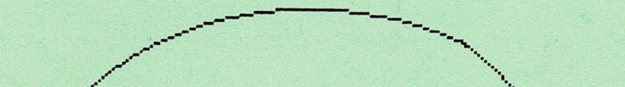
SEGAMAG
NEW SOUTH WALES

MAY 1989



Registered by Australia Post
Publication No. NBG8403 x

$4.00

OFFICERS:

PRESIDENT	Scott MacDonald	(046)668-956
VICE-PRESIDENT	John Carter	750-3271
SECRETARY	Allan Rodd	816-1618
TREASURER	Brian Minett	825-3998
ASST. TREAS.	Ted Hartley	622-2416
EDITOR	Rex Chandler	872-4256
ASST. EDITORS	Trevor Williams and Fady Sarkis	
LIBRARIAN	Brian Minett	
ASST LIB.	Jenni Williams	

HELP DESK: Scott MacDonald (all hours)
John Carter(7-9pm)

MEETINGS: SECOND SUNDAY EACH MONTH EXCEPT JANUARY

NEXT MEETING: SUNDAY, MAY 14, 1989

SERVICING: SCOTT MacDonald

HARDWARE AND SOFTWARE SALES: SCOTT MacDonald and John Carter

SEGAMAG IS PUBLISHED IN MARCH, MAY, JULY, SEPTEMBER AND NOVEMBER EACH YEAR.

AGENDA FOR MAY MEETING

DATE: Sunday, May 14, 1989
TIME: 10:00am to 4:00pm

PLACE: Gladesville Public School
Victoria Road opposite post office

10:00-11:30 Games, try some of the new ones-they are grueling. Also see Scott's new 6mHz Sega almost double the standard Sega clock speed. If it arrives in time Scott will show the new Sega Mega Drive 16 bit computer.

11:30(approx) Demonstration by David Tester of new disc drive set-up

4:00 Meeting closes.

COMPLIMENTARY TEA, COFFEE OR CORDIAL SERVED AT ALL SEGA GROUP MEETINGS.

REVIEW OF APRIL'S MEETING.

Scott MacDonald addressed the meeting on upcoming and potential hardware developments. Extended memory to 64k of RAM is available now, with 256k available imminently, and 1meg possible in the near future. In the ROM Basic it is hoped to recover or use the 32K that the ROM occupies. If that happens, then the programming possibilities of even 64,000 bytes free are immense.

Other topics were also raised. A real-time clock with battery back-up can be installed. This assists with various areas of programming including setting up of variables for up tp four disc-drives. Digital/analog and analog/digital convertors are available. A fruitful area of investigation for an electronically inclined member is the adaption of the VRAM chip of the Master System to the Sega. This would give us high resolution graphics with no bleeding and potentially an 80-column text screen. We should also then be able to play Master cartridges on our Segas. Hopefully also, Sega's new machine - the MEGA DRIVE, will be available by the end of the year. This is a 16 bit and downwardly compatible with 80 columns, high definition, and 256,000 colours. This sounds more than a little interesting.

David Tester was unfortunately unable to make it down from Bathurst. His topic on 'How to Build Your Own Disc Drive' was introduced by Scott however. The concept is to attach a drive directly to a keyboard through a currently unused I/O port (there are four of these available so up to four drives are possible). The necessary BOARD costs $9, one CHIP costing $40 is needed, and you supply the DRIVE of your choice ($80-$125 depending on type). Some extra minor parts which are needed are available. This all sounds too good to be true. We look forward to having David demonstrate his success for us.

On a different note, nominations for offices were taken in front of about 20 members as follows:-

President	Scott MacDonald
Vice-president	John Carter
Secretary	Allan Rodd
Treasurer	Brian Minett
Assistant Treas.	Ted Hartley
Editor	Rex Chandler
Assistant Ed's	Trevor Williams and Fady Sarkis
Librarian	Brian Minett
Assistant Lib	Jenny Williams

As these were the only nominations (& the only ones likely), the positions were declared filled by agreement of the members present. The Librarian positions were strengthened to look after the hiring of our existing computers, disc drives, modems etc, and to develop a library of original software for hiring. Allan's role in assembling and publishing the SEGAMAG is noted with appreciation. We wish the incumbents in these positions ongoing succes and hope that the members give every support, without which the group will not function properly.

