),M(A,2)
570 PRINT
580 IFM(A,1)>Q ORM(A,2)>U THENGOTO1780
590 NEXTA
600 COLOR6:PRINT:PRINT"      ALL INFORMATION CORRECT ?":PRINT"          AN
    SWER Y OR N !"
610 W$=INKEY$
620 IFW$="Y"ORW$="y"THEN650
630 IFW$="N"ORW$="n"THEN520
640 GOTO610
650 COLOR12:CLS:LOCATE 0,8:PRINT"DO YOU WISH TO SAVE THIS SET OF POINTS"
660 PRINT:PRINT"            ANSWER Y OR N"
670 F$=INKEY$
680 IFF$="Y"ORF$="y"THEN1410
690 IFF$="N"ORF$="n"THEN710
700 GOTO670
710 COLOR4:CLS:LOCATE0,8:PRINT"WHAT KIND OF GRAPH"
720 LOCATE10,10:PRINT"1) LINE"
730 LOCATE10,12:PRINT"2) BAR"
740 LOCATE10,14:PRINT"3) 3D BAR"
750 PRINT:PRINT"      ANSWER 1,2 OR 3 & ENTER"
760 INPUT P
770 IFP=1ORP=2ORP=3THEN780ELSE710
780 FORW=1TOB
790 MN=M(W,2)+MN
800 NEXTW
810 MV1=INT(MN/B):MV=MV1*(150/U)
820 CLS:SCREEN1,0
830 LINE(40,0)-(40,170),4:LINE(30,160)-(256,160),4
840 L=1
850 FORI=60TO250STEP20
860 FORF=0TO4
870 PSET (I,158+F),4
880 NEXTF
890 COLOR6:LOCATEI-14,172:PRINTINT((Q/10)*L):L=L+1
900 NEXTI
910 '
920 L=1
930 FORI=40TO175STEP15
940 FORF=0TO4
950 PSET (38+F,185-I),4
960 NEXTF
970 LOCATE 10,180-I:PRINT INT((U/10)*L):L=L+1
980 NEXTI
990 FORA=1TOB
1000 XX=M(A,1):YY=M(A,2)
1010 X=(XX*(200/Q))+6:Y=YY*(150/U)
1020 PSET(X+34,160-Y),7
1030 NEXTA
1040 PLAY"O5BEFABFGGCC"
1050 IFP=1THENGOSUB1210ELSEGOSUB1330
1060 LINE(40,160-MV)-(256,160-MV),13
1070 COLOR13:LOCATE 120,0:PRINT"AVERAGE"
```

```
108Ø LINE(9Ø,4)-(115,4),13
109Ø COLOR1Ø:LOCATE8Ø,182:PRINT"ANOTHER GRAPH Y OR N ?"
11ØØ A$=INKEY$
111Ø IFA$="Y"ORA$="y"THEN32Ø
112Ø IFA$="N"ORA$="n"THEN COLOR12,1:END
113Ø GOTO11ØØ
114Ø '
115Ø '
116Ø '
117Ø ' ***** LINE   GRAPH *****
118Ø '
119Ø '
12ØØ '
121Ø FORA=1TO(B-1)
122Ø XX=M(A,1):YY=M(A,2):X2=M(A+1,1):Y2=M(A+1,2)
123Ø X=(XX*(2ØØ/Q))+6:Y=YY*(15Ø/U)
124Ø X1=(X2*(2ØØ/Q))+6:Y1=Y2*(15Ø/U)
125Ø LINE(X+34,16Ø-Y)-(X1+34,16Ø-Y1),6
126Ø NEXTA
127Ø RETURN
128Ø '
129Ø '
13ØØ ' ***** BLOCK  GRAPH *****
131Ø '
132Ø '
133Ø FORA=1TOB
134Ø XX=M(A,1):YY=M(A,2)
135Ø X=(XX*(2ØØ/Q))+6:Y=YY*(15Ø/U)
136Ø IFY<MV THENBC=12 ELSE BC=7
137Ø LINE(X+31,16Ø)-(X+37,16Ø-Y),BC,BF
138Ø IFP=3THENGOSUB19ØØ
139Ø NEXTA
14ØØ RETURN
141Ø CLS:LOCATEØ,8:PRINT"    INPUT YOUR FILENAME TO SAVE THESE  RESULTS ON
     DISK:"
142Ø LOCATE5,14:INPUT Z$
143Ø OPEN "1:"+Z$ FOR OUTPUT AS #1
144Ø PRINT #1,B,Q,U
145Ø FORW=1TOB
146Ø PRINT #1,M(W,1),M(W,2)
147Ø NEXTW
148Ø PRINT #1,"END"
149Ø CLOSE 1
15ØØ GOTO71Ø
151Ø ON ERROR GOTO168Ø
152Ø CLS:LOCATEØ,8:PRINT"INPUT FILENAME OF YOUR SET OF DATA TO  BE LOADED"
153Ø LOCATE5,14:INPUT Z$
154Ø OPEN "1:"+Z$ FOR INPUT AS #1
155Ø INPUT #1,B,Q,U:DIM M(B,2)
156Ø FORW=1TOB
157Ø INPUT #1,M(W,1),M(W,2)
158Ø IF EOF(1) THEN71Ø
159Ø NEXTW
16ØØ CLOSE 1
161Ø GOTO 71Ø
162Ø Q=1Ø:U=1ØØ:B=1Ø
163Ø FORW=1TOB
164Ø READ M(W,1),M(W,2)
```

