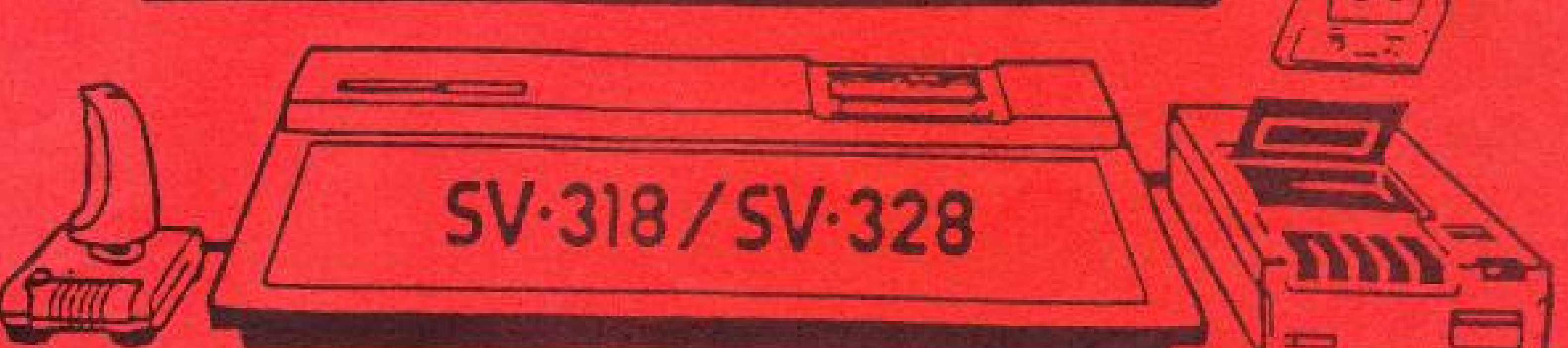
SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP



News Letter

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

Merry Christmas

ISSUE No.
2 - 3

DATE
DEC - 1984

ALL CORRESPONDENCE TO:

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

(003) 312648

ANNUAL SUBSCRIPTION

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

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

INTRODUCTION

JINGLE BELLS.... JINGLE BELLS....
JINGLE ALL THE WAY ....

Hello again, it's the festive season and I'm preparing for a busy few weeks (cleaning out the chimney for Santa e.t.c.).

So to get the Newsletter under way lets look at what we have this month.

Firstly Mr Dunning again amazes me with his Knowledge of the S.V. computer, this month may be the best yet. Keep up the good work.

We have a Menu Program which actually picks the Disk Directory entries by itself (neat trick). It has a few other tricks I believe you will like. Thanks Steve for that. It is nice to see some technical programming emerging.

Moving right along we have a review of a Coleco Game (Shame we Don't get S.V. Games to Review).

Our game program of the month is something special for Xmas. It's called Conservator and Multinash and you are sure to destroy two joysticks before you master it.

Well thats about it, all large articles this month, but good ones.

I hope you all are getting along O.K. with the C.R.C. checking program. I only had I call about it. For new members this program works with the funny letters at the beginning of each line of listings and checks that you have entered the line correctly. You can find the program in the November Issue or send $4 to the group care of the Library and we will send the program to you on Cassette.

Contributions to the Group Newsletter.

All listings <u>MUST</u> be on Cassette or Disk (same returned A.S.A.P.) with an explaination, about half a page.

Major articles are always welcome but would be appreciated if they are in one of three forms.

1) Wordstar File
2) Just Write Jr.
3) S.A.U.G. Word Processor. $3+CASS from Library.

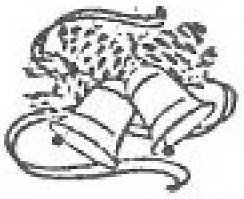
However if you must send it on paper please print.

Also please put a REM on the front of program listings showing Author e.t.c. Details.

Well I must go now and finish putting up the Xmas Tree. All the best and I hope to see you all well in the New Year.

P.S.   I hope SANTA brings you a 40 MEG Hard Disk Drive for Xmas.

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP



EXPLORING BASIC Pt-8

By L.A. Dunning.

This month I talk about Data, the Keyboard and a lost command.

DATA STATEMENTS

Basic uses three statements to support an inherent data base. These are; DATA which enables data in the form of text; READ which asignes those data statements to variables and RESTORE which sets the start of the data statements to be read. The RESTORE statement is of concern here. Normally restore has two formats: RESTORE & RESTORE ###### where ###### is a line number; the first resets the data pointer to the start of the program (RUN also does this) and the second sets the pointer to the start of the line listed. This "pointer" is located at memory location F7F4H. If no data is read or the pointer reset, it points to a location derived from another memory location - F54AH which I believe to be the first location of usuable RAM. As data is read, it points to the first byte after the previous data statement, which will either be a comma or a zero indicating end-of-line. When data is read, basic looks for the next available DATA statement or comma, or data text and reads/adjusts the pointer accordingly.

"So what"? you might say. Firstly, if you are using a cassette system you can't use a disk drive to read in constants for the program (and thus save program memory) and so must either read in data via the tape (which could take some time if much data is read) or use DATA statements (which uses twice as much memory - once for the text - once for the variable). If you set the data pointer to were ever you wish, not just the start of lines, you could save a great deal of memory by just reading the required information when needed. This technique is illustrated by Listing 1. A typical use could be for adventure games where many data statements are required or different locations, et cetera.

Secondly, if you can tell exactly where a line is in memory, you can modify that line when the program is running without the program halting. An elaborate "Passer" could be constructed which could make immediate and unplanned changes to variables, or permanent data statements could be constructed, enabling a game or program to be saved with important variables intact. Listing 2 illustrates this second possibility. Care must be taken when altering a basic line so as to avoid placing zero bytes in the wrong locations (anywhere other than a numeric constant) or in overwriting the zero byte terminating the line. Doing either will destroy the program structure by intoducing new and strange basic lines.

This second technique has an advantage over using VARPTR if a string variable to do the same thing in that once determined, the location of the line remains constant and no space is used on the line to declare the variable. Be warned however, self altering programs are avoided by professional programmers as being bad practice.