REX CHANDLER, EDITOR.

```
1 X=RND(-1):REM BANK ROBBERY By Jonathon Kirkwood.
2 PATTERNS#0,"0040201020000000"
3 PATTERNS#1,"0000000000000000"
4 PATTERNS#2,"0000000000000000"
5 PATTERNS#3,"0000000000000000"
6 PATTERNS#4,"03040404130F0303"
7 PATTERNS#5,"030302040804020E"
8 PATTERNS#6,"C0202020C0E0D0D0"
9 PATTERNS#7,"D0C0402010204070"
10 PATTERNS#8,"0103073F00000000"
11 PATTERNS#9,"0000000000000000"
12 PATTERNS#10,"80C0E0FC00000000"
13 PATTERNS#11,"0000102804000000"
14 PATTERNS#12,"0000000005070303"
15 PATTERNS#13,"0707070000000000"
16 PATTERNS#14,"00000000A0E0C0C0"
17 PATTERNS#15,"E0E0E00000000000"
18 PATTERNS#16,"000000001517130B":PATTERNS#17,"0703030000000000"
19 PATTERNS#18,"00000000A8E8C8D0"
20 PATTERNS#19,"E0C0C00000000000"
21 PATTERNS#20,"0103073F00000000"
22 PATTERNS#21,"0000000000000000"
23 PATTERNS#22,"80C0E0FC00000000"
24 PATTERNS#23,"0000000000000000"
25 PATTERNS#24,"0304040403070B0B"
26 PATTERNS#25,"0B0302040804020E"
27 PATTERNS#26,"C0202020C0E0D0D0"
28 PATTERNS#27,"D0C0402010204070"
29 PATTERNS#28,"0000000000000000"
30 PATTERNS#29,"91A8C70303C7A891"
31 PATTERNS#30,"0000000000000000"
32 PATTERNS#31,"C02EF1F1F1F12EC0"
33 SCREEN2:COLOR1,15,(0,0)-(255,191),15:CLS:PRINTCHR$(16);:SCREEN,2:MAG3:SA=0
34 GOTO38
35 IFSA=0THENPRINTCHR$(16);:LW=LEN(M$)*6
36 IFSA=1THENPRINTCHR$(17);:LW=LEN(M$)*12
37 COLORC:CS=127-(LW/2):CURSORCS,BY:PRINTM$:RETURN
38 COLOR4:PRINT CHR$(17)
39 CURSOR50,50:PRINT"B R"
40 CURSOR50,60:PRINT"A O"
41 CURSOR50,70:PRINT"N B"
42 CURSOR50,80:PRINT"K B"
43 CURSOR50,90:PRINT"  E"
44 CURSOR50,100:PRINT"  R"
```

A FUN GAME
WHERE YOU
TEST YOUR
REFLEXES
AGAINST THE
COMPUTER.

```
45 CURSOR50,110:PRINT"  Y"
46 CURSOR100,60:COLOR13:PRINTCHR$(16);"Written by":BEEP:CURSOR100,70:PRINT"Jonat
han Kirkwood":BEEP
47 COLOR1:PRINTCHR$(17);:CURSOR100,100:PRINT"SEGA ";CHR$(16);:COLOR1:PRINT"SC-30
00":COLOR1,14,(0,45)-(255,122),15
48 M$="<*> PRESS SPACEBAR TO PLAY <*>":BY=150:C=6:SA=0:GOSUB35
49 IFINKEY$<>" "THEN49
50 BEEP:CLS:COLOR1,15,(0,0)-(255,191),15
51 SCREEN1,1:CONSOLE0,24:CLS:COLOR1,11
52 PRINT"To play BANK ROBBERY"
53 PRINT
54 PRINT"  When a curtain is drawn open and a"
55 PRINT"man appears,study the man closely"
56 PRINT"before firing."
57 PRINT"  If he has his hands up,do not shoot"
58 PRINT"or else your score will decrease."
59 PRINT"If he has his gun drawn you must pressD to draw then F to fire before h
e hasa chance to fire."
60 PRINT"  When you have shot 10 of your alliesthe game is over.":PRINT
61 SC=0:AS=0:SCREEN2,1:CLS
62 LINE(0,191)-(255,191),1
63 LINE(0,191)-(79,38),14,B
64 LINE(5,43)-(37,75),14,B
65 LINE(42,43)-(74,75),14,B
66 LINE(5,80)-(37,112),14,B
67 LINE(42,80)-(74,112),14,B
68 LINE(5,117)-(37,149),14,B
69 LINE(42,117)-(74,149),14,B
70 LINE(5,154)-(37,186),14,B
71 LINE(42,154)-(74,186),14,B
72 PAINT(1,39),14
73 COLOR1,4,(5,43)-(37,75),15
74 COLOR1,4,(42,43)-(74,75),15
75 COLOR1,4,(5,80)-(37,112),15
76 COLOR1,4,(42,80)-(74,112),15
77 COLOR1,4,(5,117)-(37,149),15
78 COLOR1,4,(42,117)-(74,149),15
79 COLOR1,4,(5,154)-(37,186),15
80 COLOR1,4,(42,154)-(74,186),15
81 SCREEN1,1:CURSOR0,20:PRINT"Press spacebar":BEEP
82 IFINKEY$<>" "THEN82
83 BEEP:SCREEN2,2
84 IFAS=10THENGOTO123
```

```
85 CURSOR50,0:PRINTCHR$(16);"SCORE : ";SC
86 FORJ=0TO500+INT(RND(1)*401):NEXT
87 REMCHOOSE WINDOW
88 WS=0:X=0:Y=0:YS=24
89 WS=INT(RND(1)*8)+1
90 IFWS=1THENX=5:Y=43
91 IFWS=2THENX=42:Y=43
92 IFWS=3THENX=5:Y=80
93 IFWS=4THENX=42:Y=80
94 IFWS=5THENX=5:Y=117
95 IFWS=6THENX=42:Y=117
96 IFWS=7THENX=5:Y=154
97 IFWS=8THENX=42:Y=154
98 SPRITE0,(200,158),24,4
99 REMCHOOSE GOOD & BAD
100 D$="":R=RND(1)
101 IFR>.5THEND$="GOOD"
102 IFR<=.5THEND$="BAD"
103 E1=0:E2=0
104 IFD$="GOOD"THENE1=16:E2=20:GOTO106
105 E1=12:E2=8
106 COLOR1,15,(X,Y)-(X+32,Y+32),15
107 SPRITE1,(X,Y),E1,4:SPRITE2,(X,Y),E2,1
108 D1=0:F1=0:TG=35
109 FORI=1TOTG
110 JK$=INKEY$
111 IFJK$=""THEN115
112 IFJK$="D"THEND1=1:SPRITE0,(200,158),4,4:SPRITE5,(200,158),0,1:GOTO115
113 IFJK$="F"ANDD1=0THEN115
114 IFJK$="F"ANDD1=1THENSOUND4,2,15:FORA=0TO20:NEXTA:SOUND0:LINE(205,163)-(X+16,
Y+16),1:GOSUB125
115 NEXT I
116 IFD$="GOOD"THENSC=SC+100:CLS:GOTO62
117 SOUND4,2,15:FORA=0TO20:NEXTA:SOUND0
118 SPRITE5,(255,191),0,15
119 SPRITE0,(200,158),28,4
120 FORA=0TO1000:NEXTA:GOTO123
121 IFD$="GOOD"THENAS=AS+1:SC=SC-100:CLS:GOTO62
122 SC=SC+100:CLS:GOTO62
123 CLS:COLOR1,15,(0,0)-(255,191),15
124 M$="GAME OVER":BY=91:SA=1:C=1:GOSUB35:FORA=0TO1000:NEXTA:GOTO50
125 IFD$="GOOD"THEN121
126 GOTO122
```

