JANUARY/FEBRUARY 1985

SEGA® COMPUTER

The Official Sega User Club Magazine

User's Letters

ARTICLES BY
Michael Howard — Graphics
Ian Nicholson — The Joystick
Colin Smith — Machine Code
Andrew Flexman — Program Disection

PROGRAMS
Balune
Japanese Man
Rule Britannia
Mazatron
Serpent — Program of the Month

INTRODUCTION

Dear Member

With Phil Kenyon software hunting in the frozen wastes of the U.K, it falls to me, the good looking member of the family, to wish you all the best for the coming year.

1985 will see Grandstand striving to greatly increase the range of software available for your favourite micro. We are already well under way with negotiations for a range of software from John Sands, the Sega distributors for Australia. This will in effect boost the range of cassette titles available by almost 30 titles. We will, of course, continue to produce high quality home grown software and a brand new range of exciting arcade cartridge games from Sega Japan in our January production schedule.

January and February will be a busy time for all of us here at Grandstand as we prepare the details of our Amstrad CPC464 campaign, due for a March 1st launch. For all those involved it will mean many months of evening demonstrations, roadshows around the countryside and putting oneself in grave danger of being bitten by the dog when you eventually arrive home and he does not recognise you.

It is becoming traditional to finish the introduction with a comical story and I would like to import to you some of the strange letters and requests we get from time to time.

One new Users Club member thanked us for sending out their free cassettes but was getting a terrible screeching noise when they tried to play them on the families stereo system.

Another member complained of tape read error, and when the cassette was tested, our Service Department found that the owner had inadvertantly pressed the record key on his cassette recorder and had made a perfect recording of the familes pet canary, singing its heart out.

Finally we had a complaint that one getting their brand new computer home the proud owners were experiencing difficulties obtaining colour on the families portable T.V. A little difficult when we found it to be a black and white set!

Yours faithfully
S. Kenyon
GRANDSTAND LEISURE LTD

Contents **Page**

LOCAL SEGA USERS CONTACTS

PUKEKOHE SEGA USERS CLUB
C/o 4 Roose Avenue
Pukekohe
Contact: Selwyn
Ph. Pukekohe 86-583

AUCKLAND CENTRAL
c/o 287 Broadway Furniture
Newmarket
Contact: George Shaw
Ph. 547-543

ROTORUA
Rotorua Sega Users Club
C/- 61 Devon Street
ROTORUA

TOKOROA
Tokoroa Sega Users Group
C/- 1 Pio Pio Place
TOKOROA
Contact Geoff Phone Number 67105
Tokoroa

TARANAKI
South Taranaki Microcomputer Society
D. M. Beale
7A Clive Street
HAWERA

NAPIER
Napier Sega Users Club
Sec E. P. Lins
41 Higgins Street
NAPIER

The above are contact names and addresses for Sega Users Clubs. If you wish to have your club advertised write to Sega Users Club P.O. Box 2353, Auckland

READER'S LETTERS

Dear Editor

Is it possible to get the Sega to do power calculations such as:

X^n, ie $23^{2.7}$? I can find no reference to this in the handbook that came with the machine.

Alternatively, can the Sega obtain antilogarithms? It can do LOGS, but will it go the other way. If it won't do powers, but can do antilogarithms then I can use logs to get the powers. I suspect that it must be able to. It would be too silly for it to be able to get logarithms, but not antilogs.

Any advice would be gratefully received, as I now have a number of TSR 80 stats programs that look as if they will otherwise be easy to translate.

Regards
Dr A S Campbell

EDITORS REPLY

It is possible to do X^n type calculations using the ^ sign. This is found on the top right of the keyboard on the AUTO key.

To obtain antilogarithms use the EXP function. Please see page 130 of the Sega Level III manual.

DEAR EDITOR

Business programs are a bit thin on the ground. What's coming along?

S Cassidy
BUCKLANDS BEACH

EDITORS REPLY

This frequently asked question highlights the software equivalent of the story of the infinite number of monkeys and the infinite number of typewriters.

