SVI & MSX

— SPECTRAVIDEO —

NEWSLETTER

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

ISSUE NO.
3 - 2

ANNUAL SUBSCRIPTION	
AUSTRALIA	$20.00
OVERSEAS	$25.00
OVERSEAS AIRMAIL ...	$30.00
YEAR BOOK 83/84	$15.00

DATE
NOV - 1985

CONTENTS

NEWSLETTER CORRESPONDENCE

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

LIBRARY CORRESPONDENCE

S.A.U.G. LIBRARY,
1 CONRAD AVENUE,
GEORGE TOWN,
TASMANIA, 7253.

(003) 822919

... Introduction ...

While usual Editor is busy moving into his new home and taking care of the thousand and one details that always means, I've been asked to look after the November issue. On the software front there is a very good program for 318/328 owners and one of equal quality for our MSX 'ers. As well there are some additions to our Library list for CP/M users in the form of some games to play when the programming gets out of hand and the dreaded BDOS Error on A: appears to be coming up too often.

The reviews of the MSX games again bring home the point that support for MSX is growing and many software houses are releasing new packages that we could only dream about for 318/328 machines. That much mentioned but never seen MSX Adaptor had better come out of the closet soon. Our source of info' on the Adaptor has dried right up so if any of the users has anything to add please don't hesitate to get in touch.

Mr L. A. Dunning of "Understanding BASIC" has been busy again and the results are in this issue. Please pay special attention to the point made about line 930 in the accompanying program. There are more articles ready for this series so those members who write in whenever the articles don't appear can rest assured that they will continue.

Warning ... Warning ... Warning

It has come to our notice from various sources and from our own experience that the "ZEN" Assembler has a serious bug. If you spend several hours typing in a thousand line source program (not too long for a serious machine-code program) and try to save it to tape as the program says you can do then you will be in for a nasty shock. The program is supposed to be able to write a standard ASCII file out to tape using the inbuilt command "W". If you use the "W" command to try and save your source program everything appears to work correctly and something is definitely written out onto the cassette tape. However, when you use either the "V" command or the "R" command to Verify or Load back your source program you will find as I did that it just does not work and your 'thousand' lines of input have been lost forever. I say lost because although they are safely in memory they may as well be lost if you can't write them to tape. You have to turn off your computer sometime. This is a real pain as you can imagine and it means that should you ever want to change even one byte of the program you have to type it all in again. If you use the "WB" command to save out the BINARY Object file first then that will be okay, and you can run the program providing it is bug-free, however next time you want to edit the program or correct a bug you are faced with the daunting task of entering it all over again. OZZISOFT have promised me they will get this BUG corrected or will withdraw the product from sale if it can't be fixed. So hang off buying ZEN till we can give you better news. I would be very interested in hearing from any more members who have found the same problem.

Exploring Basic Pt-14

by L.A. Dunning

This part deals with disk drives and how to cope with them. SV Disk drives seem to come in two varieties, those that break down and those that don't. Assuming you have a system that works, the following comments should be of use.

DISK SCANNER

Before reading the rest of this article, I would suggest you type in the listing in this issue. This will enable you look at disks while reading the article. One word of warning when using the program, it always assumes that there is a disk inside the disk drive to scan. If there isn't, it will hang until you place a formatted disk inside the drive. A handy advantage of the program, one I discovered by accident, is that it can be used to read CP/M disks as well. I will talk about this later.

FORMATS

There are two ways to format a disk so that it can be used in the disk drive. The first is in BASIC and the second is in CP/M. The same CP/M program is used to format both. The difference is that on a CP/M disk, the system file is the CP/M disk operating system; on a BASIC disk it is the BASIC disk operating system. Also, a BASIC disk has a seperate directory track. Each formatted disk is divided into 40 tracks of 17 sectors each and each sector contains 256 bytes of information. Thus, each disk can contain a maximum of 174,080 bytes. Of these, 4 tracks are reserved so this figure is reduced by 17,408 bytes to 156,672 bytes. The tracks are numbered from 0 to 39, with 0 being the innermost track and 39 being the outermost. The choice of number for sectors is really arbitrary however each track is set to the same sequence from 1 to 17. Tracks 0 to 2 contain the system file that is read when the system is booted up. Track 20 is the directory track for a basic disk.

Before going any further, format a new disk in the normal manner and setup the directory track as usual, but do not save any files on the disk yet. Load the DISK SCANNER program into memory and insert the newly formatted disk into drive 1, then run the program. Track 20 is used to record three tables that are used to reference files on the disk. These are the Directory table (DIR), the Disk Allocation Table (DAT) and the File Allocation Table (FAT).

DIRECTORY TABLE

The DIR starts at sector 1 and allowance is made to sector 13. On a normal disk you wont use more than the first 6 sectors for the DIR. On double sided drives with modified operating systems I can only make an educated guess (not possessing one myself) however 13 sectors would give enough room for all possible files plus more. If you are running the program you will see that the intial sector is set to track 20, sector 1, which is the start of the DIR. On the new disk this set to a block of FFH bytes. There are no files yet.

Each entry in the DIR consists of 16 bytes. The first 9 are used for the file name which is 6 characters plus and extension of 3 characters. Shorter file names are padded with blanks. The first character has a special purpose. If it is FFH, it indicates the end of the directory and only entries before this will be listed by FILES. If it is 00H, it indicates a deleted file and this is not listed either by FILES. Such an entry is used for new files before creating a new entry. The 10th byte is the attribute byte and is used to indicate the type of file. It is easier to note this byte in octal instead of hex. The following values indicate the types present:

000	ASC File or data file
001	Machine Code file
200	Binary File

In addition the following values indicate extra attributes:

010	Read after Write	"R"
020	Write Protect	"P"
040	Screen Dump	

A binary file is the normal method that a BASIC program is saved to disk. It is a direct dump from what is in memory. All screen dumps are also binary files. An ASC file is the method used to save a BASIC program using the "A" option. It prints to disk in the same way a program is LISTed to the screen. The advantage of ASCII files is that they are easier to manipulate, but use more sectors than a binary file. Data files using PRINT# and PUT# are also in ASCII format. A Machine code file is really just a binary file and represents a dump from memory. The difference is that it is dumped directly back to memory. The 11th byte is used to point to the first entry in the FAT for the file. Bytes 12 to 16 are "reserved for future use" which means they are untouched by a normal operating system and are normally left at FFH.

DISK ALLOCATION TABLE

This is sector 14 on track 20. Originally this is blanked to 00H. This table has only two entries. The first byte is used for disk attributes, for Read after Write and Write Protect options. It is set to the same values as the attribute byte in a DIR entry. The difference is that values in the DIR entry apply only to that file, whereas a value here applies to the entire disk. The second entry is the string executed by the IPL command. This is normally no more than 15 bytes long. Any shorter strings are padded by 00H's. The rest of this sector is not used.