THE KEYBOARD

To do anything with a computer you really need to know how to

utilise it's input peripherals. On the Spectravideo, these are the cassette, disk drives, joystick and keyboard. The first two take input out of the users direct control and will not be discussed here. The joystick is capable of a straight numeric function of 9 possibilities plus the fire button. However, not everyone owns a joystick (ignoring the onboard one). Lastly, this leaves us with the keyboard.

Basic has supplied the user with a number of ways of interpreting the output produced by the keyboard. Those are:

INKEY$ This produces a string of one character or null characters as input. This is best for realtime users where immediate input is required. There are two problems. The statement picks up inputs in advance which might not be needed and the statement ASC(INKEY$) or equivilent will produce an error if INKEY$ is null. A simple loop can eliminate these two problems but can delay input considerably.

INPUT$() Using INPUT$() is preferred when input is not directly realtime. No string is produced until a correct key is pressed so null strings are avoided. Using input lengths larger than 1 is discouraged because you can forget what you have typed. This is however just right for passwords.

INPUT# This is another alternative. Declaring OPEN"KYBD:FILENAME"FORINPUTAS#1 or equivilent will enable a file buffer to be used as keyboard input. Thus INPUT#1,A$ or INPUT#1,A will input a string or numeric from buffer #1 which just happens to be the keyboard. A$=INPUT$(##,1) will also work where ## is the number of characters input.

Several observations should be made about the above. Firstly, none echo the input on the screen. You need to use a PRINT A$ in conjunction to see what you have typed. All detect the ESC, CLS, INS, Cursor Keys & DEL as producing ASCII code instead of editing functions. INPUT# requires terminators to indicate the end of the variable input - this is either ENTER followed by any other key or a COMMA, so those characters cannot be part of the variable.

INPUT This is the old workhorse of basic and the most commonly (after INKEY$) input statement used. It allows a user prompt and you can see what you type in, allowing you to edit the input if incorrect. Its drawbacks are that it produces a questionmark (which might not be wanted) and that a comma cannot be part of input as it is used as a seperator of variables input, producing an error message if you input too many variables. Also, "INPUT A" will produce an error message if you input a string. This can be avoided by using INPUTA$:A=VAL(A$) and directing the logic to the correct program flow.

LINEINPUT The two differences between LINEINPUT and INPUT are that there is no question mark prompt and that COMMAs are considered part of a string input This is handy when inputting english sentances and the like.

LINEINPUT# Assuming you open a file as in INPUT#, you can also do a LINEINPUT# which has the same effect as INPUT# except the COMMA is not considered a terminator/seperator.

With INPUT & LINEINPUT, pressing the cursor keys will redirect your point of input and characters already on the screen could be considered part of the input string, even if you don't want them! Also, pressing CLS will clear the screen which could destroy your nice display. Furthermore, neither will work properly in SCREENs 1 & 2.

All the above is fine and you may already know it all, however how do you detect keys which do not produce ASC codes, such as the

SELECT key? The answer is simple, but dependent upon knowledge of the interupts. BASIC checks regularly for keyboard input and mirrors the results in a series of bytes in memory, from FD75H to FD7FH. Each key is represented as a bit in memory and is set to "1" when unpressed, "Ø" when pressed. Listing 3 demonstrates how this is set up. Listing 4 shows which keys are represented by which bits. Location FD74H is set to 15 if a new key is pressed. Unfortunely Listing 3 is not fast enough to show this properly. Detecting any key then merely requires a peek at the proper location and a check on bit status- "IFPEEK(&HFD7D)AND16=ØTHEN" checks if SELECT is pressed. Notice how locations FD7EH & FD7FH apply only to the numeric keypad on the SV328.

Alternatively, you can do an OUT &H96, ### : J=INP(&H99) to gain information directly from the keyboard interface; where ### indicates the "row" checked with the result being stored in J. This should also work in machine code. Replacing line 7Ø in Listing 3 with that in Listing 5 demonstrates this tecnique. If you are working in machine code, then the following routines may be of use:

4Ø3DH	Reads the keyboard and puts the ASC result in A
6D26H	Performs an INPUT
6D13H	Performs a LINEINPUT

In the last two cases the result is dumped into a buffer starting at F68EH. Proper use of the above routines will depend on your application and further knowledge of the routines. Get your Dissassembler out!

To round off this discussion of the keyboard, you might find the following two functions interesing:

OUT &H88,15 : OUT &H8C,255	Turns the caps light on
OUT &H88,15 : OUT &H8C,223	Turns the caps light off
POKE &HFE38,Ø	Turns capslock off
POKE &HFE38,non zero	Turns capslock on

Thus, you can force the format of your data entry, without any manual labor on the part of the user!.

MDM

Here is a Basic Keyword Called MDM. So far as I have discovered it can be used in any of the following four formats:

```
    MDM ON
    MDM OFF
    MDM STOP
  ON MDM GOSUB
```

It appears that MDM acts as an interupt in the same manner as STRIG or SPRITE, however what causes MDM to be set? My first impression was that MDM stood for MoDeM however this is hard to test. Not having seen an operating SV modem, I can't tell if MDM would be set by an incoming signal. Logically, you would open a modem file by using MDM: as the device prefix however this causes a bad file name" error. This may be because I have no modem attached. Opening a file 1: will also cause the same error if the disk drive isn't attached.

MDM therefore remains a mystery. Other mysteries are the Paddles that should be used with the Spectravideo. As of writing, no paddles have been seen in Western Australia. Tests with ATARI & APPLE paddles don't work. Obviously the PDL() statement should work with a paddle, but what is the range of the result? Also, should STRIG work with the Fire Button (if such exists) on the paddles? I was asked by another member about these points and as yet can't answer them. producing a set of paddles from scratch should be possible if the technical specs are known, a possible project for the electronic enthusiast.

I would be interested to hear from anyone who could clarify either MDM or PDL; it seems a pity that there are facets of our computers not used purely because of ignorance.

Next issue, I deal with the dreaded subject of machine code programs; how and where to load them, how to execute them and why you should.