```
1650 NEXTW
1660 DATA 1,100,2,20,3,55,4,80,5,60,6,20,7,95,8,40,9,65,10,35
1670 GOTO710
1680 IFERR=53ORERL=1540THEN1690ELSE ON ERROR GOTO 0:END
1690 CLS:LOCATE10,8:PRINT"FILE NOT ON DISK !"
1700 PRINT:PRINT  "CHECK IF YOU HAVE MADE A TYPING ERROR  OR HAVE THE WRONG
      FILENAME !! "
1710 PRINT" OR CHECK YOUR FILENAME AGAINST THOSE  ON THIS DISK ;-THE ONES W
     ITHOUT THE    FULL STOP AFTER THE FILENAME ARE DATA  -FILES !!! ":PRIN
     T:PRINT"        ** FILES **":PRINT
1720 FILES
1730 LOCATE,,0:PRINT:PRINT"* HIT SPACE BAR WHEN READY TO RESUME *"
1740 QQ$=INKEY$
1750 IFQQ$=CHR$(32)THEN1770
1760 GOTO1740
1770 CLOSE 1 :RESUME 1510
1780 CLS:LOCATE 0,8:PRINT"NO GREATER THAN MAX. VALUE PLEASE !!":FORW=1TO300
     0:NEXTW:CLS:LOCATE0,8:PRINT"  X = "Q,"  Y= "U:PRINT:PRINT:GOTO540
1790 CLS:COLOR4,1
1800 PRINT"          *** INSTRUCTIONS ***"
1810 PRINT:PRINT"  THIS PROGRAM WILL DRAW EITHER A LINE OR BAR TYPE GRAPH O
     N THE SCREEN AND    INDICATE THE AVERAGE!"
1820 PRINT"  FIRSTLY THE NUMBER OF PONITS YOU WISHTO PLOT HAS TO BE ENTERED
     ,THEN THE AXISMAXIMUMS,THEN YOU INPUT THE POINT      CO-ORDINATES IN A
      (X) AXIS - (Y) AXIS  FORMAT."
1830 PRINT"  ONCE ALL THE INFORMATION HAS BEEN    ENTERED YOU HAVE THE OPTI
     ON OF SAVING  YOUR DATA ON DISK FOR FUTURE REFERENCE."
1840 PRINT"  THIS DATA SAVED CAN BE REPLOTTED BY  LOADING YOUR INFORMATION
     FROM DISK AT  THE BEGINNING WHEN PROMPTED."
1850 PRINT"  THE USUAL FILNAME FORMAT CAN BE USED E.G (POWER.USE). ENTER ON
     LY THE NAME,   QUOTES OR (1:) IS NOT REQUIRED !"
1860 PRINT:PRINT"    ** PRESS SPACEBAR TO RESUME **"
1870 V$=INKEY$
1880 IFV$=CHR$(32)THEN 240
1890 GOTO1870
1900 LINE(X+37,160-Y)-(X+41,154-Y),BC
1910 LINE(X+41,154-Y)-(X+35,154-Y),BC
1920 LINE(X+35,154-Y)-(X+31,160-Y),BC
1930 LINE(X+41,154-Y)-(X+41,154),BC
1940 LINE(X+41,154)-(X+37,160),BC
1950 RETURN
```

"They installed a computer and the instructions were printed on the back."

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

EXPLORING BASIC Pt-7

by L.A. Dunning.