FILE ALLOCATION TABLE

This table is repeated in sectors 15, 16 & 17 on track 20. Normally only the first 41 bytes are used. The first 40 bytes form a 'linked list' that indicates which tracks are used by which files. Each entry corresponds to one track on the disk, the first being track 0 and the last being track 39. Originally each entry on this table has one of two values. An FFH indicates a free track, a FEH indicates a reserved track. At the start, only tracks 0, 1, 2 & 20 are reserved. A reserved track is not used to store files on. As a file is created, the system decides on which track to start placing that file. Normally the system will pick unused tracks that are closest to the Directory Tract and then work outwards. The 11th byte in a DIR entry will point to the byte on the FAT corresponding to the first track used. If that

entry is a number from 0 to 39, it points to the next track and entry used. The entry for this next track will then point to the next track/entry and so on, until the file ends. If a track is the last one used for a file, its entry will be the number of sectors used (from 1 to 17) ORed with 300 octal. While a file is deleted, the first byte on the DIR entry is set to 00H and the entries on the FAT are set to FFH indicating a free track. The Attribute and pointer bytes in the DIR entry are untouched. The file itself is left on the tracks until overwritten by another file. If you KILL a file by mistake, you can use DISK SCANNER or a similar utility to restore the file, by altering entries on the DIR and FAT. All that is required is a knowledge of where the file starts and what it should look like. This will take practice. The final byte in the FAT is set to 00H and indicates the end of the table. Presumerably on an 80 track disk, the FAT is 80 entries long followed by a zero byte. You will notice that the FAT is repeated three times. I believe this is to reduce the chances of losing files due to hardware or software bugs. If you wipe sector 15, copy sectors 16 or 17 to it. The system also periodically copies the DIR, so you may find a copy of sector 1 in sector 2. As new entries are added, deleted or changed, the copy becomes obsolete until once again copied.

FILES

Each file is saved to disk on a sector by sector basis. Each 256 bytes of the file is dumped to a sector of a track, starting at sector 1 and continuing until sector 17. When all sectors of a track are used, a new track is picked and the process repeated. This is made possible by first filling a buffer and then writting the contents to the disk. This is not done until either the buffer is full, or the file closed. Each file starts at sector 1 and if less than 17 sectors are used, the remaining sectors on that track are left unused except for expansion of that file. The first byte of a screen dump file is used to indicate the type of screen saved. It will be either 0, 1 or 2. The first six bytes of a machine code file are used to indicate information about the dump. The first two indicate the start of the dump, the second two indicate the length of the dump and the last two indicate the entry point of the code. Each pair comes in the usual Low byte/High byte manner of integer values.

The use of a linked list in the FAT enables a file to be on any available track of a disk instead of consecutive tracks. If you are using a disk as a data disk and do not need the system file on it, use DISK SCANNER or equivilant to change the entries for tracks 0 to 2 on the FAT to FFH. This will give you an extra 51 sectors of storage.

SYSTEM FILES

Tracks 0 to 2 are the system file that is used when the system is booted. If you look at track 0 you will see that the last 128 bytes of every sector is identical. When the computer is booted, it scans this track and loads its instructions from it, dumping the rest of the file into memory. You might use DISK SCANNER to change the message given when first displayed. You will find this on sectors 0/5 & 0/6.

As the program also reads CP/M disks, you can alter both the directory entries and function key definitions given on Track 0. These are in sectors 3/1 onwards, and 0/2 to 0/4 respectfully. Be very careful altering the first of these, or you might lose some files. It does not conform to the description given above!

DISK SCANNER

By now you have either saved some files to check the above, or you want to do so now. I would suggest first saving DISK SCANNER in normal format, then saving it in ASC. Then save the screen and dump a segment of memory to disk using BSAVE. Then delete a file. After each change, check the DIR and FAT to see how they are altered. In case you were wondering, diagram 1 gives a listing of the MCR used in the program. It works by dumping buffer #0 to the screen. Buffer #0 is used by the system for most disk input/output. The subroutine in lines 400-410 swap two NOP's for the JR DUMP at 0016, thereby altering the MCR flow. Lines 260-270 alter values in the MCR when it is being poked into buffer #1. You can place your own values here to change the origin and width of the dump. The reason it was done in this manner was so that you could also use the routine for your own purposes, namely reading a segment of memory. All that is needed is the starting byte and this is given by the argument in the J=USR() statements. Subroutines 700-710 read and restore the function keys to whatever values they were before the program. This uses a string array B(A,N) where A is the number of the function key and N is the number of definitions needed. You can read and restore any number of definitions just by increasing N. A dummy string ,BD, is required in line 700 to prevent an error from occuring.

NEXT TIME

Continuing on the subject of disks, I'll show how to get effective use of them from basic.

Disk Scanner

Except for line 930 this is a standard SVI BASIC program and can be entered using the INPUT program if you leave out line 930 until you have finished typing in the rest of the program. Line 930 makes special use of the Cursor Arrow Keys and you will have to follow suit. Set up your function keys so that F1 to F4 equate to the four graphics arrow characters as shown in your handbook to be ASCII characters 212, 213, 214, and 215. When entering line 930 where an arrow character is called for simply press the appropriate function key as you would an ordinary keyboard key. 'old' members of the group, and those with the yearbook could refer to the short article in Newsletter 1 - 11 for August,84. For new members and those too tired to leaf through the past issues I offer this ray of hope :-

```
KEY1,CHR$(212)
KEY2,CHR$(213)
KEY3,CHR$(214)
KEY4,CHR$(215)
```