LIST 1

by : L.A. Dunning

```
EP     1Ø REM Part 8, Listing 1
FN     2Ø REM illustrates READ pointer
AI     3Ø REM Shows how pointer can be altered
HB     4Ø CLEAR2ØØØ:WIDTH39:DEFINTA-Z:CLS:DIMD(2Ø),D$(2Ø)
BE     5Ø DEF FND$=HEX$(PEEK(&HF7F4)+PEEK(&HF7F5)*256):DEF FNI=VAL("&h"+FND$)
KL     6Ø DEF FNH=VAL("&h"+LEFT$(D$(X),2)):DEF FNL=VAL("&h"+RIGHT$(D$(X),2))
FJ     7Ø RESTORE:FORX=ØTO2Ø:D$(X)=FND$:D(X)=FNI:READA$:NEXT
GG     8Ø LOCATEØ,2:PRINT"Which data statement do you wish to     read <Ø - 2Ø
          >"
LI     9Ø LOCATEØ,4:INPUTN$:X=ABS(VAL(N$)):IFX>2ØGOTO9Ø
AG    1ØØ POKE&HF7F4,FNL:POKE&HF7F5,FNH
FN    11Ø READA$:LOCATEØ,6:PRINTUSING"Pointer:######    \  \H:  Data \         \
          ";D(X);D$(X);A$:GOTO9Ø
EE    12Ø DATA zero,one,two,three,four,five,six,seven,eight,nine,ten
IB    13Ø DATA eleven,twelve,thirteen,fourteen,fifteen,sixteen,seventeen,eigh
          teen,nineteen,twenty
END
```

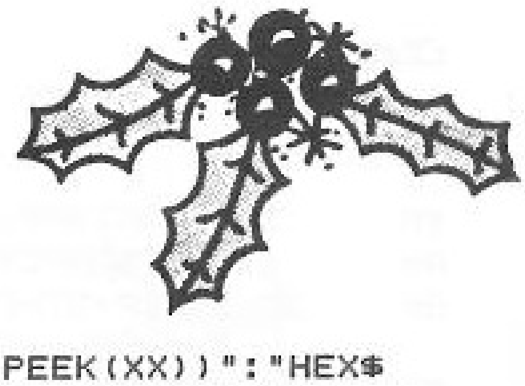
LIST 2

by : L.A. Dunning

```
FA     1Ø REM Part 8, Listing 2
FN     2Ø REM illustrates READ pointer
FE     3Ø REM Shows how DATA can be changed       permanently while program ru
          ns
CH     4Ø CLS:DEFSTRS:DEFSNGD:SL=CHR$(27)+"1":DIM D(5),S(5)
BP     5Ø PRINT"Existing Data on line 11Ø":RESTORE 11Ø:READA$:FORA=1TO5:D(A)=
          PEEK(&HF7F4)+PEEK(&HF7F5)*256:READS(A):PRINT","S(A);:NEXT:PRINT
FH     6Ø LOCATE Ø,7:PRINT"Input 5 new strings:":FORA=1TO5
AL     7Ø LOCATE Ø,9:PRINTSL;:INPUTS(A):S(A)=LEFT$(S(A),16):IFS(A)=""GOTO7Ø
BF     8Ø NEXT
IK     9Ø FORA=1TO5:FORX=1TO16:POKED(A)+X,32:NEXT:FORL=1TOLEN(S(A)):POKED(A)+
          L,ASC(MID$(S(A),L,1)):NEXT:NEXT
DJ    1ØØ PRINT"New Data on line 11Ø":RESTORE 11Ø:READA$:FORA=1TO5:READS(A):P
          RINT","S(A);:NEXT:PRINT:PRINT:LIST11Ø
BP    11Ø DATA,1................,2................,3................,4.........
          .......,5................,
END
```

LIST 3

by : L.A. Dunning



```
FB     10 REM Part 8, Listing 3
AO     20 REM illustrates KEYBOARD MATRIX
IE     30 REM Shows memory locations
IK     40 CLS:CLEAR1000:DEFINTA-Z
BA     50 DEF FNB$(X)=STRING$(8-LEN(BIN$(X)),"0")+BIN$(X)
GD     70 LOCATE0,3,0:FORB=0TO11:XX=&HFD74+B:PRINT"      "FNB$(PEEK(XX))":"HEX$
          (XX):NEXT
AM     80 GOTO70
END
```

LIST 4

by : L.A. Dunning

```
BE     10 CLS:CLEAR1000:FX$="keys"
FK     20 LOCATE0,7,0:PRINT"Which Display do you want":PRINT:PRINT"<1>  Scree
          n":PRINT:PRINT"<2>  Printer"
AB     30 A$=INPUT$(1):A=VAL(A$):IFA<1ORA>2GOTO30
NM     40 IFA=1THENFX$="crt:"+FX$:RESTORE1000ELSEFX$="lpt:"+FX$:RESTORE2000
GH     50 READ A$,B$,C$,D$,EX$:CN$=CHR$(13)+STRING$(A,10):OPENFX$FOROUTPUTAS#
          1
BN    110 PRINT#1, "        <------------BITS----------->":         PRINT#1,"ADDF
          . 7   6   5   4   3   2   1   0":          PRINT#1,"----  -   -   -
          -   -   -   -   -"CN$;
EB    120 PRINT#1,"FD74                          <-Key Press->"CN$;
CJ    130 PRINT#1, "FD75  7   6   5   4   3   2   1   0"CN$;
CD    140 PRINT#1, "FD76  /   .   =   ,   '   :   9   8"CN$;
DI    150 PRINT#1, "FD77  G   F   E   D   C   B   A   -"CN$;
CD    160 PRINT#1, "FD78  O   N   M   L   K   J   I   H"CN$;
CB    170 PRINT#1, "FD79  W   V   U   T   S   R   Q   P"CN$;
CK    180 PRINT#1, "FD7A "A$" bks  ]   \   [   Z   Y   X"CN$;
DD    190 PRINT#1, "FD7B "B$" ent stp esc rgh lgh ctl shf"CN$;
FK    200 PRINT#1,"FD7C "C$" ins cls fk5 fk4 fk3 fk2 fk1"CN$;
FO    210 PRINT#1,"FD7D "D$" ??? prt sel cap del tab spc"CN$;
BL    220 PRINT#1,"FD7E  7   6   5   4   3   2   1   0"CN$;
BL    230 PRINT#1,"FD7F  ,   .   /   *   -   +   9   8"CN$;
CF    240 PRINT#1,"ent = enter keys  stp = stop"
BI    250 PRINT#1,"bks = back space  tab = tab key"
EI    260 PRINT#1,"rgh = right grph  lgh = left grph"
HF    270 PRINT#1,"ctl = ctrl        shf = shift keys"
EA    280 PRINT#1,"cap = caps lock   spc = spacebar"
BH    290 PRINT#1,"prt = print       sel = select"
BC    300 PRINT#1,"fk# = function keys 5 - 1"
EO    305 IFA=2THENPRINT#1,EX$:PRINT#1,"----------------------------------------
          -----------------------------------------"
AG    310 CLOSE
AD    320 GOTO320
NP   1000 DATA" ↑ "," ← "," ↓ "," → ",
JA   2000 DATA UAR,LAR,DAR,RAR, U/L/D/R / AR = Up/Left/Down/Right  Arrow keys
END
```