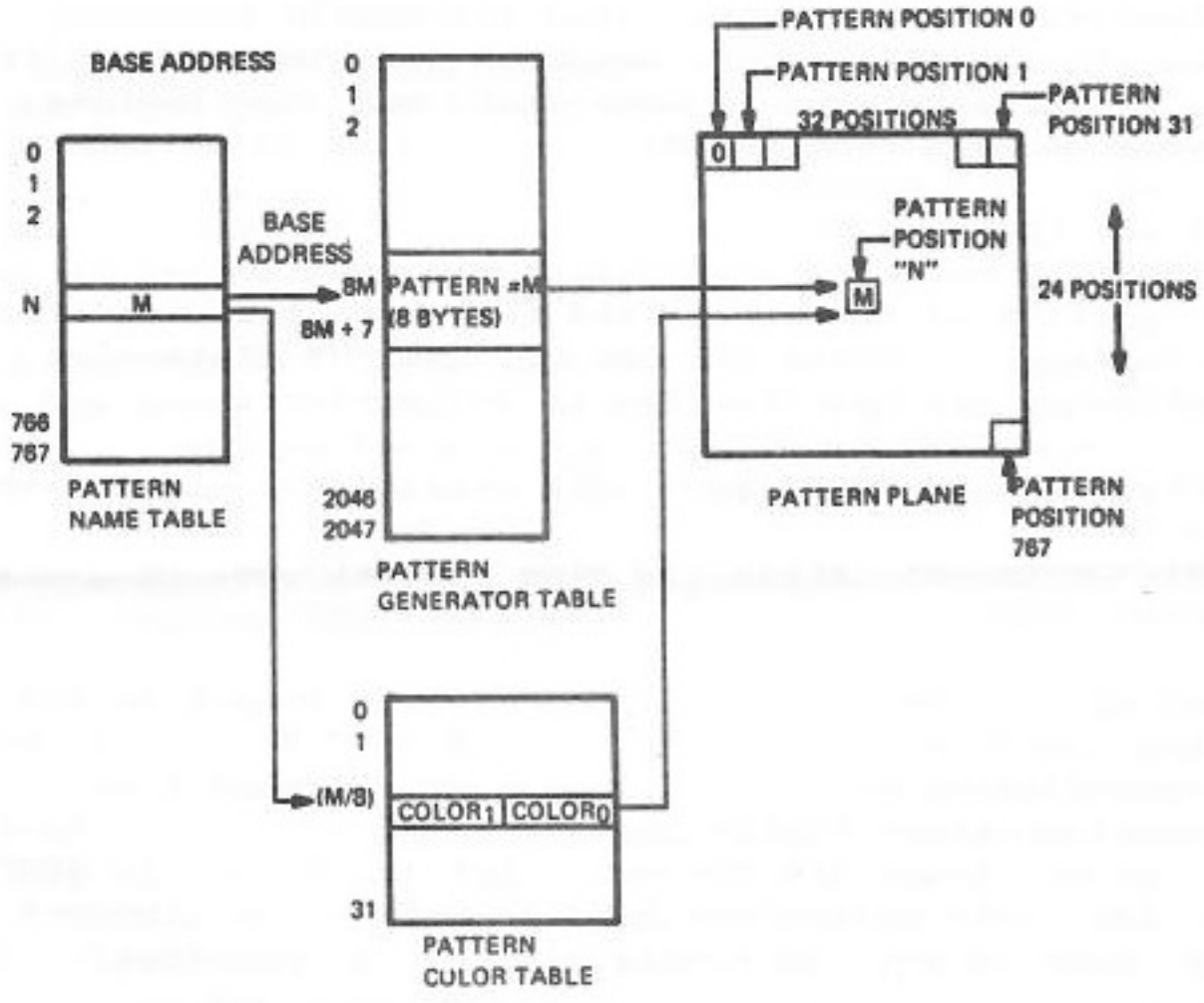
This month I describe Graphics Mode 1 and talk a bit about Number Bases and Logical Functions.

GRAPHICS MODE 1

This is the unimplemented mode on the Spectravideo. Why it wasn't added is a mystery, perhaps space limitations meant that it could not be included. This is a pity since it combines features of both Text and Graphics 2 modes and would be an excellent games mode.

The display is in Graphics 1 mode when bit 1 of register Ø and bits 3 and 4 of register 1 are set to Ø. While in this mode the screen is divided into a 32 X 24 pattern grid and all 32 SPRITES are available.

Imagine the Text Mode which was described back in part 4 of this series, however instead of having a single foreground/ background combination available at any one time imagine up to 32 possible combinations available at the same time. The mode does this by using the Pattern Color Table, but this is a reduced version from that available for GRAPHICS 2. Diagram 1 illustrates the connection.



Each entry uses the high and low 4 bits to define foreground/background colors but there are only 32 entries !. In fact each entry defines the foreground/background colors for 8 pattern definitions, so that entry 1 for example, defines colors for pattern definitions 8 to 15. Diagram 2 illustrates this relationship.

Byte No.	Pattern No.	Byte No.	Pattern No.
0	0..7	16	128..135
1	8..15	17	136..143
2	16..23	18	144..151
3	24..31	19	152..159
4	32..39	20	160..167
5	40..47	21	168..175
6	48..55	22	176..183
7	56..63	23	184..191
8	64..71	24	192..199
9	72..79	25	280..207
10	80..87	26	208..215
11	88..95	27	216..223
12	96..103	28	224..231
13	104..111	29	232..239
14	112..119	30	240..247
15	120..127	31	248-255

There are still only 256 available patterns however each is now an 8 X 8 rather than a 6 X 6 pattern as in TEXT and the color used for each is intrinsic to that pattern and not its' placement on the screen. This could be quite useful for games and the like where you might wish to move a background and a background color at the same time.