JOY-STICK MODIFICATION.

To allow twin fire buttons to be used.
First obtain a new cable assembly-
TANDY CAT NO.270-1705 for around $8.
Cut off end with male pins, (if you're not sure, plug the socket end into the computer and cut off the other end).
Pin Designation looking at front of socket :-

```
  5 4 3 2 1
--------------
-. . . . . . -
 - . . . . .-
  --------
   9 8 7 6
```

Wire functions & color :-

1 Brown Up
2 Red Down
3 Orange Left
4 Yellow Right
5 Green Not used
6 Blue Left Fire
7 Gray Not used
8 White Common
9 Black Right fire

Separate joy stick carefully noting how springs & buttons go so it can be reassembled.
The circuit board inside will have one track going to all switches, this is the common track.
The top & side fire buttons will also be connected to the common track as well as being joined after the buttons. It is here that you must break the track. A sharp knife will do but only break enough track to isolate the join.
Next step is to remove the old wires. These may not be color coded in the same way as the new ones, but dont worry.
These wires could be pushed on or soldered.
Now, to solder on the new wires use only a small iron.
The important wire is the common wire.
Place a small amount of solder on both the wire & the track to be joined, & solder.
Repeat with all remaining wires except the two not used.
How do I know which one? Just follow the tracks back from the l/r/u/d, contacts & solder on.
Re-assemble joystick & try. If it goes left instead of right, swap the orange & yellow wires and so on till all works correctly.
Any troubles, bring cable & joysticks to a club meeting for help. If you dont have the tools to do the job, ring me on 825 3998 before a meeting and I will bring everything needed. Just supply joysticks & cable assembly.

Brian Minett (02) 825 3998.

MACHINE CODE - REFERENCE.

This reference section describes the addresses in the basic interpreter where the Machine Code routines to execute specific tasks are located. These tasks may then be done by simply CALLing the specified address from your Machine Code program once the relevant preconditions (such as setting variables in the registers) has been completed. The addresses are in hexadecimal and are for the Disc Drive, with the equivalent addresses for Level 3 Basic in brackets. I've left out a few call addresses which are for advanced programmers only because they would only add to confusion (yours and mine). I've also not attempted to deal with Floating Point operations, Arithmetic and Data Processing routines, Command and Statement Processing routines, Time Handling routines and Disc and I/O. These are best left for more expert discussion or they (and the ones below) can be perused in the document on 'Basic Interpreter Routines' available from the Library.

0069(2BC5) Write address, set in VRAM. HL=VRAM address.

006C(2BC8) Data write to VRAM. A=data.

006F(2BCB) Read address, set in VRAM. HL=VRAM address.

0072(2BCE) Data read out of VRAM. A=data read out.

0075(2BC2) VDP status register read. A=contents of status register. Check bit 5 for sprite collissions.

0081(39DC) Convert X-Y co-ordinates into VRAM address and bit position. D=Y(0-191) E=X(0-255). HL=Pattern generator table address in VRAM (graphics mode). When color table address is required, add &H2000 to this address. A=bit position (0 to 7).

0084(39E5) Text screen display.

0087(39E2) Graphics screen display.

008A(39D0) Text screen, back color set. A=color(0 to 15)

008D(39D3) Text screen, character color set. A-color(0 to 15).

0090(39D6) Graphics screen, "0" (back) color set. A=color(0 to 15).

0093(39D9) Graphics screen, "1" (character, dots & lines) color set. A=color(0 to 15).

0096(39F4) Text screen scroll. H=upper & L=lower limit of block to be scrolled. ie H=2 & L=15 means that lines 2 through 15 are scrolled by one line).

0099(39EE) Dot plot in graphics screen. D=Y(0-191) E=X(0-255) A=0(erase) or 1(plot).