LIST 5

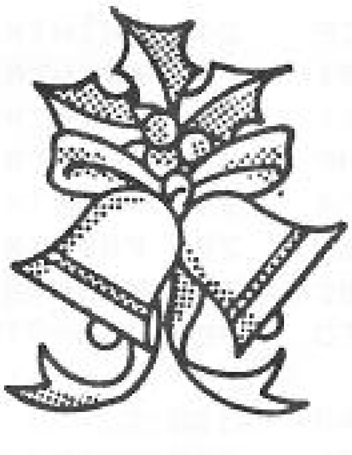
by : L.A. Dunning

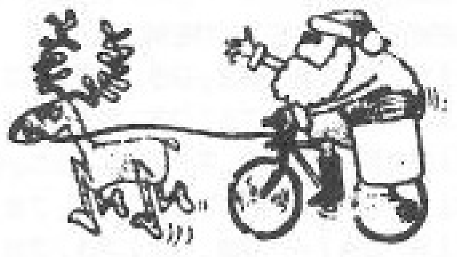
```
CC     70 LOCATE0,3,0:FORB=0TO11:OUT&H96,B:XX=INP(&H99):PRINT"      "FNB$(XX):N
          EXT
END
```

Menu

by : S.W. McNamee

```
BH      5 CLEAR2ØØØ:ONERRORGOTO3Ø:GOTO1ØØØ
AN      6 ONERRORGOTOØ:LOCATE,,1:SCREEN,1:END
GK     3Ø IFERR=5THEN RESUMENEXT ELSE PRINT:PRINT"File Error":FORN=1TO3ØØ:NEX
          TN:RESUME1ØØ:'E=ERR:PRINTERL:ERROR E
BG     6Ø RETURN
BE     7Ø RETURN25ØØ
EJ     8Ø RETURN1ØØ
GB     9Ø COLOR12,1:SCREENØ,1
BI    1ØØ SCREENØ,1
CF    11Ø CLS:POKE-456,Ø:PRINTCHR$(27)+"p":LOCATE1Ø,Ø:PRINT" FUNCTION MENU ":
          PRINTCHR$(27)+"q"
DB    12Ø PRINT:PRINT
JG    13Ø PRINT"Color change menu"TAB(3Ø)" ... 1":PRINT
KK    14Ø PRINT"Function key menu"TAB(3Ø)" ... 2":PRINT
IC    15Ø PRINT"Change key defenition"TAB(3Ø)" ... 3":PRINT
ND    16Ø PRINT"Print disk 1 files menu"TAB(3Ø)" ... 4":PRINT
LH    165 PRINT"Print disk 2 files menu"TAB(3Ø)" ... 5":PRINT:GOTO19Ø
CK    17Ø PRINT"Calculator"TAB(3Ø)" ... 5":PRINT:
FB    18Ø PRINT"Print disk library"TAB(3Ø)" ... 6":PRINT
JM    19Ø LOCATE5,2Ø:PRINT"Press the key No. desired"
AE    2ØØ S$=INKEY$:IFS$=""THEN2ØØ
HF    21Ø S=VAL(S$):IFS<1ORS>5THEN2ØØ
CK    22Ø ONSGOTO11ØØ,9ØØØ,3ØØØ,4ØØØ,4Ø1Ø,6ØØØ,7ØØØ,8ØØØ,9ØØØ
AF   1ØØØ ONSTOPGOSUB6:STOPON:GOSUB2ØØØ:GOSUB8ØØØ:GOTO9Ø
FN   1Ø1Ø KEY1,"MENU"
FL   1Ø2Ø KEY3,""
FN   1Ø3Ø KEY2,""
HL   1Ø4Ø KEY5,"PROGRAM"
AM   1Ø5Ø GOTO9Ø
FG   11ØØ CLS:PRINTTAB(1Ø):PRINTCHR$(27)+"p"+" COLOR CHANGE MENU "+CHR$(27)+"
          q"
GB   111Ø PRINT:PRINT
DH   112Ø PRINTTAB(5)"FOREGROUND"TAB(2Ø)"BACKGROUND"
FA   113Ø PRINT
PB   114Ø PRINT"Green"TAB(15)"Black"TAB(32)" ... 1"
PH   115Ø PRINT"White"TAB(15)"Black"TAB(32)" ... 2"
DO   116Ø PRINT"White"TAB(15)"Blue"TAB(32)" ... 3"
PB   117Ø PRINT"Black"TAB(15)"Green"TAB(32)" ... 4"
PH   118Ø PRINT"Black"TAB(15)"White"TAB(32)" ... 5"
EA   119Ø PRINT"Black"TAB(15)"Yellow"TAB(32)" ... 6"
EJ   1192 PRINT"Magenta"TAB(15)"Yellow"TAB(32)" ... 7"
GG   1194 PRINT"Yellow"TAB(15)"Magenta"TAB(32)" ... 8"
EB   1196 PRINT"Green"TAB(15)"Yellow"TAB(32)" ... 9"
AG   121Ø PRINT:PRINT"Return to menu"TAB(29)" < Esc >"
GM   13ØØ LOCATEØ,2Ø:PRINT"Press selection number";
EE   131Ø A$=INKEY$:IFA$=""THEN131Ø
AK   1315 IFA$=CHR$(27)THEN1ØØ
EL   132Ø A=VAL(A$):IFA<1ORA>9THEN131Ø
EE   133Ø ONAGOTO141Ø,142Ø,143Ø,144Ø,145Ø,146Ø,147Ø,15ØØ,151Ø
DL   141Ø COLOR12,1:GOTO1ØØ
DP   142Ø COLOR15,1:GOTO1ØØ
```





```
EE   143Ø COLOR15,5:GOTO1ØØ
DE   144Ø COLOR1,12:GOTO1ØØ
DI   145Ø COLOR1,15:GOTO1ØØ
DE   146Ø COLOR1,1Ø:GOTO1ØØ
CK   147Ø COLOR13,1Ø:GOTO1ØØ
BM   15ØØ COLOR1Ø,13:GOTO1ØØ
CB   151Ø COLOR12,11:GOTO1ØØ
HC   2ØØØ KEY6,"LOAD"+CHR$(34)+"2:"
JL   2Ø1Ø KEY7,"SAVE"+CHR$(34)+"2:"
FN   2Ø2Ø KEY8,"COLOR12,1"+CHR$(13)
DK   2Ø3Ø KEY1Ø,"GOTO"
II   2Ø4Ø KEY4,"print"+CHR$(34)
IJ   2Ø5Ø KEY1,"run"+CHR$(13)
GA   2Ø6Ø KEY9,"SCREENØ,1"+CHR$(13)
GP   2Ø7Ø KEY3,"LOCATE"
GF   2Ø8Ø KEY2,"list"
LE   2Ø9Ø KEY5,"print":RETURN
GJ   25ØØ GOSUB2ØØØ:CLS:NEW
KJ   3ØØØ CLS:KEY(1)OFF:PRINT"Input Ø , Ø to return to menu.":PRINT:PRINT
CN   3Ø1Ø LOCATEØ,5:INPUT"Input Key No. , Defenition.";KN,D$:IFKN=ØTHEN1ØØ
DI   3Ø2Ø IFKN<ØORKN>1ØTHENBEEP:PRINT"Illegal Key No.":FORN=1TO5ØØ:NEXTN:GOTO
          3ØØØ
ND   3Ø3Ø KEYKN,D$:BEEP:PRINT"Key";KN;"= ";D$:FORN=1TO5ØØ:NEXTN:GOTO3ØØØ
DJ   4ØØØ DV=1:GOTO42ØØ
DJ   4Ø1Ø DV=2:GOTO42ØØ
AE   42ØØ POKE-456,1
FI   42Ø5 SCREEN,Ø:ERASED$:DIMD$(36):D=Ø
AI   421Ø FORS=1TO13
FO   422Ø F$=DSKI$(DV,2Ø,S)
FD   423Ø FORP=1TO256STEP16
AO   424Ø PC=ASC(MID$(F$,P,1)):IFPC=ØTHEN44ØØ
CE   425Ø IFPC=255THEN45ØØ
BD   426Ø D$(D)=MID$(F$,P,9):D=D+1