DISK SCANNER

by : L.A. Dunning.

```
AG      10 REM DISK SCANNER
FM      19 REM Initialisation
DJ      20 WIDTH39:MAXFILES=2:CLEAR1000:DEFINTA-Z:DEFSTR B:DIM DX(15),DY(15),B(6)
        ,BK(10,1),I(1),D(1),T(1),S(1),BD:XA=VARPTR(#0)+9:XB=VARPTR(#1)+9:B="DI
        SK:":PX=14:PY=2:D=1:T=20:S=1:DT=1:MT=0:AA=0:FORA=0TO8:READ DX(A),DY(A)
        :NEXT:GOSUB210
FO      30 BS=CHR$(8):BC=CHR$(27):DEF FNBH(X)=STRING$(2-LEN(HEX$(X)),"0")+HEX$(X)
        :DEF FNBO(X)=STRING$(3-LEN(OCT$(X)),"0")+OCT$(X):VX=4:VY=PY+6
HP      40 N=0:GOSUB700:GOSUB930:KEY1," HEX ":ONSTOPGOSUB3000:STOP ON:KEY ON:DEFU
        SR=VARPTR(#1)+9:GOSUB730:FIELD#0,128ASB(0),128ASB(1)
HP      50 FORA=2TO6:READB(A):NEXT
AP      99 REM Viewing Routines
DA     100 D$=DSKI$(D,T,S)
DH     110 I$=INKEY$:J=USR(XA)
IJ     120 LOCATEPX+1,PY+18,0:PRINTUSING"D:##  T:##  S:##";D,T,S:GOSUB200:IFD1=0A
        NDD2=0GOTO120
DH     130 IFD2THEN D=3-D
BB     140 T=T+DY(D1):T=T-((T<0)-(T>39))*40:S=S+DX(D1):S=S-((S<1)-(S>17))*17
AC     150 GOTO 100
IA     199 REM General Routines #1
IJ     200 FORTL=1TO10:NEXT:D1=STICK(0)ORSTICK(1):D2=STRIG(0)ORSTRIG(1):RETURN
DD     210 XX=XB:CC=0
FF     220 READ BV:L=LEN(BV)
DN     230 IF L=3 GOTO 250 ELSE VV=VAL("&H"+BV)
CA     240 POKE XX+CC,VV:CC=CC+1:GOTO220
AF     250 IF BV="END"THENRETURN
IH     260 IF BV="wit"THENVV=16
EC     270 IF BV="vid"THENVV=40+PX+PY*40+2
AD     280 GOTO240
ID     399 REM General Routines#2
BC     400 IFDT=1 THEN POKEXX+22,0:POKEXX+23,0 ELSE POKEXX+22,&H18:POKEXX+23,&HF
BJ     410 DT=1-DT:KEY1," "+MID$("ASCHEX",1+DT*3,3):J=USR(XA):RETURN
BL     430 BEEP:GOSUB 740:CX=0:CY=0:LOCATEVX-1,VY-1:PRINT"Hex:Oct":LOCATEVX,VY+2:
        PRINT"Mod:":RETURN440
AL     440 I$=INKEY$:LOCATEPX+CX+1,PY+CY+1,1:GOSUB200
EA     450 GOSUB650:LOCATEVX,VY,0:PRINTFNBH(VV)":"FNBO(VV):GOTO440
GN     499 REM Modification Routines
CM     510 LOCATEPX+CX+1,PY+CY+1:PRINTBC"p"MID$("HA",AA+1,1)BC"q":VV$=""
HM     520 LOCATEVX+4,VY+2,1:PRINTVV$;:BI=INKEY$:AC=((PEEK(&HFD7D)AND16)=0):IFACT
        HEN AA=1-AA:GOTO510
CF     530 IFBI=""GOTO520
FI     540 IFBI=CHR$(27)GOTO640
AF     550 ON AA GOTO 630
DK     560 IFINSTR("0123456789ABCDEFabcdef",BI)ANDLEN(VV$)<2THENVV$=VV$+BI
ED     570 IF BI=BSANDLEN(VV$)>0THENVV$=LEFT$(VV$,LEN(VV$)-1)
BD     580 IFBI=" "ORBI=BS THEN IFLEN(VV$)<1THENVV$=VV$+HEX$(VV\16) ELSE GOSUB600
        :D1=3:GOSUB650:GOTO510
CM     590 IFBI<>CHR$(13)GOTO520
OJ     600 LL=LEN(VV$):NV=VAL("&h"+VV$):IFLL=2THENVV=NV
CE     610 IFLL=1THENVV=VVAND15:VV=VV+NV*16
AI     620 POKEXA+PP,VV:GOTO640
BO     630 VV=ASC(BI):GOSUB620:D1=3:GOSUB650:GOTO510
```

```
BJ    640 J=USR(XA):RETURN
CL    650 CY=CY+DY(D1):CY=CY-((CY<0)-(CY>15))*16:CX=CX+DX(D1):CX=CX-((CX<0)-(CX>
      15))*16:PP=CX+CY*16:VV=PEEK(XA+PP):RETURN
IN    660 DSKO$D,T,S:D$=DSKI$(D,T,S):J=USR(XA):RETURN
AE    670 BEEP:GOSUB730:RETURN100
HP    699 REM General Routines#3
IM    700 JK=VARPTR(BD):POKEJK,16:POKEJK+2,250:FORA=1TO10:POKEJK+1,14+A*16:BK(A,
      N)=BD:BK(A,N)=LEFT$(BK(A,N),INSTR(BK(A,N),CHR$(0))-1):NEXT:RETURN
FO    710 FORK=1TO10:KEYK,"":KEYK,BK(K,N):NEXT:RETURN
HB    730 MM=0:GOSUB760:LOCATE3,3:PRINT"Track   ":LOCATE3,4:PRINT"Sector     ":KEY
      2,"<COPY>":KEY3,"<MOD.>":KEY4,"<HELP>":KEY5,"<FREE>":ONKEYGOSUB400,800
      ,430,1000,830:RETURN
DK    740 MM=1:GOSUB760:LOCATE3,3:PRINT"Up/Down":LOCATE3,4:PRINT"Left/Right":KEY
      2,"<COPY>":KEY3,"<VIEW>":KEY4,"<ALT.>":KEY5,"<BURN>":ONKEYGOSUB400,800
      ,670,510,660:RETURN
DA    750 MM=2:GOSUB760:LOCATE3,3:PRINT"          ":LOCATE3,4:PRINT"          ":KEY
      3,"<MOD.>":KEY2,"<VIEW>":KEY4,"<SLCT>":KEY5,"<DUMP>":ONKEYGOSUB400,670
      ,430,860,820:RETURN
BJ    760 LOCATE0,0:PRINTMID$("VIEWMOD.COPY",1+MM*4,4)" MODE":FORA=5TO20:LOCATE0
      ,A:PRINTSPC(12):LOCATEVX,VY:PRINT"  ";:NEXT:RETURN
ID    799 REM Copy Routines
AK    800 BEEP:GOSUB750:I(0)=0:I(1)=0:RETURN810
BF    810 I$=INKEY$:GOTO 810
LA    820 IF I(0)ANDI(1) THEN D$=DSKI$(D(0),T(0),S(0)):DSKO$D(1),T(1),S(1):D=D(1
      ):T=T(1):S=S(1):J=USR(XA):RETURN ELSE RETURN
AP    830 T=20:S=15:RETURN100
DG    840 PRINTBS;BS;:LINEINPUTVV$:VV=ABS(VAL(LEFT$(VV$,2))):IFINSTR(VV$,"*")THE
      NI(E)=0:E=2:RETURN920 ELSE RETURN
BB    860 FORE=0TO1
AI    870 LOCATE0,6+E*6,1:PRINTMID$("FROMTO  ",1+E*4,4)":"
AE    880 PRINTUSING"  DISK:# ";D(E);:GOSUB840:D(E)=VV
AG    890 PRINTUSING" TRACK:##";T(E);:GOSUB840:T(E)=VV
FF    900 PRINTUSING"SECTOR:##";S(E);:GOSUB840:S(E)=VV
NK    910 I(E)=1+(D(E)<1ORD(E)>2ORT(E)>39ORS(E)<1ORS(E)>39):IFI(E)=0GOTO870 ELSE
       D$=DSKI$(D(E),T(E),S(E)):J=USR(XA)
AI    920 NEXT:LOCATE,,0:RETURN
DK   1000 CLS:PRINT" DISK SCANNER INSTRUCTIONS":PRINT:PRINT" This program allows
       the user to view, modify and copy disk sectors on an SV  902 disk dri
      ve.":PRINT
HG   1010 PRINT" There are 3 modes to use, press":PRINT"<0> to return to VIEW":P
      RINT"<1> for VIEW information":PRINT"<2> for MOD.":PRINT"<3> for COPY"
BB   1020 PRINT:PRINT" ^STOP to exit program":PRINT:PRINT" Each sector is displa
      yed in a 16x16  grid in either HEX or ASC display. HEX dumps each val
      ue as a straight patten"
BL   1030 PRINT"value. ASC shows ASCII characters in  the proper manner and inv
      erts values 0 to 31."
CD   1040 VV$=INPUT$(1):VV=VAL(VV$):IFVV=0THENGOSUB930:GOSUB760:J=USR(XA):RETURN
       ELSE ONVVGOTO1100,1200,1300
IN   1100 CLS:PRINT" VIEW MODE":PRINT:PRINT" This allows quick viewing of the di
      sk sectors. The arrow keys or joystick  will alter the track/sector
      viewed.  The Spacebar or firebutton will alter  which disk is viewed
      ."
CP   1110 PRINT:PRINTB(6):PRINT:PRINTB(2):PRINT"F2 "B(3):PRINT"F3 "B(4):PRINT"F4
       Go to HELP display":PRINT"F5 Set Track/Sector to 20/15"
AD   1140 GOTO1400
FJ   1200 CLS:PRINT" MOD. MODE":PRINT:PRINT" This allows you to view the content
      s  of individual sectors in greater detailand alter them. You may use
      the select key to choose either Hex or ASC style  of input while in AL
      TER."
```

```
HG    1210 PRINT:PRINTB(6):PRINT:PRINTB(2):PRINT"F2 "B(3):PRINT"F3 "B(5):PRINT"F4
        ALTER a row in the buffer":PRINT"F5 BURN the buffer to the sector"
CD    1220 PRINT"   currently displayed"
AC    1240 GOTO1400
EM    1300 CLS:PRINT" COPY MODE":PRINT:PRINT" This allows you to copy one sector
       of  a disk to another sector on either the same or different disk."
EC    1310 PRINT:PRINTB(6):PRINT:PRINTB(2):PRINT"F2 "B(5):PRINT"F3 "B(4):PRINT"F4
        SeLeCT origin and destination          sectors":PRINT"F5 DUMP 1st sec
       tor to 2nd sector"
AB    1340 GOTO1400
BO    1400 LOCATE0,21:PRINT"PRESS <ENTER> TO CONTINUE";
BF    1410 VV$=INPUT$(1):IFVV$<>CHR$(13)GOTO1410ELSEGOTO1000
BG    1990 RETURN
GL    2000 DATA 0,0,0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1
AI    2010  DATA CD,C3,1C,E5,DD,E1,01,00
CM    2020  DATA 00,11,28,00,21,vid,00,CD
EP    2030  DATA 3C,37,DD,7E,00,C5,18,0F
DP    2040  DATA 0E,7E,B9,30,0A,0E,20,B9
EB    2050  DATA 38,03,91,18,02,C6,60,D3
GM    2060  DATA 80,C1,0C,DD,23,3E,wit,B9
CG    2070  DATA 20,06,19,CD,3C,37,0E,00
BF    2080  DATA 10,D8,C9,END
JC    2090 DATA F1 Change Display from HEX/ASC,Go to COPY mode,Go to MOD. mode,Go
        to VIEW mode,Function Keys will do the following-
EF    3000 LOCATE 0,20,1:GOSUB710
END
```