Can it be used from BASIC ?. The problem is that there is no support given to this mode other than the SPRITE commands. Listing 1 is a crude attempt to fool the computer that the display is actually SCREEN Ø. I leave you to make your own conclusions about the effectiveness of such from BASIC.

'GET' and 'PUT'

As you may be aware BASIC has two commands that can be used to obtain a portion of the screen and then dump that information back onto the screen. These are the GET and PUT statements and their implementation on the Spectravideo is not perfect.

Firstly, what does GET do? GET obtains the video information of an imaginary box on the screen as defined by two opposite corners and dumps this into an array. The format is :- GET (X1,Y1)-(X2,Y2),ARRAY NAME.

X1 and X2 are the horizontal coordinates from Ø to 255. Y1 and Y2 are the vertical coordinates from Ø to 191. X1 and Y1 are absolute coordinates and do not change with screen type. X2 and Y2 however depend on which SCREEN you are using. If you ae in SCREEN 1 then the above ranges are correct, but if you are in SCREEN 2 you should divide both parameters by 4 to obtain the correct position. The array name is any legitimate name of a previously DIMensioned array.

When GET is called BASIC does the following :-

a) Determines the height and width of the rectangle and the upper left hand corner of the area.

b) Assigns the values of width times 4 to element Ø and of the height to element 1 of the array.

c) Starting with element 2 of the array, it works left to right, top to bottom through the defined area and dumps a color value to

that array for each pixel in that area.

Thus the dimensions and contents of an area can be defined in an array.

There are some quirks however. It needs only 4 bits to define a color so two pixels can be defined in one byte. If the width is an odd number of pixels a dummy column of pixels is added the respective bits are zeroed. The number of bytes needed to hold an area is equal to (H X W)/2

Where H is the height of the rectangle, W is the width of the rectangle ORed with 2. The number of elements you must DIM in an array for GET will depend on the type of array DIMensioned. Integer arrays use 2 bytes per element, single precision arrays use 4 bytes and double precision arrays use 8 bytes per element. You should add 1 to the final number to account for elements Ø and 1 in the array, which are reserved for width and height values. Listing 2 demonstrates how such an area is stored.

PUT (X,Y), Array name, Operation

Where X and Y are the horizontal and vertical coordinates of the top left hand corner of the dump, the Array name is that of the array being dumped and Operation is the Logical Function performed by the dump.

As the array is dumped on a pixel by pixel basis, this can cause problems in SCREEN 1. Since only two colors can be defined in each line of a character position, arrays dumped out of sync' with those positions and the position it was gotten from will cause unexpected results. Normally the last color dumped will change an earlier color as BASIC tries to outguess the user. In SCREEN 2 this is no problem because the color of each pixel is independent of any other pixel. It is almost as if the PUT statement were written for another machine.

What effect do the operations have ? PSET dumps the color as recorded in the array. PRESET inverts all bits in the array before dumping them. AND performs a Logical AND on the incoming color and the color already at that pixel location. OR does a Logical OR in the same manner and XOR does a Logical XOR between colors. Comparisons are done on a Pixel or 4 bit basis. eg BLACK (ØØØ1b) XOR WHITE (1111b) produces GREY (1110b).

Normally in SCREEN 1 it is just not worth doing any operation otherthan PSET because of the unpredicatble results. One way around this, if you are only working with two colors, is to declare the background transparent (ØØØØb) and put the background color in the border. You can then change the background color without severe changes in the PUT function.

Knowing how a screen area is dumped to an array means that you could design your own array which had never been initialized by GET and then PUT it on the screen, or manipulate both height and width to change the image....save it to disk as an alternative way of storing data (screen files are difficult to manipulate) and other possibilities. Listing 3 demonstrates some manipulation of an array.

NUMBER BASES

Have you ever wondered about the other number bases used on the Spectravideo? We all know how to use Decimal, but what about binary, octal, or hexadecimal? If we restrict ourselves to the INTEGER range

(-32768 to +32767) then all four bases can be used. The important thing to realize here is that each is represented in the same manner, by two bytes in succession in memory. This gives 16 bits to represent the number. Fifteen bits are used to store the quantity and the most significant bit is used to store the sign of the number. (Ø = positive, 1 = negative). The number 1 is really ØØØØØØØØØØØØØØ1, Ø is ØØØØØØØØØØØØØØØ, and -1 is 1111111111111111.