IE   44ØØ NEXTP,S
HL   45ØØ CLS:PRINTTAB(1Ø)CHR$(27)+"p"+" DISK"DV"FILE SELECTION "+CHR$(27)+"q
          "
GA   451Ø PRINT:PRINT
BE   452Ø FORD=ØTO17:IFD+48>57THENDC=D+55ELSEDC=D+48
AL   453Ø IFD$(D)=""THEN47ØØ
GH   454Ø PRINTD$(D)TAB(11)" ... "+CHR$(DC)
GB   455Ø NEXTD:LOCATEØ,4
GN   462Ø FORD=18TO36:IFD+48>57THENDC=D+55ELSEDC=D+48
AM   463Ø IFD$(D)=""THEN47ØØ
FG   464Ø LOCATE2Ø,D-15:PRINTD$(D)TAB(32)" ... "+CHR$(DC)
FE   465Ø NEXTD
AP   47ØØ LOCATEØ,22
IO   471Ø PRINT"Make selection by pressing key or":PRINT"Press <ESC> to retur
          n to menu.";
EE   472Ø A$=INKEY$:IFA$=""THEN472Ø
AK   4725 IFA$=CHR$(27)THEN1ØØ
CP   473Ø DC=ASC(A$):IFDC>57THEND1=DC-55ELSED1=DC-48
FI   474Ø IFD1=>DTHEN472Ø
BA   4745 IFMID$(D$(D1),7,3)="DAT"ORMID$(D$(D1),7,3)="dat"THEN472Ø
AE   475Ø IFDV=1THEND1$="1:"+D$(D1)ELSED1$="2:"+D$(D1)
BH   476Ø RUND1$
CG   5ØØØ GOTO1ØØ:INPUTA$
EH   5Ø1Ø GOTO1ØØ:A$="1Ø"+A$
```

```
AE   6ØØØ GOTO1ØØ:'reserved for future expansion
AD   7ØØØ GOTO1ØØ:'reserved for future expansion
EL   8ØØØ RESTORE1ØØØØ:FORN=ØTO2:FORL=1TO1Ø
DE   8Ø1Ø READK$(N,L):NEXTL,N:RETURN
AD   85ØØ '  KEY MENU 1
DF   851Ø DATA 82,85,78,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY1
CP   8511 DATA 76,73,83,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:' KEY2
DB   8512 DATA76,79,67,65,84,69,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY3
DF   8513 DATA 8Ø,82,73,78,84,34,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY4
DG   8514 DATA 8Ø,82,73,78,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY5
DI   8515 DATA 76,79,65,68,34,49,58,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY6
DJ   8516 DATA 83,65,86,69,34,49,58,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY7
CK   8517 DATA 67,79,76,79,82,49,5Ø,44,49,13,Ø,Ø,Ø,Ø,Ø:'KEY8
BP   8518 DATA 83,67,82,69,69,78,48,44,49,13,Ø,Ø,Ø,Ø,Ø:'KEY9
AG   8519 DATA 71,79,84,79,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY1Ø
AD   86ØØ '  KEY MENU 2
DM   861Ø DATA 67,79,76,79,82,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY1
DC   8611 DATA 65,85,84,79,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY2
DO   8612 DATA 71,79,84,79,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY3
CO   8613 DATA 76,73,83,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY4
DE   8614 DATA 82,85,78,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY5
CJ   8615 DATA 67,79,76,79,82,49,53,44,52,44,52,13,Ø,Ø,Ø:'KEY6
BP   8616 DATA 67,76,79,65,68,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY7
CM   8617 DATA 67,79,78,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY8
CG   8618 DATA 76,73,83,84,46,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY9
FE   8619 DATA 12,82,85,78,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY1Ø
AD   87ØØ ' KEY MENU 3
EA   871Ø DATA 76,79,65,68,34,49,58,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY1
CN   8711 DATA 76,73,83,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:' KEY2
DF   8712 DATA 7Ø,73,76,69,83,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY3
DG   8713 DATA 8Ø,82,73,78,84,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY4
DD   8714 DATA 82,85,78,13,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY5
DJ   8715 DATA 83,65,86,69,34,49,58,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY6
CI   8716 DATA 67,79,76,79,82,49,5Ø,44,49,13,Ø,Ø,Ø,Ø,Ø:'KEY7
DE   8717 DATA 83,67,82,69,69,78,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY8
DN   8718 DATA 71,79,84,79,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:'KEY9
EJ   8719 DATA 67,

END
```