This is line 930 - Type this in now and don't forget about the arrows

```
930 CLS:LOCATEPX+1,0:PRINT"DISK SCANNER":LOCATEPX+1,1:PRINT"by L.A.Dunning
 ":LOCATE0,3:PRINT"↑↓":PRINT"←→":LOCATEPX,PY:PRINT" ┌"STRING$(16,"─")"
 ┐":FORA=1TO16:LOCATEPX,PY+A,0:PRINT" │"SPACE$(16)"│";:NEXT:LOCATEPX,PY+
 17:PRINT" └"STRING$(16,"─")"┘":RETURN
```

The other characters in line 930 are standard SVI Graphics accessed using the Left & Right Graph Keys plus the appropriate other keys on the keyboard. Look them up in your handbook.

ASTRAL ATTACK (for MSX)

by : Andrew Lacey.

This program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84) or the Year Book.

```
CJ     100 REM ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
CK     110 REM ■■                          ■■
DM     120 REM ■■ MSX LASER BATTLE         ■■
CM     130 REM ■■                          ■■
CN     140 REM ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
CO     150 REM
CP     160 REM
EI     170 COLOR1,1,1:SCREEN1,1:VDP(6)=VDP(4):PUTSPRITE1,(20,10),7,77:PUTSPRI
           TE2,(32,22),8,83:PUTSPRITE3,(44,34),9,88:PUTSPRITE4,(68,68),7,80
           :PUTSPRITE5,(80,80),8,82:PUTSPRITE6,(92,92),9,79:PUTSPRITE7,(104
           ,104),10,71:PUTSPRITE8,(116,116),11,82
AO     180 PUTSPRITE9,(128,128),12,65:PUTSPRITE10,(140,140),13,77:FORI=-0TO19
           6:PUTSPRITE11,(200,I),4,98:PUTSPRITE12,(212,I),4,121:NEXTI:FORQ=
           1TO750:NEXT:SCREEN1,1:VDP(6)=VDP(4):PUTSPRITE13,(32,32),10,67:PU
           TSPRITE14,(48,48),9,79
ED     190 PUTSPRITE9,(64,64),12,76:PUTSPRITE16,(80,80),13,67:PUTSPRITE17,(96
           ,96),14,79:PUTSPRITE18,(112,112),10,77:PUTSPRITE21,(120,160),13,
           49:PUTSPRITE22,(132,160),13,57:PUTSPRITE19,(144,160),13,56:PUTSP
           RITE20,(156,160),13,53:FORQ=1TO1234:NEXT
EK     200 KEYOFF:SCREEN1,2:COLOR15,1,1
BH     210 FORI=1TO8:READQ:A$=A$+CHR$(Q):NEXT:SPRITE$(0)=A$:A$=""
AD     220 FORI=1TO32:READQ:A$=A$+CHR$(Q):NEXT:SPRITE$(1)=A$
CB     230 DEFUSR0=60000!:DEFUSR1=60118!:POKE59996!,10:POKE59997!,1:POKE59998
           !,3:POKE59999!,4:DEFUSR2=60220!:FORI=60220!TO60248!:READQ:POKEI,
           Q:NEXT
DE     240 FORI=1088TO1344STEP64:FORJ=0TO7:READQ:VPOKEI+J,Q:NEXT:NEXT
CC     250 PUTSPRITE1,(100,45),7:PUTSPRITE0,(127,100),11:NM=3:ONSTRIG GOSUB41
           0
DL     260 CLS:PRINT" LASER BATTLE":FORI=1TO18:PRINT:NEXT:PRINT"HIT ANY KEY T
           O BEGIN"
HM     270 IFINKEY$=""THEND=USR1(D):FORI=1TO100:NEXT:PLAY"l9m1000s14n33":GOTO
           260
AF     280 CLS:PUTSPRITE1,(100,200)
CG     290 VPOKE8209,255:VPOKE8210,49:VPOKE8211,177:VPOKE8212,97:VPOKE8213,24
           1
HI     300 FORI=6720TO6751:VPOKEI,136:NEXT:FORI=6816TO6847:VPOKEI,23:NEXT:VPO
           KE6787,160:VPOKE6799,160:VPOKE6812,160
AN     310 PG=15:GOSUB1230:PUTSPRITE0,(120,150),15:STRIG(0)ON
CI     320 POKE59999!,8:D=USR0(D):PG=INT(VPEEK(6913)/8)
BK     330 IFPH=2THEN GOSUB1030ELSEIFPH=0THENGOSUB640:GOTO350
CG     340 GOSUB780
BH     350 IFRND(1)<.01ANDPH=1THENPH=2
BI     360 IFPH=0THEN390
AI     370 IFRND(1)<.80RPH=2THEN390
CF     380 GOSUB530
AP     390 IFVPEEK(6912)=200THEN1090
AG     400 GOTO320
DK     410 D=RND(1):STRIG(0)OFF:PLAY"19M1000S14N33"
AG     420 IH=0:I2=0:FORI1=1TO4:IFGS(I1,1)=PGTHENI2=I1
AF     430 NEXT:POKE59991!,152
CA     440 IFI2<>0THENISP=6144+32*GS(I2,2)+PG:IH=2:GOTO480
```

```
EA    450 IM=VPEEK(6917):IP=PG*8
BA    460 IFIP<IM+20RIP-IM>60RVPEEK(6916)=200THENISP=6144+PG:GOTO480
AN    470 IH=1:ISP=6208+PG
EA    480 POKE59992!,ISPMOD256:POKE59993!,ISP\256:I3=6720+PG:POKE59994!,I3MO
          D256:POKE59995!,I3\256:D=USR2(D)
DA    490 'POKE59991!,32:FORI3=1TO20:NEXT:D=USR2(D)
BO    500 IFIH=1THENGOSUB950
BF    510 IFIH=2THENGOSUB1000
CG    520 STRIG(0)ON:RETURN
GJ    530 IFGS(1,1)=0ANDGS(2,1)=0ANDGS(3,1)=0ANDGS(4,1)=0THENRETURNELSESTRIG
          (0)OFF
BA    540 FG=FG+1:IFFG=5THENFG=1
AN    550 IFGS(FG,1)=0THEN540
CH    560 FL=0:K1=0:POKE59991!,152:I=6176+32*GS(FG,2)+GS(FG,1):POKE59992!,IM
          OD256:POKE59993!,I\256
EO    570 IFVPEEK(6720+GS(FG,1))=136THENJ=6752+GS(FG,1):GOTO600
CA    580 J=6816+GS(FG,1):IFGS((FG,1)=PGTHENK1=1
BK    590 K=GS(FG,1):IFK=3ORK=15ORK=28THENFL=1
FE    600 POKE59994!,JMOD256:POKE59995!,J256:D=USR2(D):IFFL=1THENFORJ=1TO10:
          COLOR1,15,15:FORT=1TO30:NEXT:PLAY"139M59000S8N2":COLOR15,1,1:FOR