The &B, &O, and &H prefixes are therefore mostly cosmetic in nature; while what is listed appears differently, it is still stored the same in memory. The complementing statements of STR$, BIN$, OCT$, and HEX$ work in the reverse manner, producing a cosmetic display for the users' convenience. The last three do have an annoying fault. STR$ places a space character at the start of its' string for use with the negative sign: eg " 1134" and "-1134". The other statements do not do this since in that representation there IS no negative sign. Also the output is of variable length and when put into a PRINTUSING statement justifies to the left. To avoid this the following function will produce a numerically correct constant string:-

```
DEF FN##$(N,L) = STRING$(L-LEN(####$(N)),"Ø")+####$(N)
```

Where # is the function name, N is the number to be converted, L is the standard length of the string produced, and ### is either BIN, OCT, or HEX, depending on the type of conversion. L might be replaced by a constant if you want a standard length all the time. In practice, binary and hexadecimal are used most often and Octal is a hangover from the early days of computing when this was the output given by those early machines. Pity the poor programmers. Binary is most useful when examining individual bits or logicalfunctions; hexadecimal is more useful when dealing with memory blocks (Spectravideo memory goes from ØØØØH to FFFFH) and machine code.

NOT, AND, OR, XOR, IMP, and EQV all work on a bit by bit basis and so may produce unexpected results. In BASIC all results are in the INTEGER range and include the sign-bit. This is why NOT(Ø) is -1 LISTING 4 demonstrates the effect of logical operations. When dealing with different bases while in BASIC it should be remembered that their real function is as a convenience to the user.

Next month I will talk about INPUT/OUTPUT routines, MENUS, and Editing techniques.

Listing 1

by : L.A. Dunning

```
10 REM Listing 1 - Part 7
20 REM Almost sets up GRAPHIC I mode
30 CLEAR1000:DEFINTA-Z:FG=15:BG=4
40 COLORFG,BG:SCREEN0:OUT129,88:OUT129,131:FORA=0TO31:B=(A\2):C=15-B:IFB=
   2THENB=FGELSEIFB=13THENC=BG
50 Z=C+B*16:VPOKE5632+A,Z:NEXT
60 J=PEEK(&HFA07):J=JAND239:POKE&HFA07,J:OUT129,J:OUT129,129
70 FORA=0TO255:VPOKEA,A:NEXT
80 LOCATE0,9:PRINT" Very difficult to use  in basic":KEY1,"screen0"+CHR$(
   13)
```

Listing 2

by : L.A. Dunning

```
10 REM Listing 2 - Part 7
20 REM Demonstrates how GET is stored
30 CLEAR1000:DEFINTA:INPUT" SCREEN (1-2) ";TZ:IFTZ<1ORTZ>2GOTO30ELSESC=TZ
   ^2:COLOR15,4,5:SCREENTZ:DIMA(65)
40 DEF FNH$(N)=STRING$(4-LEN(HEX$(N)),"0")+HEX$(N)
50 LINE(200,0)-(214,14),1,B:LINE(201,1)-(214,13),15:PSET(205,10),14:GET(2
   00,0)-(214\SC,14\SC),A:A$=INPUT$(1)
60 WD=(A(0)\(4*SC))OR2:HT=A(1)
70 N=2:XX=1:YY=1:PRINT" ";
80 A$=FNH$(A(N)):FORL=4TO2STEP-2:X$=MID$(A$,L-1,1)+MID$(A$,L,1):PRINTX$;:
   XX=XX+2:IFXX>WDTHENPRINT:PRINT" ";:XX=1:YY=YY+1
90 NEXT:N=N+1:IFYY<=HTGOTO80
100 A$=INPUT$(1):SCREEN0:GOTO30
```

Listing 3

by : L.A. Dunning

```
10 REM Listing 3 - Part 7
20 REM Demonstrates how GET is stored
30 CLEAR1000:DEFINTA:DIMA(65)
40 INPUT" SCREEN (1-2) ";TZ:IFTZ<1ORTZ>2GOTO30ELSESC=TZ^2:COLOR15,4,5:SCR
   EENTZ:X=200:Y=8
50 LINE(X,Y)-(X+7,Y+7),1,BF:LINE(X+8,Y)-(X+15,Y+8),14,BF:LINE(X,Y+8)-(X+8
   ,Y+15),15,BF:LINE(X+8,Y+8)-(X+15,Y+15),9,BF:GET(X,Y)-((X+15)/SC,(Y+15)
   /SC),A:A$=INPUT$(1)
60 WD=A(0):HT=A(1)
70 A(0)=WD*2:A(1)=HT/2:PUT(16,16),A,PSET
80 A(0)=WD/2:A(1)=HT*2:PUT(16,90),A,PSET
90 A(0)=WD:A(1)=HT:IFTZ=2THENSWAPA(2),A(4):SWAPA(3),A(5):GOTO110
100 FORB=2TO33STEP2:SWAPA(B),A(B+32):SWAPA(B+1),A(B+33):NEXT
110 PUT(96,16),A,PSET
120 A$=INPUT$(1):SCREEN0:GOTO40
```