Menu Program

This is a program for Disk owners and is slightly more technical than your normal Newsletter program. With this program you may read the disk Directory and select a program from the screen menu. This program is very handy when used with the IPL program. Anyway it does alot more than mentioned above , so just type it in and experiment with it.

REVIEW OF "LOOPING"

by J. Collins.

For the COLECOVISION game console or the adaptor on Spectravideo computers.

LOOPING is the name, and you certainly can spend some of your game time doing just that once you learn to 'fly' your plane....in fact a good tight outside loop may be just what you need to avoid one of the attacking balloons....but it does not give you any real idea of what the game is about....read on.

After the usual menu screen for game difficulty level etc your first screen has your 'plane on the runway in a futuristic looking city...this screen appears to be three dimensioned. Take off using the joystick and first task is to open the gate to the next screen...that's right player...before you do anything else you'd better get the gate open. To do this you have to shoot and destroy a single rocket while this is sitting on its' launch pad and in the first and second levels this is fairly easy. In arcade level it becomes darn difficult.

Anyhow once destroyed, the 'gate open' message is flashed on your screen and you can then set about shooting up blue balloons which appear from the bottom of your screen and try to collide with your plane....they go up, down, and also right and left and will get you if you're careless. While shooting up the balloons you score points and can build up quite a score once you learn to 'fly' but to get to the next screen you must also fly right until the open gate appears. Line up the entrance and zoom right in....Now your world changes to a maze-like screen of thick blue pipes.

Fly through this maze if you can being careful not to touch the pipes, the roof, or the floor...look out as you fly horizontally for the whole screen scrolls from right to left or vice versa, depending on your direction and this makes life difficult when you turn a corner or change direction. At one point in the pipe maze you enter a 'room' protected by two giant taps which let go very large, very deadly, green 'blobs' and of course these don't do your 'plane any good at all if they hit...in fact any collision with any object explodes you right back to the start screen to start over.

Having dodged the 'blobs' you enter the last room of the maze and here you are faced with a set of bad-tempered bouncing balls. (I couldn't think of a better way to describe these balls...they even sound bad-tempered). Shoot as many as you can, fly round the rest and aim for any one of three very small gaps in the room walls. If you manage this and don't meet any more angry balls then you must fly to a vertical blue line and touch it with the nose of the plane. Success music, lights, and hey presto you're back at start looking right down the barrel of doing all that over again at the next level of difficulty.

I wont spoil it for you by taking you through the more difficult levels....suffice to say that like all of these games they do get more difficult as you get more skilled and that's the beauty of them anyhow....let me say that 'LOOPING' can be just as addictive as any of them and rates as a darn good game requiring a fair amount of practice to get anywhere near the high scores. Once you get a little skilled you can do everything faster by holding down one of the fire buttons which acts as an accelerator....speed can also get you out of any tight situations with balloons, blobs, or bouncing balls, but it must be used carefully. Graphics are excellent on each screen and sound is better than most other games I've played....listen to the planes' engine note when you do an outside loop.

'LOOPING' needs at least one QUICKSHOT-III joystick to select game level but second player can use any compatible stick. At around $40..00 depending on where you shop it is good Colecovision value.

More soon................

Program of the Month

In this game you are a little Gremlin called Multinash and you've got yourself lost!! To find your way home you must eat the scenery while dodging a horrible little Gremlin called Conservator who is rather partial to keeping the scenery untouched. If he catches you than you lose a life.

All the scenery contains an abundance of Vitamin A, which is good for your sight. When you've eaten enough you will be able to see a Yellow Door, which has no Vitamin A in it, behind which is a tunnel. When you've eaten all the door up you are sucked through the tunnel and the game starts again only this time the Conservator is more aggressive. After five screens you get Home.

On the right hand side of the screen is a gauge which shows how much Vitamin A is in your blood and on the left hand side of the screen is the number of lives you have left (the number coincides with the number of Blue Dots). You can use any joystick port but it can not be changed during the game. To eat something just position Multinash over it and press the Fire Button.

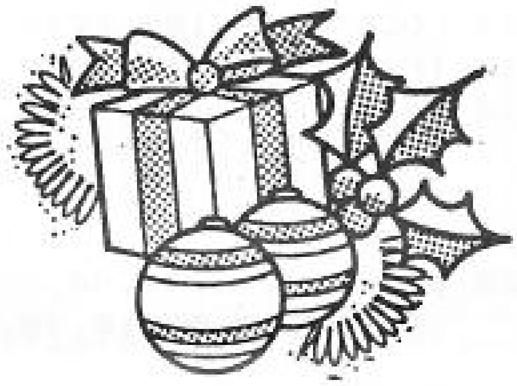
NOTE: Line 1Ø5 prints the S.V. Logo and it's a good idea to leave it out until you have the program typed in and saved correctly.