009C(39F1) Line draw/erase in graphics screen. D=Start Y E=Start X; H=End Y L=End X; A=0(erase) <>0(draw); B=0(dot connection) <>0(side or overlapping connection).

009F(39FA) Character pattern read. A=Character code. Look in [PATERN]8 ie 8 bytes starting from ABE4(9413), for output data.
00A2(39F7) Character pattern write (into Pattern Generator Table in VRAM). A=character code, Have your pattern in [PATERN]8 ie 8 bytes starting from ABE4(9413).
00A5(428F) Sprite attribute table write. A=Plane no(0 to 31). Have your data in [PATERN]4 ie 4 bytes starting from ABE4(9413); #1=Y, #2=X, #3=sprite name no, #4=EC & color. EC is Early Clock ie set left nibble to 8 for smooth scrolling off-screen left and always set to 0 when scrolling off-screen right. The right nibble is the color.
00A8(8428C) Sprite attribute table read. A=Plane no. Output data is found in [PATERN]4 as above.
00AB(4289) Sprite pattern write. A=sprite name(0 to 255), Have your pattern in [PATERN]8 ie 8 bytes starting from ABE4(9413).
00AE(4298) Sprite color set. A=plane no(0 to 31), C=color code(0 to 15).
00B1(4295) Sprite name set. A=plane no(0 to 31), C=color code(0 to 15).
00B4(4292) Sprite position set. A=plane no(0 to 31), D=Y, E=X.
00B7(429B) Sprite early clock bit set. A=plane no(0 to 31), C=1(Set EC), C=0(Reset EC).
00BA(42CB) Text screen clear.
00BD(42C8) Text screen cursor home.
00C0(42CE) Text screen upper/lower limit set. H=upper & L=lower limit. This is part of the CONSOLE function in Basic.
00C3(4A12) Cursor move. A=moving direction (0:Right, 1:Left, 2:Up, 3:down).
00C6(4A15) Text screen line feed.
00C9(4A0F)Text screen cursor return`to line head.
00CC(4A0C) Text screen line clear.
00CF(4A09) Text screen delete.
00D2(4A06) Text screen (character) insert.
00D8(4A1B) Text screen skip to next tab positiuon.
00DB(4A1E) Text screen cursor display. [CURDAT]1 is code of character under cursor, (I think its AECD for disc drive).
00DE(4A21) Text screen cursor erase. [CURDAT]1 is code of character under cursor. (I think its AECD for disc drive.)
00E1(4A24) Text screen, one character display. A=Charactrer code.
00E7(42C5) Text screen, VRAM address compute. D= Vertical & E= horizontal position of address to be computed. HL= corresponding VRAM address.
00EA(42D7) Text screen, one line read. DE= Start address of memory to store read data in.
00ED(4A27) Graphics screen, 8-bit plot. D=Y & E=X, A=pattern.

00F0(4A2A) Graphics screeen, 8-byte plot. D-X & E-Y, [PATTERN]8 - pattern, ie 8 byes starting from ABE4(9413).
00F3(4A2D) Graphics screen character plot. D=Y & E=X, A=character code.
00F6(4A30) Graphics screen, character size set. A=0 for standard & A=1 for double size.
00F9(4A33) Graphics screen, cursor position read. D=Y & E=X.
00FC(4A36) Graphics screen, cursor position set. D=Y & E=X.
00FF(4A39) Graphics screen, cursor home.
0102(4A3C) Graphics screen, cursor move. A= 0:right, 1:left, 2:up, 3:down.
0105(4A3F) Graphics screen, back-color set (for a specified area). A= Color code, D=Y & E=X of upper-left corner, H=Y & L=X of lower-right corner.
0108(4A42) Graphics screen clear.
010B(4A45) Graphics screen tab skip (to next tab position).
010E(4A4B) & 4E72(4A96) Graphics screen block erase & block paint resp. D=Y & E=X of upper left corner, H=Y & L=X of lower right corner.
0111(4a4E) Graphics screen back drop color set. A= color code.
0114(4A51) Graphics screen point read. D=Y, E=X. A=0 if off and A=1 if on.
0117(4A54) Graphics screen delete (character under cursor).
011A(4A57) Graphics screen one character display (at cursor position). A= character code.
011D(4A6C) graphics screen paint. D=Y, E=X, A= paint color.
0120(5A66) Sprite size/enlargement initialize (resets to standard size and enlargement specification).
016B(4A18) Beep output. A=0 to 3 ie 0=BEEP 0, 1=BEEP 1, 2=BEEP, and 3=BEEP 2.

Two very simple examples to illustrate the use of Machine Code follow. Type in the Basic listing only as the Dissassembled version is for explanation only. The first prints the letter "A" on the screen and the second draws a line on the graphics screen.

The following pages give addresses in the Basic Interpretor Work Area for Disc Basic and then for Level 3 Basic. These addressses are used by Call routines to define relevant data and to accept the status of conditions which result from the Call. Accordingly they are very important in programming.

My objective has been to provide Call routines and work area addresses relevant to introductory graphics programming in Machine Code. For specific enquiries, please either address them to The Editor or see Scott or myself at the meetings. We will be happy to give whatever assistance we can.