Theoretically, eventually one of them will write the entire works of Shakespeare — but do not hold your breath...

Games software contains relatively few user controls, and the programs themselves are relatively easily and obviously tested. Business and utility software requires use of virtually every key on the keyboard, plus the shifts and controls. An infinite number of monkeys and an infinite number of computers is required to test something like a word processor through every possible combination of control and display sequence.

The task of testing and design of the user interface is several orders of magnitude greater than an arcade game, and the consequences of imperfect software potentially more troublesome. Add to this the temptation strewn in the path of the programmer by the wealth of features that lurk within the SEGA so 'business' software takes time to develop.

We have the SEGA 32K Word Processor and this will certainly be welcomed by the many Members who want to use the SEGA as WP system.

DEAR EDITOR

I have a Sega SF7000 control station. I would like to have a dual disk drive for business is this possible?

EDITOR'S REPLY

Across the Tasman the John Sands corporation have interfaced a dual disk drive. We hope to be seeing one of these units this year. The Australians have also managed to Network 8 Computers to 1 disk drive.

We also have the following business orientated packages available:

Easy Writer
32K Personal Record Keeper
Accounts Receivable
Accounts Payable
Mailing List
File System
Graph and Chart Presentation

More business orientated software is forthcoming on disk and cassette soon.

DEAR EDITOR

I have had a SC3000 for just on 3 months now and my family and I am having a lot of fun using it. However, I would like some more educational software. Do you have anything coming.

EDITOR'S REPLY

After listening through many closed doors and standing near open windows I hear that there are 3 new educational programmes that will be released in January. There is also a shipment of software arriving from Australia in the near future. This should include educational software.

To see just what new software is on the way please see the reviews section of this magazine.

DEAR EDITOR

One thing has been puzzling me about the Sega, that is how to use DATA statements.

Could you please inform me about this.

Ben Holst
HELENSVILLE

P.S. Great magazine

EDITORS REPLY

Please see the programming explanation section of this magazine for help with DATA READ and RESTORE.

DEAR EDITOR

I am a new computer owner so I am hoping you can clear up a small problem I am having.

We have a copy of your "MUNCHMAN" cassette. I can only get this to load on the first side. I know there is program on the second side but it will not load.

D Hatley
BUCKLANDS BEACH

EDITORS REPLY

All the cassettes produced by Grandstand are recorded on both sides. It is exactly the same program on each side of the cassette, however, each side is recorded a slightly different level. This means that if your computer refuses to load on side one, then the different levels on side two are more likely to allow you to load the program into your computer.

Dear Editor

The article "Program Dissection" in the September issue of "Sega Computer" is the most helpful we have read. We hope there will be many more such articles to follow.