Conservator & Multinash

by : M.A. Perrett

```
DB     10 REM ***************************
FM     20 REM *                         *
EA     30 REM * File name  : c&m.       *
GA     40 REM * Date       : 18/Nov./84 *
BM     50 REM * Writen by  : M.Perrett  *
GA     60 REM * Bytes used : 3.8K       *
GB     70 REM *                         *
DI     80 REM ***************************
GD     90 REM
BN    100  COLOR 15,1,1:SCREEN 1:BEEP
AE    101 LOCATE 62,120:PRINT"Conservertor"
DH    102 LOCATE125,130:PRINT"and"
AG    103 LOCATE142,140:PRINT"Multinash"
FC    104 LOCATE 72,154:PRINT"By ... M.A.Perrett"
CL    105 ON STOP GOSUB 242:STOP ON:IF PEEK(&H4788)=&H3ETHENDEFUSR=&H4788:A=U
          SR(Ø)ELSE FOR I=ØTO1ØØØ:NEXT
AO    106 DEFINT A-Z:FORI=ØTO13:SOUNDI,Ø:NEXT
DH    107 COLOR 15,1,1:SCREEN 1:STOP OFF
FE    108 BEEP:SOUND 1,1:SOUND 8,7
BI    109 LOCATE 88,5Ø :PRINT "LEVEL OF SKILL"
CD    11Ø LOCATE 88,51 :PRINT "LEVEL OF SKILL"
DI    111 LINE (82,6Ø)-(176,62),13,BF
AM    112 LOCATE 88,8Ø :PRINT"1 = PRACTICE"
CL    113 LOCATE 88,1ØØ:PRINT"2 = NORMAL"
CC    114 LOCATE 88,12Ø:PRINT"3 = ARCADE"
DE    115 IF INKEY$<>"" GOTO 115
GC    116 I=(I+1)MOD16:SOUND 1,I:A$=INKEY$:IF A$="" THEN 116
GD    117 S=VAL(A$):IF S<1OR S>3 THEN116
DJ    118 IF S=1 THEN S!=.1: SK=Ø :L=5 ELSE S!=S/5:SK=-1:L=4
CD    119 DATAØ,-1,1,-1,1,Ø,1,1,Ø,1,-1,1,-1,Ø,-1,-1
AD    12Ø DATA 79,1D,27,46,8A,1E,71,ØF,F5,3Ø,18,1D,27,C2,1E,1E,3C,7Ø,E8,C4,A2
          ,FØ,1C,EØ,5E,18,3Ø,7Ø,E8,86,FØ,FØ
BE    121 DATA 1E,62,2A,17,19,31,71,ØF,F1,31,19,1D,27,C2,2,1E,FØ,8C,A8,DØ,3Ø,
          18,1C,EØ,1E,18,3Ø,7Ø,E8,86,8Ø,FØ
AJ    122  COLOR 1,12,13:SCREEN1:SOUND 1,2
AH    123 LOCATE 98,7Ø:PRINT"PRESS YOUR"
BL    124 LOCATE 62,9Ø:PRINT"JOYSTICK'S FIRE BUTTON"
JN    125 LOCATE71,11Ø:PRINT"WHEN YOU ARE READY."
BF    126 G=-1:TIME=Ø
HE    127 FOR I=ØTO2: IF STRIG(I) THEN G=I
CI    128 NEXT:IF TIME>5ØØØ THEN RUN 1Ø6
AK    129 IF G=-1THEN 127
CG    13Ø COLOR 1Ø,1,12: SCREEN 2,2
AI    131 SOUND1Ø,16:SOUND11,2ØØ:SOUND 5,1
CG    132 SOUND 1,Ø:SOUND Ø,Ø:SOUND 9,7
CE    133 ONINTERVAL=5GOSUB153
AM    134 ONSPRITEGOSUB163
BH    135 ONSTRIGGOSUB185,185,185
BA    136 FORI=1TO8:READD(I),E(I)
CI    137 NEXT
DI    138 GOSUB 2Ø8
```

```
EC     139 RESTORE12Ø:FORI=ØTO 1: B$=""
BM     14Ø FORJ=1TO32
AH     141 READA$
DF     142 B$=B$+CHR$(VAL("&H"+A$))
DP     143 NEXT:SPRITE$(I)=B$
CE     144 NEXT
CL     145 GOSUB 243
AH     146 IFX>232THENX=232:A=-1ELSEIFX<5THENX=5:A=1
ED     147 IFX!>218THENX!=218:A!=ØELSEIFX!<17THENX!=17:A!=Ø
AJ     148 IFY>178THENY=178:B=-1ELSEIFY<5THENY=5:B=1
FD     149 IFY!>166THENY!=166:B!=ØELSEIFY!<17THENY!=17:B!=Ø
DE     15Ø IFTIME>12THENPSET(N,O),P
BJ     151 STRIG(G) ON
AC     152 GOTO146
BJ     153 SOUND3,2
CM     154 S=STICK(G)
AG     155 A=A+D(S):B=B+E(S)
KJ     156 X=X+A:Y=Y+B
CH     157 PUTSPRITE1,(X,Y),5
AJ     158 IFRND(1)<.4ANDSKANDS>ØTHENA!=A!+D(S):B!=B!+E(S)ELSEA!=A!+AC!(ABS(X!
           <X)):B!=B!+AC!(ABS(Y!<Y))
AE     159 X!=X!+A!:Y!=Y!+B!
CO     16Ø PUTSPRITEØ,(X!,Y!),9
BJ     161 SOUND3,5
BN     162 RETURN
DM     163 SPRITEOFF
BL     164 INTERVALOFF
GE     165 STRIG(G)OFF
GI     166 LINE(X-8,19Ø)-(X+2Ø,19Ø),11
JC     167 LINE(X-4,19Ø)-(X+16,19Ø),4
DC     168 IF Y<1THEN Y=1
CO     169 FOR I!=YTO 2ØØ
CK     17Ø PUT SPRITE1,(X,I!),15