Happy Hexhunting,
REX CHANDLER.
EDITOR.

```
10 SCREEN 2,2:CLS:R=&HF000:REM "A"
20 MC$="CD 87 00 16 45 1E 45 3E 41 CD F3 00 C9"
21 REM FOR LEVEL 3 BASIC:MC$="CD E2 39 16 45 1E 45 3E 41 CD 2D 4A C9"
30 FOR I=1 TO LEN(MC$)-1 STEP 3:POKE R,VAL("&H"+MID$(MC$,I,2)):R=R+1:NEXT
40 CALL&HF000
50 GOTO 50

F000 CD8700     CALL 0087H
F003 1645       LD   D,45H
F005 1E45       LD   E,45H
F007 3E41       LD   A,41H
F009 CDF300     CALL 00F3H
F00C C9         RET
```

```
10 SCREEN 2,2:CLS:R=&HF000:REM "LINE"
20 MC$="CD 87 00 16 25 1E 25 26 70 2E 70 3E 01 06 01 CD 9C 00 C9 "
21 REM FOR LEVEL 3 BASIC:MC$="CD E2 39 16 25 1E 25 26 70 2E 70 3E 01 06 01
 CD F1 39 C9 "
30 FOR I=1 TO LEN(MC$)-1 STEP 3:POKE R,VAL("&H"+MID$(MC$,I,2)):R=R+1:NEXT
40 CALL&HF000
50 GOTO 50

F000 CD8700     CALL 0087H
F003 1625       LD   D,25H
F005 1E25       LD   E,25H
F007 2670       LD   H,70H
F009 2E70       LD   L,70H
F00B 3E01       LD   A,01H
F00D 0601       LD   B,01H
F00F CD9C00     CALL 009CH
F012 C9         RET
```

Label	Addr.	Byte No.	Contents
QRAC	9800	8	Floating point accumulator
QRST	9808	1	Floating point status register
QROVS	9809	1	Sign at overflow time
QROEF	980A	1	---¦
QROEA	980B	2	---¦ Floating
QRUEF	980D	1	---¦ point
QRUEA	980E	2	---¦ Error
QRZEF	9810	1	---¦ processing
QRZEA	9811	2	---¦ table
QRDEF	9813	1	---¦
QRDEA	9814	2	---¦
TXTBGN	9954	2	Beginning address of text storage area
ARYBGN	9956	2	Beginning address of subscripted variable storage area
VARBGN	9958	2	Beginning address of variable storage area
FREBGN	995A	2	Beginning address of free area
FREEND	995C	2	End address of free area + 1
LINEMD	995E	1	Internal code- Line mode
LINELN	995F	1	- Line length - 1
LINENO	9960	2	- Line No.
LINEIM	9964	256	- Line image
PUTADR	9A8B	2	PUT routine entry address
GETADR	9A8D	2	GET routine entry address
STLEN1	9A8F	1	Character string register 1- Character length
STREG1	9A90	255	Contents of character string
STLEN2	9B8F	1	Character string register 2- Character length
STREG2	9B90	255	Contents of character string
ASTACK	9C8F	256	Numeric operation stack
ASTPTR	9D90	2	- Stack pointer
BSTACK	9D92	320	Character string operation stack
BSTPTR	9FD3	2	- Stack pointer
STACKB	9F10	1024	CPU stack area
RNDGEA	A310	1	Species of random number
RNDGEB	A311	1	Species of random number
RNDGEC	A312	1	Species of random number
ROUNDB	A314	1	Type of rounding
ROUNDC	A315	1	Point of rounding

```
VRAMSV   A316 2048  VRAM save area
VRAMMD   AB16    1  Display mode                      KEYIM0   AC1B    1   ---¦
SPRSIZ   AB17    1  Sprite size                       KEYIM1   AC1C    1   ---¦
SPRMAG   AB18    1  Sprite magnification              KEYIM2   AC1D    1   ---¦ Key
COLORT   AB19    1  Text screen color                 KEYIM3   AC1E    1   ---¦ Input
COLORG   AB1A    1  Graphics screen color             KEYIM4   AC1F    1   ---¦ Buffer
COLORD   ABE1    1  Back drop color                   KEYIM5   AC20    1   ---¦ (raw data)
SCRULM   ABE2    1  Text screen upper limit           KEYIM6   AC21    1   ---¦
SCRLLM   ABE3    1  Text screen lower limit           KEYIM7   AC22    1   ---¦
PATERN   ABE4    8  Pattern buffer                    KEYIMC   AC23    1   ---¦

GREENG   AC3F    1  Switching among alphabet- dieresis- graphics
SMLCAP   AC40    1  Switching between uppercase-lowercase
KEYCRC   AC41    1  Presence-absence click
CURCOL   AC44    1  Cursor-column                     KEYIR0   AC2E    1   ---¦
CURROW   AC45    1  Cursor-row                        KEYIR1   AC2F    1   ---¦
                                                      KEYIR2   AC30    1   ---¦
TIMEA    AC46    2  Time adjustment counter           KEYIR3   AC31    1   ---¦ Key
TIMEI    AC48    1  (1/60 - 1/50)                     KEYIR4   AC32    1   ---¦ Input
TIMES    AC49    1                 second             KEYIR5   AC33    1   ---¦ Buffer
TIMEM    AC4A    1                 minute             KEYIR6   AC34    1   ---¦ (rising data)
TIMEH    AC4B    1                 hour               KEYIR7   AC35    1   ---¦

PRTADR   AC4C    2  Printer output routine entry      KEYIA0   AC36    1   ---¦
CRTADR   AC4E    2  Display output routine entry      KEYIA1   AC37    1   ---¦ Key
DSKADR   AC50    2  Disk output routine entry         KEYIA2   AC38    1   ---¦ Input
PSIOAD   AC52    2  RS-232C output routine entry      KEYIA3   AC39    1   ---¦ Buffer
GKBDAD   AC54    2  Keyboard input routine entry      KEYIA4   AC3A    1   ---¦ (auto
GDSKAD   AC56    2  Disk input routine entry          KEYIA5   AC3B    1   ---¦ (repeat
GSIOAD   AC58    2  RS-232C input routine entry       KEYIA6   AC3C    1   ---¦ (data)
                                                      KEYIA7   AC3D    1   ---¦
KEYBUF   AC5A  257  Key input buffer
CSTACK   AD5E  112  GOSUB stack
CSTPTR   ADCF    2         Stack pointer
DSTACK   ADD1  200  FOR/NEXT stack
DSTPTR   AE9A    2            Stack pointer
CHSIZE   AEB5    1  Graphics screen - Character size
GRCURX   AEB6    1                    Cursor X
GRCURY   AEB7    1                    Cursor Y
CRTMOD   AEB8    1  CRT mode
EXLINE   AEBD    2  Line number at execution time
EXADDR   AEBF    2  Address of next line
POWERF   AF6D    1  Power ON flag
FILMAX   B02A    1  File max.
DIRBUF   B02B  256  Directory buffer
DIRPNT   B12B    1            -pointer
FATBUF   B12C  256  FAT buffer
--------------------------------------------------------------------------------
```