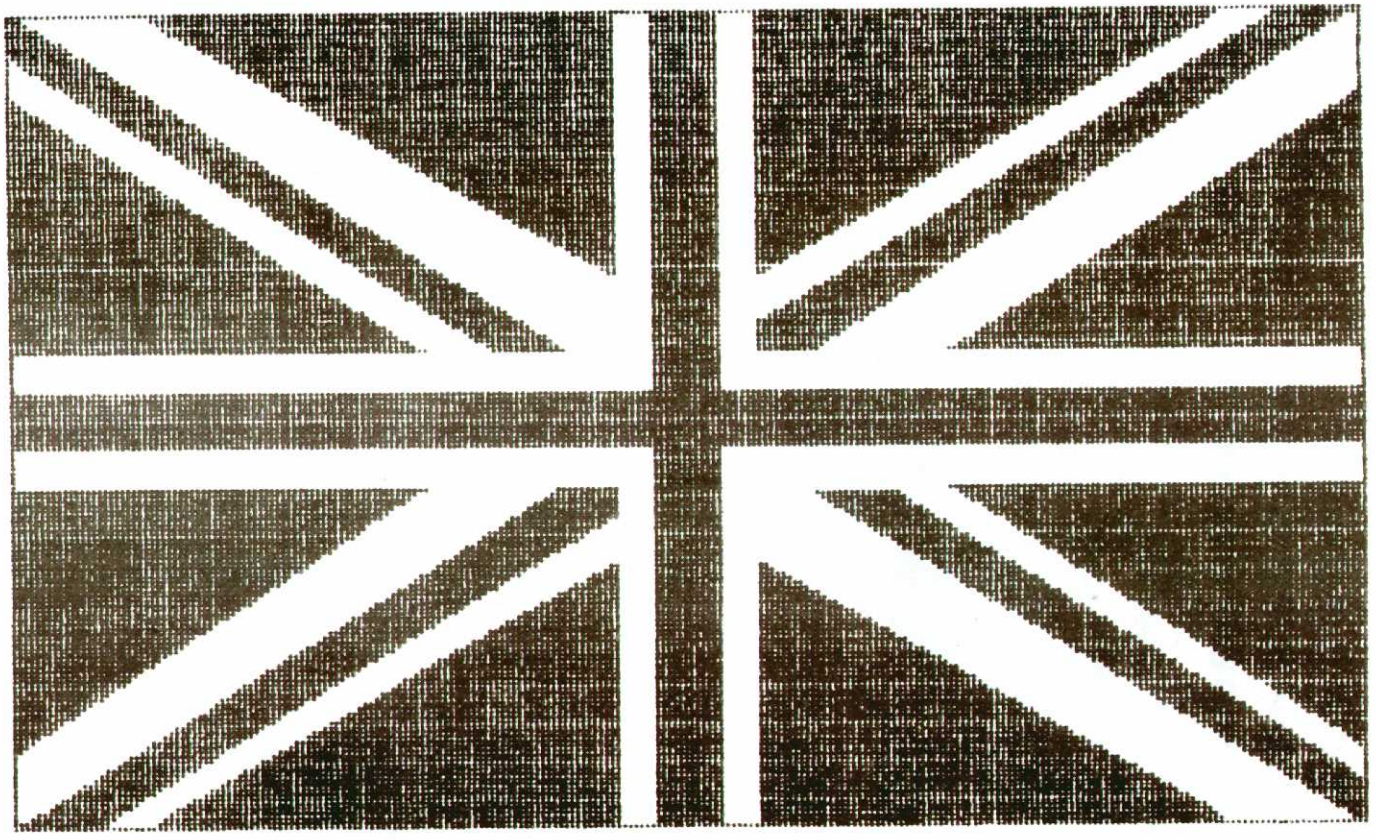
We look forward to receiving further issues of this magazine.

F K Maynard
Wellington

Editors Reply
Thanks for your kind comments. We plan to do many more such articles and some more articles on Dissection.

RULE BRITANNIA

This program should appeal to all the British patriachs. Lines 1Ø-1ØØ draw the British flag and lines 11Ø-2ØØ play Rule Britannia.