AK     171 SOUND 2,I!
AC     172 NEXT:SOUND 2,Ø
FO     173 IF X!<1 THEN X!=1
CL     174 FORI!=X!TO 218
EB     175 PUTSPRITEØ,(I!,Y!),9
BA     176 SOUND 3,I!
CE     177 NEXT
JP     178 LINE(X-8,19Ø)-(X+2Ø,19Ø),6
CM     179 FORI=1TO 2ØØ:NEXT I
AG     18Ø PSET(6Ø,L*16),1
AO     181 L=L-1:IF L<ØTHEN195
AG     182 SOUND13,Ø:SOUND 3,2
BH     183 FORI=1TO1ØØ:NEXT
AD     184  GOTO 245
FN     185 STRIG(G)STOP: PSET(N,O),P
GH     186 N=X+7:O=Y+8:SOUND8,12:SOUND8,Ø
GN     187 P=POINT(N,O):TIME=Ø
GJ     188 IF P<2ORP=8THENPSET(N,O),14 :IF P<1 THEN P=Ø:RETURN ELSE RETURN
LB     189 PSET(N,O),13:SOUND 13,4
EA     19Ø IF P=1ØTHEND=D+1:P=8:IFD=48THEN2Ø2 ELSE RETURN
BK     191 SC=SC+1
JE     192 IF (SCAND127)=Ø THEN IF D=ØTHENLOCATE12Ø,9Ø:COLOR1Ø:PRINT"█"
HM     193 PSET(248,SC/4),SCAND15:P=1
BM     194 RETURN
```



```
DE    195 IF INKEY$<>"" GOTO 195
BC    196 SCREENØ,Ø: COLOR 5
IG    197 LOCATE1Ø,12,Ø:PRINT"Your score is"SC+SN*1ØØ
AB    198 LOCATE18,2Ø:PRINT"Hit Enter To Restart"
BB    199 TIME=Ø
EL    2ØØ IF INKEY$<>CHR$(13)ANDTIME<5ØØØ GOTO 2ØØ
AP    2Ø1 CLS: RUN 1Ø6
KA    2Ø2 INTERVALOFF:STRIG(G)OFF:SPRITEOFF
BK    2Ø3 SN=SN+1:IFSN=5 THEN 234
BJ    2Ø4 PUT SPRITE1,(Ø,2Ø9)
BJ    2Ø5 PUT SPRITEØ,(Ø,2Ø9)
DN    2Ø6 L=L+1:S!=S!+.1:D=Ø:GOSUB 2Ø8
CL    2Ø7 INTERVAL ON:SPRITE ON:GOTO 243
DM    2Ø8 GOSUB 219
LI    2Ø9 COLOR 11:LOCATE 1ØØ-H/2,5Ø+H:PRINT"●":H=H+3Ø
BE    21Ø R=RND(-6):FOR I=ØTO 9:PSET(RND(1)*22Ø,RND(1)*16Ø),11:NEXT
CG    211 X=8Ø :GOSUB 223
AA    212 IF SK=Ø THEN X=156:GOSUB 223
CG    213 X=19Ø:GOSUB 223
CF    214 X=6Ø :GOSUB 227
BN    215 X=128:GOSUB 227
CA    216 X=148:GOSUB 227
AH    217 LINE(Ø,19Ø)-(255,19Ø),6
CJ    218 RETURN
BE    219 K=8:FOR J=1TO2
EA    22Ø FOR I=94 TO Ø STEP-4
BH    221 LINE (I+32,I)-(224-I,192-I),K,B
BM    222 NEXTI:K=1:NEXT J:RETURN
OM    223 COLOR 2 :LOCATE X+Ø,126:PRINT"●"
IM    224 COLOR 12:LOCATEX+13,136:PRINT"♪"
AD    225 COLOR 6 :LOCATE X+Ø,158:PRINT"|"
CG    226 RETURN
FI    227 PSET(X+8,164),4
JO    228 LINE(X+4,168)-(X+12,168),4
FH    229 PSET(X+8,172),4
II    23Ø PSET(X+8,168),11
GE    231 LINE(X+Ø,18Ø)-(X+8,18Ø),3
KE    232 LINE(X+4,176)-(X+4,184),12
CC    233 RETURN
FD    234 SPRITE OFF:STRIG(G)OFF:INTERVALON
CH    235 SPRITE$(Ø)=""
DH    236 GOSUB 219
FA    237 COLOR 4:CLS:LOCATE Ø,6Ø:PRINT "  YOU ARE":PRINT:PRINT"    HOME"
CH    238 PLAY "T24ØO4c4d8e8f4g2c2R2O5d2c2
AM    239 FOR I=ØTO5ØØØ:NEXT
BM    24Ø INTERVAL OFF
AF    241 GOTO 195
CM    242 SCREEN Ø,1: COLOR 15,1:END
CM    243 Y=Ø:Y!=17Ø
DE    244 AC!(Ø)=-S!:AC!(1)=S!:O=255
AP    245 X=Ø:A=Ø:B=Ø: X!=218:A!=Ø:B!=Ø
AN    246 IF L THEN FORI=1TOL:PSET(6Ø,I*16),4:NEXT
DG    247 FORI=32TO192STEP32:PSET(24Ø,I),6:NEXT
EG    248 INTERVALON:SPRITE ON:RETURN
END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

buy sell or trade

FOR SALE

SV318 Computer
SV9Ø3 Cassette
SV6Ø1 Expander
SV9Ø2 Disk
SV8Ø1 Controller
SV8Ø3 16K Ram
PLUS Disks, Tapes
& Cartridges

Will Sell The Lot for $1ØØØ

Conact: F.W. Charters,
6 Driftwood Street,
Sunnybank Hills,
Brisbane 41Ø9.
Call : (Ø7) 273 1847

COLECO MOD

adapter

Coleco modification to allow the expander to remain plugged in the back & game programs to be run from disk is available from B.Parker Phone (Ø7) 349Ø315 for further details.

Modification cost is $6Ø or a D.I.Y. Book for $2Ø.