Label	Addr.	Byte No.	Contents
QRAC	8000	8	Floating point accumulator
QRST	8008	1	Floating point status register
QROVS	8009	1	Sign at overflow time
QROEF	800A	1	---¦
QROEA	800B	2	---¦ Floating
QRUEF	800D	1	---¦ point
QRUEA	800E	2	---¦ Error
QRZEF	8010	1	---¦ processing
QRZEA	8011	2	---¦ table
QRDEF	8013	1	---¦
QRDEA	8014	2	---¦
TXTBGN	8160	2	Beginning address of text storage area
ARYBGN	8162	2	Beginning address of subscripted variable storage area
VARBGN	8164	2	Beginning address of variable storage area
FREBGN	8166	2	Beginning address of free area
FREEND	8168	2	End address of free area + 1
LINEMD	816A	1	Internal code- Line mode
LINELN	816B	1	- Line length - 1
LINENO	816C	2	- Line No.
LINEIM	8170	256	- Line image
STLEN1	82A2	1	Character string register 1- Character length
STREG1	82A3	255	Contents of character string
STLEN2	83A2	1	Character string register 2- Character length
STREG2	83A3	255	Contents of character string
ASTACK	84A2	256	Numeric operation stack
ASTPTR	85A3	2	- Stack pointer
BSTACK	85A5	320	Character string operation stack
BSTPTR	86E6	2	- Stack pointer
STACKB	8730	1024	CPU stack area
RNDGEA	8B30	1	Species of random number
RNDGEB	8B31	1	Species of random number
RNDGEC	8B32	1	Species of random number
ROUNDB	8B34	1	Type of rounding
ROUNDC	8B35	1	Point of rounding

VRAMSV	8B36	2048	VRAM save area
VRAMMD	9336	1	Display mode
SPRSIZ	9337	1	Sprite size
SPRMAG	9338	1	Sprite magnification
COLORT	9339	1	Text screen color
COLORG	933A	1	Graphics screen color
COLORD	9410	1	Back drop color
SCRULM	9411	1	Text screen upper limit
SCRLLM	9412	1	Text screen lower limit
PATERN	9413	8	Pattern buffer

KEYIA0	947B	1	---¦	
KEYIA1	947C	1	---¦	Key
KEYIA2	947D	1	---¦	Input
KEYIA3	947E	1	---¦	Buffer
KEYIA4	947F	1	---¦	(auto
KEYIA5	9480	1	---¦	(repeat
KEYIA6	9481	1	---¦	(data)
KEYIA7	9482	1	---¦	

GREENG	9484	1	Switching among alphabet- dieresis- graphics
SMLCAP	9485	1	Switching between uppercase-lowercase
KEYCRC	9486	1	Presence-absence click
CURCOL	9489	1	Cursor-column
CURROW	948A	1	Cursor-row

TIMEA	948B	2	Time adjustment counter
TIMEI	948D	1	(1/60 - 1/50)
TIMES	948E	1	second
TIMEM	948F	1	minute
TIMEH	9490	1	hour

PRTADR	94A0	2	Printer output routine entry
CRTADR	94A2	2	Display output routine entry
KEYBUF	94A4	257	Key input buffer
CSTACK	95A7	112	GOSUB stack
CSTPTR	9618	2	Stack pointer
DSTACK	961A	200	FOR/NEXT stack
DSTPTR	96E3	2	Stack pointer
CHSIZE	96FE	1	Graphics screen - Character size
GRCURX	96FF	1	Cursor X
GRCURY	9700	1	Cursor Y
CRTMOD	9701	1	CRT mode
EXLINE	9705	2	Line number at execution time
EXADDR	9707	2	Address of next line
POWERF	97E2	1	Power ON flag