Listing 4

by : L.A. Dunning

```
 1Ø REM Listing 4 - Part 7
 2Ø REM demonstrates integers
 3Ø CLEAR2ØØØ:DEFINTA-Z:STOPON:ONSTOPGOSUB17Ø:DEF FNH$(N)=STRING$(4-LEN(HE
    X$(N)),"Ø")+HEX$(N):DEF FNB$(N)=STRING$(16-LEN(BIN$(N)),"Ø")+BIN$(N):D
    EF FNO$(N)=STRING$(6-LEN(OCT$(N)),"Ø")+OCT$(N):C$=CHR$(27)+"K":CLS
 4Ø LOCATE,2:PRINTC$;:INPUT"INPUT a 1st number->";E$:N1=VAL(E$)
 5Ø LOCATE,6:PRINTC$;:INPUT"INPUT a 2nd number->";N$:N2=VAL(N$)
 6Ø LOCATE,1Ø:PRINT"INPUT an action between numbers":PRINTC$;:INPUT"- and,
    or,xor,imp,equ =>";A$
 7Ø O=INSTR("##andor xorimpequ",A$)\3:IFO=ØGOTO6ØELSEONOGOSUB11Ø,12Ø,13Ø,1
    4Ø,15Ø
 8Ø PRINT:PRINT"Number Dec. Hex  Binary            Octal "
 9Ø PRINT"1st  ";:N=N1:GOSUB16Ø:PRINT"2nd  ";:N=N2:GOSUB16Ø:PRINTUSING"1\
    \2";A$;:N=N3:GOSUB16Ø
1ØØ GOTO4Ø
11Ø N3=N1ANDN2:RETURN
12Ø N3=N1ORN2:RETURN
13Ø N3=N1XORN2:RETURN
14Ø N3=N1IMPN2:RETURN
15Ø N3=N1EQUN2:RETURN
16Ø PRINTUSING"####### \   \ \                \ \     \";N,FNH$(N),FNB$(N),FNO$
    (N):RETURN
17Ø CLS
```

HAND

by : D. Napper

```
 10 COLOR10, 1,1:C2=2:SCREEN1,2: LOCATE80,20:PRINT"JOYSTICK DRAWER":PRINT:
    PRINT"  Use the joystick in port 1 to move the     hand around the pag
    e other functions       and their keys below."
 20 PRINT
 30 PRINT"  trigger to put point"
 40 PRINT"  enter   to mark a point"
 50 PRINT"  1,b,c   draws lines,boxes and circles for          x position"
 51 PRINT"  f       fill areas with foreground color"
 60 PRINT"  p       changes pen color"
 70 PRINT"  []\     changes pen speed"
 80 PRINT"  1&2     changes foreground color"
 90 PRINT"  ,.      changes ratio of circles"
100 PRINT"  x       clears screen foreground color"
110 PRINT"  s       saves screen"
120 PRINT"  a       loads screen"
121 LOCATE40,180:PRINT"PRESS ANY KEY TO CONTINUE"
130 A$=INKEY$:IFA$=""THEN130
140 SCREEN1,2:CLICKOFF:R=1.3
150 FORT=1 TO16:READA$,C$:B$=B$+CHR$(VAL(A$)):D$=D$+CHR$(VAL(C$))
160 NEXT:
170 SPRITE$(0)=B$+D$
180 DATA 128,0,96,248,119,220,55,222,11,159,4,7,2,31,5,31,31,191,15,95,7,1
    75,6,215,1,231,1,254,0,124,1,254
190 CLS
200  COLOR ,C: LINE(10,10)-(245,182),15,B:PAINT(250,5),15:LOCATE70,2: COLO
    R 1:PRINT"JOYSTICK DRAWER"
210 '
220 Z=1:X=100:Y=100
230 D=STICK(0)+STICK(1)
240 V=STRIG(0)+STRIG(1)
250 LINE(40,0)-(60,10),C,BF
260 PUT SPRITE0,(X,Y),C2
270 A$=INKEY$
280 IF D=1THENY=Y-Z
290 IF D=2THENY=Y-Z:X=X+Z
300 IF D=3THENX=X+Z
310 IF D=4THENX=X+Z:Y=Y+Z
320 IF D=5THENY=Y+Z
330 IF D=6THENY=Y+Z:X=X-Z
340 IF D=7THENX=X-Z
350 IF D=8THENX=X-Z:Y=Y-Z
360 IF Y<10THENY=10
370 IF Y>180THENY=180
380 IF X<10THENX=10
390 IF X>245THENX=245
400 IFV=-1THENPSET(X,Y),C
410 IF A$="1"THENC=C+1
420 IF A$="2"THENC=C-1
430 IF C>15THENC=1
440 IF C<0THEN C=15
450 IF A$="f"THENPAINT(X,Y),C
```

```
460 IF A$="x"THENCLS: GOTO 190
470 IF A$=CHR$(13)THENX1=X:Y1=Y
480 IF A$="1"THENLINE(X1,Y1)-(X,Y),C
490 IF A$="["THENZ=1
500 IF A$="]"THENZ=2
510 IF A$="\"THENZ=10
520 IF A$="b"THENLINE(X1,Y1)-(X,Y),C,B
530 IF A$="s"GOTO680
540 IF A$="."THENR =R+.1
550 IF A$=","THENR =R-.1
560 IF A$="p"THENC2=C2+1
570 IF C2>15 THEN C2=1
580 IF R<.1THENR=.1
590 IF R>5THENR=5
600 IF A$="a"THENLOCATE40,184: COLOR 1:PRINT" PRESS PLAY ON TAPE": GOTO 70
    0
610 D=R*(X-X1)
620 IF X<X1THEND=R*(X1-X)
630 IFR<1THEND=(X-X1)
640 IF C=1OR C=0THENC1=15ELSEC1=1
650 IF A$="c"THENCIRCLE(X1,Y1),D,C,,,R
660 CIRCLE(50,5),5,C1,,,R
670  GOTO 230
680 LINE( 0, 0)-(245,10),15,BF:FOR T=1 TO4000:NEXT:CSAVE"plans",S
690  GOTO 200
700 CLOAD
710 GOTO200
```

S.V. LOGO

The following will print the Spectravideo LOGO on your SCREEN.

```
10 DEF USR0=&H3420 : A=USR0(0)
20 DEF USR0=&H4782 : A=USR0(0)
30 DEF USR0=&H3541 : A=USR0(0)
```

NOTE : If you run this program and your computer locks up just delete line 10 and run agin.