          T=1TO30:NEXT:NEXT:GOTO1090
KO    610 IFK1=1THENPUTSPRITE0,(100,200):FORK=1TO7:PLAY"164M1000S14N20N21N20
          N21":NEXT
EN    620 PLAY"119M380S10N50":POKE59991!,32:D=USR2(D)
CG    630 STRIG(0)ON:RETURN
BL    640 IFGS(1,1)<>0THEN660
FN    650 I=INT(RND(1)*31+1):IFI=GS(2,1)ORI=GS(3,1)ORI=GS(4,1)THEN650ELSEGS(
          1,1)=I:GHS(1,2)=0:VPOKE6144+I,144:RETURN
BM    660 IFGS(2,1)<>0THEN680
JA    670 I=INT(RND(1)*31+1):IFI=GS(1,1)ORI=GS(3,1)ORI=GS(4,1)THEN670ELSEGS(
          2,1)=I:GS(2,2)=0:VPOKE6144+I,144:RETURN
BC    680 IFGS(3,1)<>0THEN700
JD    690 I=INT(RND(1)*31+1):IFI=GS(1,1)ORI=GS(2,1)ORI=GS(4,1)THEN690ELSEGS(
          3,1)=I:GS(3,2)=0:VPOKE6144+I,144:RETURN
BO    700 IFGS(4,1)<>0THEN720
JE    710 I=INT(RND(1)*31+1):IFI=GS(1,1)ORI=GS(2,1)ORI=GS(3,1)THEN710ELSEGS(
          4,1)=I:GS(4,2)=0:VPOKE6144+I,144:RETURN
AI    720 IFGS(1,3)=1ANDGS(2,3)+1ANDGS(3,3)=1ANDGS(4,3)=1THENPH=1:RETURN
EF    730 J=INT(RND(1)*4+1):IFGS(J,3)=1THEN730
GH    740 VPOKEGS(J,1)+32*GS(J,2)+6144,32:GS(J,2)=GS(J,2)+1
EB    750 IFGS(J,1)<>0THENVPOKEGS(J,1)+32*GS(J,2)+6144,144
JE    760 IFRND(1)<.1ORGS(J,2)>=7THENGS(J,3)=1
CA    770 RETURN
CJ    780 IFVPEEK(6916)<>200THEN800
BL    790 MC=2:PUTSPRITE1,(0,15):RETURN
BO    800 POKE59997!,1:POKE59998!,3:POKE59999!,6:D=USR1(D):IFVPEEK(6917)>251
          THENPUTSPRITE1,(200,200):RETURN
CD    810 MC=INT((VPEEK(6917)+8)/8):IFRND(1)<.85THENRETURN
IJ    820 IFMC<>GS(1,1)ANDMC<>GS(2,1)ANDMC<>GS(3,1)ANDMC<>GS(4,1)THENGOSUB84
          0
CF    830 RETURN
CK    840 PLAY"124M160S8N67"
CB    850 STRIG(0)OFF:FL=0:K1=0:POKE59991!,152:I=6240+MC:POKE59992!,IMOD256:
          POKE59993!,I\256
DD    860 IFVPEEK(MC+6720)=136THENJ=6752+MC:GOTO890
CG    870 J=6816+MC:IFMC=3ORMC=15ORMC=28THENFL=1
BI    880 IFMC=PGTHENK1=1
```

```
BE    890 POKE59994!,JMOD256:POKE59995!,J\256:D=USR2(D)
DI    900 IFFL=1THENFORJ=1TO10:COLOR1,15,15:FORT=1TO30:NEXT:PLAY"139M5900
          N2":COLOR15,1,1:FORT=1TO30:NEXT:NEXT:GOTO1090
CO    910 PLAY"124M160S8N67"
BF    920 IFK1=1THENPUTSPRITE0,(100,200):FORJ=1TO7:PLAYL64M1000S14N20N21N20N
          21":NEXT
AE    930 POKE59991!,32:D=USR2(D)
CE    940 STRIG(0)ON:RETURN
BJ    950 SC=SC+50:GOSUB1230
DB    960 PLAY"164M60000S8N20N21N24N28N40N45N43N29N20N15L3N10"
EC    970 FORI4=1TO60:VPOKE14368+INT(RND(1)*30),INT(RND(1)*255):NEXT:SPRITE$
          (1)=A$
BL    980 PUTSPRITE1,(200,200)
CA    990 RETURN
AF   1000 VPOKE6144+GS(I2,1)+32*GS(I2,2),168:SC=SC+10:GOSUB1230:FORT=1TO30:NE
          XT
DN   1010 VPOKE6144+GS(I2,1)+32*GS(I2,2),32:GS(I2,1)=0
BD   1020 RETURN
DL   1030 IFGS(1,1)<>0ORGS(2,1)<>0ORGS(3,1)<>0ORGS(4,1)<>0THENSTRIG(0)OFF:GOT
          01050
DM   1040 FORJ=1TO4:GS(J,3)=0:NEXT:PH=0:RETURN
BC   1050 J=INT(RND(1)*4+1):IFGS(J,1)=0THEN1050ELSESTRIG(0)ON
EO   1060 VPOKE6144+GS(J,1)+32*GS(J,2),32:IFGS(J,2)>0THENGS(J,2)=GS(J,2)-1
LA   1070 VPOKE6144+GS(J,1)+32*GS(J,2),144:IFGS(J,2)=0THENVPOKE6144+GS(J,1),3
          2:GS(J,1)=0
BJ   1080 RETURN
DF   1090 IFVPEEK(6912)<>200THEN1170
HC   1100 STRIG(0)OFF:FORT=1TO1500:NEXT
CL   1110 CLS:PUTSPRITE0,(100,20):PUTSPRITE1,(100,200):PRINT"          LASER BA
          TTLE":PRINT:PRINT:PRINT
BM   1120 NM=NM-1:IFNM=0THEN1200
DJ   1130 PRINT"      MEN LEFT: ";NM
DO   1140 FORJ=1TO2500:NEXT
BA   1150 FORJ=1TO4:GS(J,1)=0:NEXT:PH=0
CN   1160 GOTO280
HJ   1170 STRIG(0)OFF:FORT=1TO1500:NEXT
EM   1180 CLS:PUTSPRITE0,(100,200):PUTSPRITE1,(100,200):PRINT"          LASER
          BATTLE":PRINT:PRINT:PRINT
GC   1190 PRINT"YOUR POWER PLANT HAS BEEN ":PRINT"DESTROYED":PRINT:PRINT:GOTO
          1120
HC   1200 PRINT:PRINT:PRINT
BI   1210 PRINT"      YOUR SCORE WAS ";SC
GG   1220 IFINKEY$=""THENENDESLE620
AE   1230 FORI=1TO23:PRINT:NEXT:PRINT"SCORE:";SC;CHR$(11);
BD   1240 RETURN
DM   1250 DATA 24,60,24,60,24,52,122,255
KE   1260 DATA 15,16,32,32,9,198,203,254,255,207,199,9,33,48,16,15,240,8,4,4,'
          198,99,,211,127,255,243,227,198,132,12,8,240
AI   1270 DATA 42,88,234,58,87,234,79,205,44,235,17,33,0,25,237,91,90,234,124
          ,186,194,67,235,125,187,194,67,235,201
FK   1280 DATA 90,165,90,165,90,165,66,129
DK   1290 DATA 24,126,213,171,255,66,60,24
EA   1300 DATA 24,24,24,24,24,24,24,24
AL   1310 DATA 24,36,90,165,219,165,219,255
5B   1320 DATA 2,144,4,17,64,4,161,8
END
```