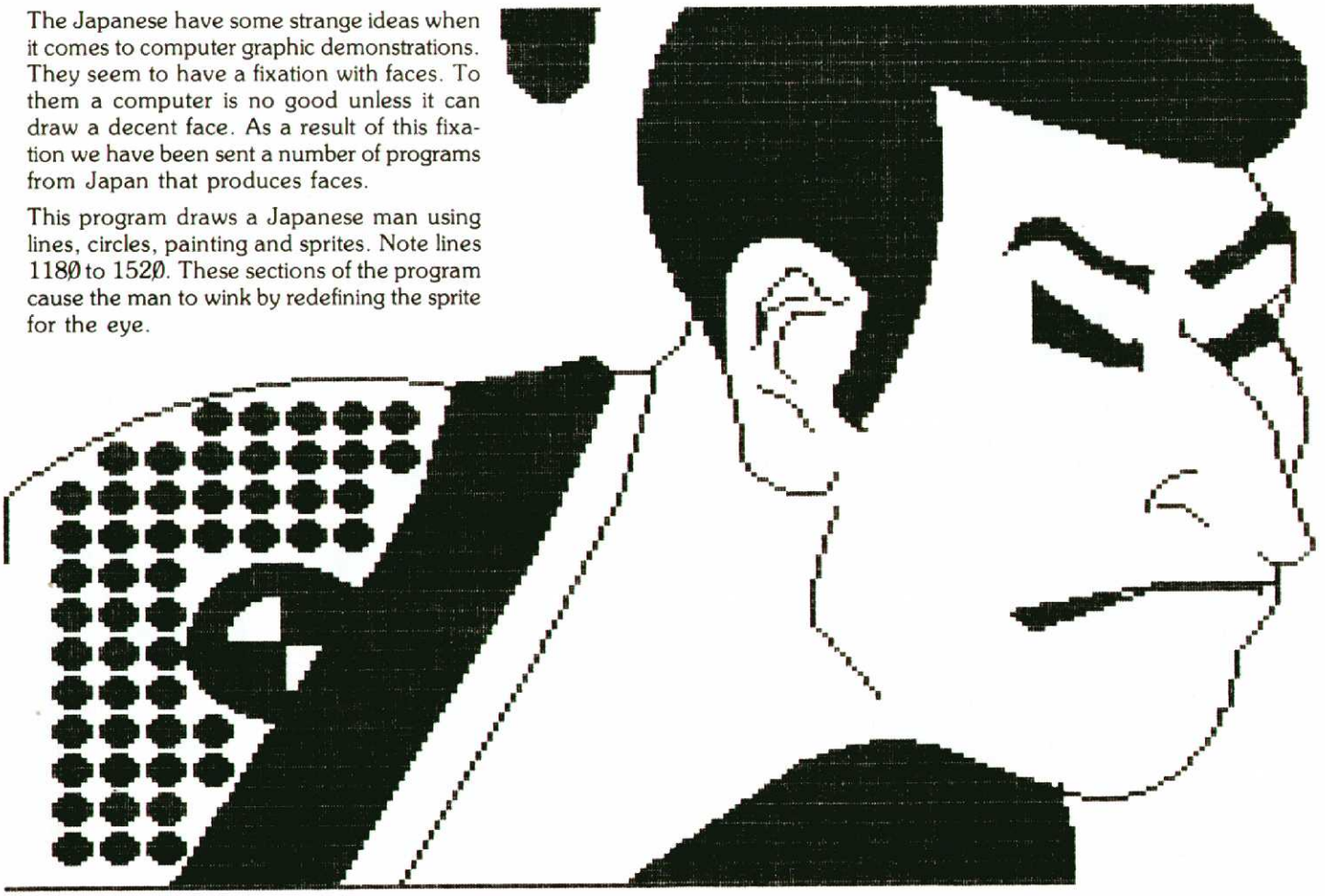
```
10 SCREEN2,1:CLS:COLOR15,15,(0,0)-(255,191),15:SCREEN2,2
20 LINE(10,10)-(245,181),8,B:LINE(10,10)-(122,90),,B:LINE(245,10)-(133,90),,B:LI
NE(245,181)-(133,101),,B:LINE(10,181)-(122,101),,B
30 LINE(115,11)-(115,65),4:LINE-(35,11):LINE(140,11)-(140,65):LINE-(220,11):LINE
(140,180)-(140,126):LINE-(220,180):LINE(115,180)-(115,126):LINE-(35,180)
40 LINE(11,81)-(82,81):LINE-(11,28):LINE(244,81)-(173,81):LINE-(244,28):LINE(244
,110)-(173,110):LINE-(244,165):LINE(11,110)-(82,110):LINE-(11,165)
50 LINE(11,11)-(106,81),8:LINE-(95,81):LINE-(11,19)
60 LINE(244,11)-(150,81):LINE-(140,81):LINE-(140,75):LINE-(230,11)
70 LINE(244,180)-(150,111):LINE-(163,111):LINE-(244,172)
80 LINE(11,180)-(105,110):LINE-(115,110):LINE-(115,116):LINE-(26,180)
90 PAINT(123,11),8:PAINT(11,29),4:PAINT(36,11):PAINT(209,11):PAINT(244,29):PAINT
(244,111):PAINT(141,127):PAINT(113,128):PAINT(11,111)
100 PAINT(12,12),8:PAINT(240,11):PAINT(243,179):PAINT(18,179)
110 FORJ=1TO3:FORX=1TO26:READC1,C2,C3,D:SOUND1,C1,14:SOUND2,C2,14:SOUND3,C3,14:F
ORDE=1TOD:NEXTDE,X:RESTORE130:NEXTJ
120 SOUND0:END
130 DATA523,311,208,50,30000,30000,30000,25,523,311,208,25
140 DATA554,349,208,24,30000,30000,30000,1,554,349,208,24,30000,30000,30000,25,5
23,349,220,25
150 DATA554,349,117,25,523,349,220,25,466,349,139,25,415,349,147,25
160 DATA392,233,156,75,30000,30000,30000,25
170 DATA622,208,131,50,554,196,117,50
180 DATA523,311,208,12,415,311,208,12,554,311,117,12,466,311,117,12,622,208,131,
37,554,349,208,37
190 DATA523,311,208,75,466,392,156,75
200 DATA415,262,208,100,30000,30000,30000,25
```