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

FREEFALL

is a short program from

Bernard Scott's

THE MISSILE & OTHER PROGRAMS

for

SPECTRAVIDEO 318/328

This new book of 20 programs contains games, graphics displays, music and sounds library, novelties and educational programs.
Copious notes are included with most programs to help you understand their construction and to allow you to extend them and to write your own programs.
Several of the programs occur in different versions to let you discover the different capabilities of the SPECTRAVIDEO.
The book is printed on A4 sheets, stapled, but with pre-punched holes for easy insertion into a cover, and for ease of typing.
The paper used in the book is heavier than that used in the sample program.

To minimise cost, this book is available only by mail.
For your copy, write to

ABSPECT SOFTWARE,
R.M.B. 288
CRESWICK
VICTORIA 3363

enclosing $9.95 to cover cost of book, packing and postage.

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

LIBRARY NOTES

by J. Collins.

Some very interesting programs have been submitted recently for possible inclusion in the 'For Sale' list and two that have been added to the list are reviewed below. Please continue to send us your handiwork.....you could be sitting on a goldmine with that masterpiece of yours.

Let's look now at a version of RUBIKS' CUBE submitted by member R. G. Crawford of South Australia :- First thing I must state here is that I personally hate the Cube in any shape or form.....I can't get even one side to look like it should and thus have decided that it is evil and a tool of the devil. My son does not agree, but then he can get it all out and also beats me at most video games.

Back to the program.....Comprehensive instructions are provided inside the program and these allow easy manipulation of the cube faces on screen.....all the manual twists and turns are possible and although slow because of BASIC are not so slow as to annoy. Color has been used well and the screen presentation is in 3D perspective with a front, side, and top face all visible at once. Really not a lot more to say here.....Cube fanatics will like it, others may try it finding it easier than the real thing. A reset key has been provided in the program to take cube back to starting condition and you don't get that on the plastic version. Suitable for 318/328 cassette or disk.

At $5.ØØ it's good value so come on cubists.

Second is a FILING CABINET utility program from member T. McGee of N.S.W.. This is 'Menu' driven and has quite a versatile range of operations allowing entry, storeage, and manipulation of data. Program will be supplied with printed list of instructions. It isn't dBASE 11 and of course does not have all of the deluxe features of full database programs but it doesn't cost $7ØØ.ØØ either and can be used with your standard system. If you want a file manager give it a try.....at $5.ØØ you wont break the bank. Suitable only for cassette but easily adapted by disk users for their own systems. Is 22k Bytes long with 1ØØØØ bytes allocated for string space. Not suitable for unexpanded 318 machines.

September was a very busy month for the library with twenty requests for software, newsletter programs, and twelve other requests for general assistance. That may not seem like a lot but when combined with all other pursuits, (like living, working etc) it eats into time. I think I know now why the editor had a smile on his face when I agreed to help out with the library. Come to think back on it now it actually looked more like a great big grin ! Enough of this, till next month.