MACHINE-CODE SUPPORTER. (for MSX)

by : Andrew Lacey.

This program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84) or the Year Book.

```
BJ      100 CLS
LC      110 PRINT" MACHINE  * * * * * * * * *
PL      120 PRINT"* * *  CODE * * * * * * * *"
CE      130 PRINT"* * * * * SUPPORT * * * * *"
DC      140 PRINT"* * * * * * * * * PROGRAM *"
AJ      150 FOR I = 60000!TO 60217!:READQ:POKEI,Q:NEXT
BD      160 DATA 219,170,230,240,33,92,234,94,246,8,0,0,0,0,0,0,0,0,0,0,0,0,211,
            170,219,169,71,0,0,0,0,0,0,0,0
DA      170 DATA 0,0,0,230,128,194,151,234,203,91,202,151,234,62,3,50,94,234,1
            95,209,234,120,230,64,194,170,234,203,83,202,170,234,62,2
EF      180 DATA 50,94,234,195,209,234,120,230,32,194,189,234,203,67,202,189,2
            34,62,0,50,94,234,195,209,234,120,230,16,194,208,234,203,75,202,
            208,234,62,1,50,94,234,195,209,234,201,62,0,50,93,234,58
BA      190 DATA 93,234,203,39,203,39,79,6,0,33,0,27,9,58,95,234,71,58,94,234,
            87,254,0,194,250,234,205,27,235,121,152,79,195,44,235,254
AH      200 DATA 1,194,6,235,35,205,27,235,195,244,234,254,2,194,20,235,205,27
            ,235,121,128,79,195,44,235,35,205,27,235,195,14,235,125,243,211
DA      210 DATA 153,124,211,153,0,0,0,0,219,152,79,183,251,201,125,243,211,15
            3,124,246,64,211,153,121,211,152,251,201
END
```

INSTRUCTIONS TO MSX USERS

Enter the Machine Code Supporter Program first and when sure it is bug-free save it to tape. (note that you can have these MSX programs on disk and load and save them normally, but they will not run once you are in disk-BASIC mode....they must be run from Cassette BASIC mode only). When sure that the support program is working type in and save the main program on cassette immediately following the support program. To use, you simply load and run the support program and then load and run the main program. Joysticks are not supported by the machine-code support program but keyboard cursor keys are. The same machine-code support program is used for controlling many more MSX games so keep it handy. We intend to publish further MSX games by Andrew Lacey.