THE SCOWLING MAN

BY SEGA

The Japanese have some strange ideas when it comes to computer graphic demonstrations. They seem to have a fixation with faces. To them a computer is no good unless it can draw a decent face. As a result of this fixation we have been sent a number of programs from Japan that produces faces.

This program draws a Japanese man using lines, circles, painting and sprites. Note lines 118Ø to 152Ø. These sections of the program cause the man to wink by redefining the sprite for the eye.



```
10 SCREEN 2,2:CLS
20 REM ** JAPANESE MAN **
30 RESTORE 940
40 FORL=0TO96:READ A,B,C,D
50 LINE (A,B)-(C,D),1
60 NEXTL
70 LINE(47,120)-(47,139),2:LINE(38,130)-(55,130),2
80 CIRCLE(217,110),5,1,1,0,0.4
90 CIRCLE(161,113),60,1,1,0.87,0.93
100 CIRCLE(194,85),27,1,1,0,0.09
110 CIRCLE(194,85),27,1,1,0.91,1
120 CIRCLE(184,140),23,1,1,0.04,0.25
130 CIRCLE(204,17),17,1,1,0,0.17
140 CIRCLE(204,17),17,1,1,0.92,1
150 CIRCLE(150,176),25,1,1,0.65,0.85
160 CIRCLE(177,52),8,1,1,0.65,0.83
170 CIRCLE(178,60),12,1,1,0.65,0.78
180 CIRCLE(93,13),6,1,1,0,0.5
190 CIRCLE(169,30),61,1,1,0.4,0.6
200 CIRCLE(99,60),18,1,1,0.02,0.165
210 CIRCLE(59,160),84,1,1,0.6,0.79
220 CIRCLE(132,57),10,1,1,0.65,0.88
230 CIRCLE(136,89),11,1,1,0.21,0.49
240 CIRCLE(60,35),99,1,1,0,0.092
250 CIRCLE(60,35),99,1,1,0.965,1
260 CIRCLE(149,80),8,1,1,0.33,0.6
270 CIRCLE(125,90),12,1,1,0.8,0.96
```

```
280 CIRCLE(202,104),8,1,1,0.5,0.75
290 CIRCLE(47,130),16,10,1,0.28,0.9
300 CIRCLE(47,130),10,2,1,0.2,0.95
310 CIRCLE(194,125),57,1,1,0.495,0.57
320 RESTORE 1140:FORD=0TO51:READ X,Y:CIRCLE(X,Y),3,4,1,0,1,BF:NEXTD
330 PAINT(90,10),1:PAINT(150,5),1:PAINT(180,46),1:PAINT(215,46),1:PAINT(172,49),
1
340 PAINT(50,115),10: PAINT(80,90),9:PAINT(160,160),5
350 PAINT(180,65),1:PAINT(210,70),1
360 PAINT(175,125),6:PAINT(50,125),2:PAINT(45,135),2
370 PATTERNS#0,"0001010103030307"
380 PATTERNS#1,"07070F0F0F1F1F3F"
390 PATTERNS#2,"FEFEFEFCFCFCFCFC"
400 PATTERNS#3,"F8F8F8F8F0F0F0F0"