FREEFALL

A skydiver knows the joy of floating free. He also realises the danger. Here is a game which allows you to become a skydiver while sitting in front of your SV.

```
100 COLOR 8,15,4: SCREEN 2
110 LOCATE 35,80: PRINT"FREEFALL"
120 FOR T = 1 TO 500: NEXT
130 COLOR,15,4: SCREEN 1
140 COLOR 10: LOCATE 40,50: PRINT"YOU MUST FREEFALL 1000 METRES"
150 COLOR 12: LOCATE 25,70: PRINT"OPEN PARACHUTE WHEN LIGHT IS GREEN"
160 COLOR 1: LOCATE 10,90: PRINT"USE THE SPACEBAR TO OPEN YOUR
    PARACHUTE"
170 COLOR 8: LOCATE 80,110: PRINT"THERE IS DANGER"
180 LOCATE 50,120: PRINT"HITTING THE GROUND TOO FAST"
190 LOCATE 85,130: PRINT"WILL KILL YOU"
200 COLOR 7: LOCATE 60,180: PRINT"press any key to go on"
210 IF INKEY$ = "" THEN 210
220 FA = 1000
230 COLOR,1,4: SCREEN 1
240 A$ = "": B$ = "": C$ = ""
250 RESTORE 560: FOR T = 1 TO 8: READ A: A$ = A$+CHR$(A): NEXT:
    SPRITE$(0) = A$
260 RESTORE 570: FOR T = 1 TO 8: READ B: B$ = B$+CHR$(B): NEXT:
    SPRITE$(1) = B$
270 RESTORE 580: FOR T = 1 TO 8: READ C: C$ = C$+CHR$(c): NEXT:
    SPRITE$(2) = C$
280 LINE(50,175)-(205,180),10,BF
290 X = 128: Y = 5: M = 131: E = 9
300 K = 0: I = .5: J = .1
310 LINE(0,0)-(15,15),8,BF
320 Y = Y+I: E = E+I: I = I+J
330 PUT SPRITE 0,(X,Y),K: PUT SPRITE 1,(X+7,Y),K
340 PUT SPRITE 2,(M,E),3
350 IF E = 90 THEN GOSUB 540
360 IF POINT(5,5) = 8 OR J = -.2 OR J = 0 THEN 380
370 IF STRIG(0) THEN K = 13: J = -.2: HI = E-9
380 IF I < .5 THEN J = 0
390 IF POINT(M,E+7) = 1 THEN 320
400 IF I > 2 THEN 440
```

```
410 IF HI*10 > 1000 THEN 470
420 COLOR 13,1: SCREEN 0,0: LOCATE 7,10: PRINT"YOUR FREEFALL WAS"
    HI*10"METRES"
430 FOR T = 1 TO 1000: NEXT: GOTO 230
440 COLOR 1,8,8: SCREEN 2
450 LOCATE 70,90: PRINT"SPLAT"
460 FOR T = 1 TO 1000: NEXT: GOTO 230
470 COLOR 13,1: SCREEN 1: LOCATE 45,100: PRINT"YOUR FREEFALL WAS"
    HI*10"METRES"
480 IF HI*10 > FA THEN FA = HI*10: COLOR 7: LOCATE 70,120: PRINT"
    YOUR JUMP IS BEST": GOTO 500
490 COLOR 8: LOCATE 50,120: PRINT"THE BEST JUMP IS"FA"METRES"
500 COLOR 2: LOCATE 35,160: PRINT"DO YOU WANT TO JUMP AGAIN (Y/N)?"
510 A$ = INKEY$: IF A$ <> "Y" AND A$ <> "N" THEN 510
520 IF A$ = "Y" THEN 230
530 COLOR 15,4,5: SCREEN 0,0: LOCATE 15,10: PRINT"GOODBYE": END
540 LINE(0,0)-(15,15),12,BF
550 RETURN
560 DATA 7,56,64,64,32,16,8,4
570 DATA 224,28,2,2,4,8,16,32
580 DATA 28,28,8,73,62,28,20,20
```

NOTES

100/120	title
130/210	information and instructions
220	initial best fall
240	cleans out the strings each time round
250/270	read information for 3 sprites
280	the ground
290	coordinates for patachute and person
300	initial color of parachute (invisible): initial speed of freefall: acceleration
310	red light in top left corner of screen
320	for movement of person, etc
330/340	sprites for parachute, sprite for person
350	after some freefall gosub to change color of light
360	stops cheating with spacebar
370	spacebar pressed parachute opens: person slows down: measures distance of freefall
380	if parachute terminal velocity has been reached no more slowing
390	if you are still in the air go around again
400	if you hit the ground too fast