KEYIM0	9460	1	---¦	
KEYIM1	9461	1	---¦	
KEYIM2	9462	1	---¦	Key
KEYIM3	9463	1	---¦	Input
KEYIM4	9464	1	---¦	Buffer
KEYIM5	9465	1	---¦	(raw data)
KEYIM6	9466	1	---¦	
KEYIM7	9467	1	---¦	
KEYIM8	9468	1	---¦	

KEYIR0	9473	1	---¦	
KEYIR1	9474	1	---¦	
KEYIR2	9475	1	---¦	
KEYIR3	9476	1	---¦	Key
KEYIR4	9477	1	---¦	Input
KEYIR5	9478	1	---¦	Buffer
KEYIR6	9479	1	---¦	(rising data)
KEYIR7	947A	1	---¦	

```
10 REM **  RESET  SF-400 PRINTER  **
11 REM by Scott MacDonald.
12 REM  To reset the line length for     the Plotter/Printer or Serial type
printer linked to the SC-3000 SEGA   keyboard. Type RUN then press <CR>
13 REM Enter the line length desired     when requested then press CR. This
Basic program then self destructs.
14 REM
15 CLS:PRINT,,,,"    **  RESET  SF-400  PRINTER  **":B=0:N=PEEK(&H8169):IFN=10T
HENN=&HFF:V$="DISK":LIMIT&HFF00:B=1:GOTO19
16 V$="3B":IFN=&HC7THENV$="3A"
17 IFN=&HBFTHENV$="2B"
18 IFPEEK(&H815F)=0THENPOKE&H8169,N-1:POKE&H815F,1
19 R=N*256-65536:PRINT,,,,"INSTALLING ROUTINE FOR ";V$;" BASIC",,,
20 INPUT"Line length ? ";LL:PRINT,,"Doing it now":IFBTHENPOKE&H0BD2,LL:GOSUB22:
POKE&HB70F,0:END
21 POKE&H94A0,0:POKE&H94A1,N:GOSUB23:GOSUB22:CALLN*256+144-65536:END
22 PRINT,,"ALL DONE":BEEP:BEEP:RETURN
23 MC$="F5C5D5E5217E20CD351A3016217E20CD351A300EFE0A286E211F00CD351A30023E2057F
501A00FDBDDCB77280EC50EFBCD033AC10B78B1285218EC":GOSUB26
24 MC$="F3003E0BD3DF0E2ECD033A0608CB0A3F3E0ACE00D3DF0E2ECD033A10F03E0A000000000
000D3DF0E64CD033AFBF1FE0D2814FE2038153A04973C320497FE":GOSUB26
25 MC$=HEX$(LL)+"200A3E0DC34C703E00320497E1D1C1F1C93E40CD4824210098C3621A"
26 FORM=1TOLEN(MC$)STEP2:POKER,VAL("&H"+MID$(MC$,M,2)):R=R+1:NEXT:RETURN
```

```
1 REM *** ROMAN NUMBER CONVERSION ***
2 REM by Scott MacDonald.
10 FORN=1TO9:READU$(N),T$(N):NEXT:A=0
20 DATAI,X,II,XX,III,XXX,IV,XL,V,L,VI,LX,VII,LXX,VIII,LXXX,IX,XC
30 FORN=0TO9:FORM=0TO9:PRINTA;" = ";T$(N);U$(M):A=A+1:NEXTM,N
40 FORN=0TO9:FORM=0TO9:PRINTA;" = C";T$(N);U$(M):A=A+1:NEXTM,N:PRINTA;" = CC"
```

```
1 REM  THIRD-DIMENSION  PLOTTING             Copyright   SCOTT MACDONALD
         FILE: 3D-PLOT.001
10 SCREEN2,2:CLS:X1=127.5:X2=X1*X1:Y0=191:Y1=Y0/2:Y2=Y1:CURSOR2,183:PRINT"F=
(R-1)*COS(R*18)":GOSUB30:END
20 Y=Y-Y1:PSET(X1+N,Y):PSET(X1-N,Y):NEXTM,N:RETURN
30 FORN=0TOX1:X4=N*N:L=SQR(X2-X4):FORM=-LTOLSTEP6:R=SQR(X4+M*M)/X1:Y=Y0-(M/3
+(R-1)*COS(R*18)*Y2):IFM=-LTHENK=Y:H=Y:GOTO20
40 IFY>KTHENK=Y:GOTO20
50 IFY<HTHENH=Y:GOTO20
60 NEXTM,N:RETURN                               FORMULA:  F=(R-1)*COS(R*18)
```

EDITOR'S NOTE.

High scores in any of the games will be printed here on notification to The Editor. I think an honorary system is best to stimulate intrest. Other members may of course request that you demonstrate the secrets and skills which permitted such scores.

JENNY WILLIAMS achieved 84,200 (maze 9) on CHACK'N POP.

A man's coat was left at the April meeting. It is currently being held by Allan Rodd who will return it given a correct description.

Advertisments relating to SEGA computer matters - buying, selling, information exchange etc, will be printed in SEGAMAG free of charge to members.