SPEEDY

by : Author unknown

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV. 84.) or The Year Book.

```
FJ      90 REM THIS IS - SPEEDY -
ML      91 REM INSTRUCTIONS ARE IN THE BODY OF THE PROGRAM
DP      92 REM SEE HOW YOU MATCH UP AGAINST -999-
DE     100 CLS: LOCATE 18,8: PRINT"PRESS
                                        'I' FOR INFORMATION
                                              'S' TO START AT ONCE

          "
BM     110 A$ = INKEY$: IF A$ = "" THEN 110
GN     120 IF A$ = "I" THEN GOSUB 1000
CN     130 LO = 999
BF     140 X = 240: Y = 170: NO = 200: SC = 0
DC     150 COLOR 13,1,5: SCREEN 1
BG     160 LOCATE 0,30
DH     170 RN = INT(RND(-TIME)*5)+4: FOR T = 1 TO RN: PRINT" ";: NEXT: SC =
          SC+RN
LE     180 RN = INT(RND(-TIME)*4)+2: FOR T = 1 TO RN: PRINT";: NEXT: SC =
          SC+RN
CG     190 IF SC < 680 THEN 170
CJ     200 RESTORE 700: FOR T = 1 TO 8: READ A: Z$ = Z$ + CHR$(A): NEXT: SP
          RITE$(0) = Z$
AN     210 CIRCLE(16,16),10,9,,,1.3: PAINT(16,16),9: PUT SPRITE 0,(X,Y),3
DG     220 LOCATE 100,184: COLOR 5: PRINT"BEST SCORE"LO
BK     230 A$ = INKEY$: IF A$ = "" THEN 230
CB     240 IF A$ <> "R" THEN 250 ELSE COLOR,14: SCREEN 0: FOR T = 1 TO 500:
          NEXT: GOTO 140
BD     250 H1 = INT(TIME/60)
AK     260 D = STICK(0)
DC     270 PUT SPRITE 0,(X,Y),3
CL     280 X = X + (D = 7)*2 - (D = 3)*2
DC     290 Y = Y + (D = 1)*2 - (D = 5)*2
DC     300 IF X < 2 OR X > 250 THEN X = 240: Y = 170
CO     310 IF Y < 2 OR Y > 189 THEN X = 240: Y = 170
DO     320 IF POINT(X+3,Y+3) <> 1 AND POINT (X+3,Y+3) <> 5 THEN 400
AN     330 NO = NO - 40: IF NO < 60 THEN NO = 200
AB     340 SOUND 0,NO: SOUND 8,6: GOTO 260
FH     400 IF POINT(X+3,Y+3) = 13 THEN X = 240: Y = 170: GOTO 330
CN     410 SCREEN,0: SCREEN 0
BH     420 FOR Z = 1 TO 3::FOR X = 220 TO 0 STEP-10: FOR W = 1 TO 2
GI     430 COLOR,4*Z: SOUND 0,X: SOUND 8,6+3*Z
EO     440 NEXT W,X,Z: SOUND 8,0: COLOR15,8
AH     450 H2 = INT(TIME/60): PRINT"YOUR SCORE IS:"(H2-H1)
GJ     460 IF LO <= (H2-H1) THEN PRINT: PRINT"THE BEST SCORE IS" LO
FA     470 IF LO > (H2-H1) THEN PRINT: PRINT"YOUR SCORE IS THE LOWEST": LO
          = (H2-H1)                                                    CE
KJ     480 PRINT:PRINT"ANOTHER GAME (type YES/NO and press
          NTER])"
AO     490 INPUT Q$
BI     500 IF Q$ = "no" OR Q$ = "NO" THEN PRINTTAB(15)"GOODBYE": LOCATE ,,1
          : END
NE     510 IF Q$ <> "no" OR Q$ <> "NO" THEN CLS: COLOR 13,1: LOCATE 15,10:
          PRINT"stand by": FOR T = 1 TO 500: NEXT: GOTO 140
```

```
CE    700  DATA 0,68,0,16,0,68,0,0
AA   1000 COLOR,15,5: SCREEN 1
GH   1010 LOCATE 30,90: COLOR 13: PRINT"ARE YOU READY FOR ACTION WITH"
DH   1020 FOR T = 1 TO 1000: NEXT
FO   1030 COLOR ,1,2: SCREEN 2
AK   1040 LOCATE 50,90: COLOR 9: PRINT"SPEEDY"
DK   1050 FOR T = 1 TO 1000: NEXT
GH   1060  COLOR 15,1,5: SCREEN 1
DA   1070 LOCATE10,8:  PRINT"USING THE CURSOR CONTROLS OR JOYSTICK"
GC   1071 LOCATE 10,16: PRINT"MOVE THE PLAYER, THE ";: COLOR 3: PRINT"GREE
     N ";: COLOR 15: PRINT"DOTS IN THE"
GF   1072 LOCATE 10,24: PRINT"BOTTOM RIGHT HAND CORNER OF THE SCREEN"
AI   1080 LOCATE 10,32: PRINT"ALONG THE black PATH THROUGH THE MAZE TO"
BK   1081 LOCATE 10,40: PRINT"THE ";: COLOR 9: PRINT"RED ";:COLOR 15: PRIN
     T"CIRCLE IN THE TOP LEFT CORNER."
IC   1090 LOCATE20,56: COLOR 10: PRINT"THE CENTRE DOT IS THE IMPORTANT ONE
     ."
FL   1091 LOCATE 10,64: PRINT"IF IT HITS A ";:COLOR 13: PRINT"WALL ";: COL
     OR 10: PRINT"OR GOES TOO CLOSE TO "
FH   1092 LOCATE 20,72: PRINT"THE SCREEN EDGE, BACK TO THE START."
AI   1100 LOCATE 50,96: COLOR 7: PRINT"TRY FOR THE LOWEST SCORE"
FA   1110 LOCATE 6,112: COLOR 15: PRINT"press any key when you are ready t
     o start"
FM   1111 LOCATE 10,128: PRINT"IF THE MAZE IS IMPOSSIBLE, PRESS 'R'"
CN   1112 LOCATE 40,136: PRINT"BEFORE YOU MOVE THE PLAYER."
EE   1120 A$ = INKEY$: IF A$ = "" THEN 1120
BD   1130 RETURN
END
```

How to play SPEEDY

Once you are sure you have the program typed in correctly save it to tape or disk then type RUN. A MAZE will be created on your screen and down in the bottom right-hand corner you will see a shape made of five dots. This is your playing piece, and I will tell you now that the center dot is the only one of any significance. Any of the other four dots may be allowed to touch any other part of the maze but keep the center dot well away. Object of the game is to negotiate through the maze, in the quickest possible time and touch the glowing ball in the top left corner of the screen. This just might prove slightly harder than it sounds. One point to remember is that if the game produces a maze which is too hard or is in fact impossible you can get it improved by typing R before striking any other key. Have fun.

REVIEW OF MSX CARTRIDGE SOFTWARE.

by J. Collins.