Programming assistance is available at meetings or via SEGAMAG. A programmer with experience relevant to your situation will assist. We don't just play games at meetings.

MARKET PLACE.

FOR SALE: SEGA SC-3000 COMPUTER,
WITH NINE CARTRIDGE GAMES PLUS SIX EDUCATIONAL GAMES AND TWO JOYSTICKS, ALL SET READY TO GO. WORTH $654. SELLING FOR $300 o.n.o.

(02) 543 6949.

```
1 REM GRAPHIC.100 by Scott MacDonald:- DRAWS TRINAGLE SHAPES.
5 SCREEN2,2:CLS
10 DIMX(20),Y(20):FORN=1TO20:X=INT(RND(1)*254):Y=INT(RND(1)*191)+1:X(N)=X:
Y(N)=Y:FORB=1TON:LINE(X,Y)-(X(B),Y(B)):NEXTB,N
```

```
10 REM ***  PRIME NUMBERS  ***
20 REM by Scott MacDonald:- IDENTIFIES PRIME NUMBERS.
30 CLS:INPUT"Start number ";Q:IFQ<3THENQ=3
35 IFQ/2=INT(Q/2)THENQ=Q-1
40 INPUT"  End number ";R:IFR<Q+1THEN30
80 FORN=QTORSTEP2:FORM=3TOSQR(N)+1STEP2:IFNMODMTHENNEXTM:PRINTN;" ";
90 NEXTN
```

LETTERS TO THE EDITOR.

Just sending in a short listing for people who don't have time (or effort) to design sprites (or graphics characters) on paper. This listing covers MAG 0 sprites and CHR$ codes. I also have a listing for MAG 1 sprites if anyone is interested.

There are no instructions in the program so I have included them here.

After typing RUN, the screen shows:

```
GOTO 80

........
........
........
........
........
........
........
........
```

Use the cursor keys to move over the grid of dots and press 'X', (X is the only character the program responds to). The dots equal '0''s and the 'X's equal '1''s. If you make a mistake, replace the 'X' with a full stop, NOT a space.

When you're happy with the pattern, move the cursor up to the 'GOTO 80' and hit enter.

After a few seconds, some numbers will flash up (simply to break the monotony) and a pattern code will show up. To see the CHR$ to make sure its satisfactory, flick over to the graphics screen (SHIFT-BREAK ED). It shows the MAG 0 and MAG 2 versions of the CHR$.

B.N.LACEY.

```
10 SCREEN 2,1:COLOR 2,1,,1:CLS:SCREEN 1,1:CLS:DIM D$(15),B$(15)
20 P=&H3C02
30 CURSOR 0,0:PRINT  "GOTO 80"
40 FOR Y=1 TO 8:FOR X=0 TO 7
50 VPOKE Y*40+X+P,46
60 NEXT:NEXT:CURSOR 0,9
70 FOR A=0 TO 15:READ D$(A):NEXT:FOR A=0 TO 15:READ B$(A):NEXT:END
80 A$="":FOR Y=1 TO 8:FOR X=0 TO 7
90 A=VPEEK(Y*40+X+P):IF A=46 THEN A$=A$+"0"
100 IF A=88 THEN A$=A$+"1"
110 NEXT X,Y
120 IF LEN(A$)<>64 THEN END
130 D$="":FOR X=1 TO 64 STEP 4
140 B$=MID$(A$,X,4)
150 FOR Y=0 TO 15:IF B$=D$(Y) THEN D$=D$+B$(Y)
160 NEXT Y:CURSOR 0,12:PRINT X;" ":NEXT X
170 PRINT D$:PATTERNS#0,D$:PATTERNC#255,D$:SCREEN 2,1:CLS:CURSOR 120,100:PRINT CHR$(255)
180 MAG2
190 SPRITE 0,(150,100),0,2
200 STOP
210 DATA "0000",0001,0010,0011,0100
220 DATA "0101",0110,0111,1000,1001
230 DATA "1010",1011,1100,1101,1110
240 DATA "1111"
250 DATA "0",1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
```

Club Library

Cassettes

Educational

1 Basic 1 Tutorial

2 Demon Devision

3 Marauding Multiplication

4 Satellite Subtraction

5 Super Tape 1(general knowledge)

6 Watch Me Draw

Adventure Games

1 Environoid'Text Adventure

2 Heroic Conquest'Text Adventure

3 Solar Conquest'Space Adventure

4 Ice Cream Stall

Buisness

1 Mailing List

2 Programs 1 (Various Loan And Interest calculations)

3 Programs 2 (Data & Analysis Calculations)

Books

1 Exploring Graphics

2 Basic Machine Code Listing For SC-3000 & Disk Basic

B.Minnett,
LIBRARIAN.

The above may be hired on a weekly or monthly basis by members at modest rates. Enquiries may also be made of The Librarian regarding the hiring of the Group's hardware by established members. The more expensive items will involve a deposit (returned) as well. We intend to build up the Library into one of SSUG'S features.

Any donations of original game tapes or cartridges or other programmes will be gratefully received. We will even purchase better quality pre-loved ones if the price is right.

EDITOR.

SEGAMAG New South Wales
Published by Sydney Users Group
PO Box 455
Gladesville N.S.W. 2111

MR ALLAN 20 RODD
19 CHAMPION RD
GLADESVILLE
2111 N.S.W.