"ALI BABA AND THE FORTY THIEVES" CATCH THE "CRAZY TRAIN"

While last in Launceston I visited the computer section of our local Myer Store and browsed through the range of MSX software. Two titles particularly caught my eye and they are now being destruction-tested by my son, his friends, my wife, and myself. First to be looked at is ALI BABA & THE FORTY THIEVES. This is a Dodgem and Catchem type game. There's a maze type screen, a horrible green Head Thief called DON, up to forty of his subordinate thieves, some Magic Dust, and of course there's yourself, or should I say Ali-Baba. The scene is that you have some treasure, the thieves want it and will steal it unless you can prevent them, and DON is there to help them by chasing very accurately after Ali-Baba. You have some magic dust which you can sprinkle in Dons' path and this is very effective although its' effect is very short lived. Supply of this dust appears to be unlimited. After a certain time limit an area of the maze becomes a mystery zone and if you can direct Ali-Baba through the right part of this zone he can run at twice his normal speed which is very handy. Running through another part of the mystery zone can turn Ali-Baba into a giant with the power to dong Don, but run through the wrong part and Don becomes the giant from whom Ali-Baba must flee. Final part of the mystery zone will cause the thieves vault to open and Ali-Baba can enter and recover his stolen treasure. Action continues on each screen in much the same way except that you need to eliminate 10 thieves, then 20, then 30, then 40 in the final level. As expected, 10 thieves is relatively easy, 20 can give you tendonitis of the wrist joint, 30 goes close to breaking the joystick, and I've only got the word of the cartridge manufacturer that catching 40 is even possible ! The movement of all characters is very fast, response to the joysticks is excellent, and the keyboard cursor keys can be used as well. The background music and sound effects are well done. A simple game theme but well done and most enjoyable. This is a SONY HIT-BIT game in Cartridge format only and is excellent value at around $25.00.

The other game I purchased is the SONY HIT-BIT cartridge version of CRAZY TRAIN by Konami. Like all Konami games this one is adapted from the original Arcade Version and is excellent. What you get first up is a small screen with various seemingly unconnected railway lines. The screen is made up of different colored squares, each with a portion of the tracks. On your screen there is one black or blank square and you can move other portions of the screen around to fill up this square and by so doing you create new and different patterns of track. All the while your engine is puffing merrily along the tracks you create. Object is to pass as many of the Stations as you can in the quickest time. If you take too long the whole screen turns nasty black and you are driving your engine in the dark, which is not easy. On the first screen there are six stations. There are also Black Ghost Trains and these must be avoided at all costs. Collision costs you a life. Also fatal is leaving sections of track

unfinished by not watching where your blank square is. A train will run off the end of a block quite happily with dire results. Passing all the stations moves you on to the second level and here the rail system is larger, you still have the blank section, there are more stations and of course there are more ghost trains to be avoided.

These can be shunted to continuous loops of track if you are quick but watching a speeding train while dodging ghost engines and trying to work two or three squares ahead is quite tricky and demands a lot of concentration. The third and fourth screens introduce clever touches like more of everything, lines that end in buffers in the middle of nowhere and of course more ghost trains. I really like this game. The concept is very good and very well executed. Music is excellent and sound effects are true to life. If you are a real masochist you can hold down the trigger and make everything happen at double speed. Cost of Crazy Train is around $25.00.

Both of these games are highly recommended.

CP/M GAMES DISK OFFER

As mentioned in the introduction we have some games to offer in the CP/M Format. They certainly aren't full of mind-boggling graphics nor do they have-sound effects or bells and whistles. They are handy for whiling away some spare time and they just might save your sanity one dark night when the program you are working on wont and the whole box of bugs appears to have taken up residence in your 328. There's chess and checkers and a maze game and more. Send in a formatted CP/M disk with your system on it and include $6.00 for post and handling or send $12.00 if you want us to supply the disk as well. Also available for MSX CP/M systems. Please be sure to state which computer you want them for.

Software / Article Competition.

This competition closes as from last post on 30th November and any entries received after this time will not be eligible for judging and wont win any prizes. Full details of the lucky winners will be given in the December issue of the newsletter. My best wishes to all and good luck.

GROUP AUTHOR SOFTWARE-LIBRARY PROGRAM LIST

ASKING PRICE Includes Pack and Post.		OUR MEDIA CASSETTE	OUR MEDIA DISK	YOUR MEDIA CASSETTE	YOUR MEDIA DISK
3D-MAZE $5.00		9.00	11.00	5.00	5.00
CALENDARS $3.00		7.00	9.00	3.00	3.00
MURDER $10.00	##	14.00	16.00	10.00	10.00
MYSTERIOUS MANOR $5.00		9.00	11.00	5.00	5.00
COUNT DRACULAR $10.00		14.00	16.00	10.00	10.00
CRUNCH $10.00	##	14.00	16.00	10.00	10.00
DISASSEMBLER $5.00	##	9.00	11.00	5.00	5.00
ELIZA $10.00	##	14.00	16.00	10.00	10.00
MARVYN $10.00	##	14.00	16.00	10.00	10.00
MIGHTY MORMAR $5.00		9.00	11.00	5.00	5.00
HOUSE OF FRANKENSTEIN $5.00		9.00	11.00	5.00	5.00
PACMAN $10.00		14.00	16.00	10.00	10.00
SUPER IMP/ED $10.00		14.00	16.00	10.00	10.00
JOYSTICK SPRITE $10.00	##	14.00	16.00	10.00	10.00
FILES $5.00		9.00	11.00	5.00	5.00
RUBIKS CUBE $10.00		14.00	16.00	10.00	10.00
X'BERT $10.00		14.00	16.00	10.00	10.00
FIVE GAME PACK $6.00		10.00	12.00	6.00	6.00
ASMED/LOADER $11.00		15.00	-----	11.00	-----
WP318/WP328 $5.00		9.00	11.00	5.00	5.00
MSX GAMES PACKAGE $7.50		11.50	13.50	7.50	7.50
BASIC UTILITIES $6.50		-----	12.50	-----	6.50
DRAW-2 $7.50		11.50	13.50	7.50	7.50
SVI ARTIST $7.50		11.50	13.50	7.50	7.50

The programs shown with two hash marks after the name are available for both SVI318/328 computers and also have been ported across to MSX.

Please be sure when ordering that you specify which computer you want the software for.

CP/M PUBLIC DOMAIN SOFTWARE

UNERA.COM		$10.00
ADHEX.COM		$10.00
DISK.COM		$10.00
ADVENT.COM	*	$ 5.00
MODEM7.COM	*	$ 5.00 (Also available for MSX MACHINES)
YAM.COM	*	$ 5.00
XDIR.COM		$ 5.00
ZCPR2		$ 5.00
CATALOG.COM		$ 5.00
CP/M GAMES DISK		$ 6.00

All CP/M software is suitable for DISK use only and the prices shown do not include cost of media.

If you don't supply disks please add normal $6.00 media charge to the price shown against the item you want.

If you are supplying disks please FORMAT each disk.

Those CP/M items with an asterisk after the program name fill a normal SS/DD disk and other items can not be included on the disk. Example.....if you want MODEM7 and XDIR.COM you will need two disks.