410 PATTERNS#4,"0103030307070F0F"
420 PATTERNS#5,"0F1F1F3F3F3F7F7F"
430 PATTERNS#6,"FFFFFFFFFEFEFEFE"
440 PATTERNS#7,"FCFCFCF8F8F8F0F0"
450 PATTERNS#8,"0307070F0F1F1F3F"
460 PATTERNS#9,"3F7F7F7FFFFFFFFF"
470 PATTERNS#10,"FFFFFEFEFEFCFCF8"
480 PATTERNS#11,"F8F0F0F0E0E0C0C0"
490 PATTERNS#12,"07070F0F1F1F3F3F"
500 PATTERNS#13,"7F7FFFFFFFFFFFFF"
510 PATTERNS#14,"FFFFFEFEFCFCF8F8"
520 PATTERNS#15,"F0F0F0E0E0C0C080"
530 PATTERNS#16,"07070F0F1F1F3F3F"
540 PATTERNS#17,"7F7FFFFFFFFFFFFF"
550 PATTERNS#18,"FFFFFEFEFCFCF8F8"
560 PATTERNS#19,"F0F0E0E0C0C08080"
570 PATTERNS#20,"0F0F1F1F3F3F7F7F"
580 PATTERNS#21,"FFFFFFFFFFFFFFFF"
590 PATTERNS#22,"FEFEFCFCF8F8F0F0"
600 PATTERNS#23,"F0E0E0C0C0808000"
610 PATTERNS#24,"0F0F1F1F3F3F7F7F"
620 PATTERNS#25,"FFFFFFFFFFFFFFFF"
630 PATTERNS#26,"FFFEFEFCFCF8F8F0"
640 PATTERNS#27,"F0E0E0C0C0808000"
650 PATTERNS#28,"0000010103030707"
660 PATTERNS#29,"0F0F1F1F3F3F7FFF"
670 PATTERNS#30,"FFFFFFFFFFFFFFFF"
680 PATTERNS#31,"FFFFFEFCFCF8F8F0"
690 PATTERNS#32,"0000E0F8FEFFFFFF"
700 PATTERNS#33,"FF3F0F0707030100"
710 PATTERNS#34,"000000000C0E0E0"
720 PATTERNS#35,"C0C0C0808000000"
730 PATTERNS#36,"00000080C0E0F0FC"
740 PATTERNS#37,"FFFFFFFF7F1F0702"
750 PATTERNS#38,"0000000000000030"
760 PATTERNS#39,"F0E0C0C0C0808000"
770 PATTERNS#40,"3F7FFFFFFFFF8F8"
780 PATTERNS#41,"FEFFFF7F3F3F1F0F"
790 PATTERNS#42,"0000000000000000"
800 PATTERNS#43,"0000000000000000"
810 MAG1
820 SPRITE0,(96,75),0,1
830 SPRITE1,(92,91),4,1
840 SPRITE2,(86,107),8,1
850 SPRITE7,(80,121),12,1
860 SPRITE8,(74,134),16,1
870 SPRITE9,(69,146),20,1
880 SPRITE10,(64,156),24,1
890 SPRITE11,(56,164),28,1
900 SPRITE20,(49,112),32,10
910 SPRITE21,(39,132),36,10
920 SPRITE22,(31,121),40,10
930 REM ** LINE DATA **
940 DATA 85,0,87,14,99,14,101,0,212,0,219,9,211,33,199,31,199,31,174,22,174,22,1
51,12
950 DATA 211,33,216,43,217,43,213,44,213,44,209,50,209,50,205,53,205,53,203,53,2
03,53
```

```
960 DATA 201,55,201,55,204,58,204,58,215,49,215,49,219,50,219,50,217,43,219,50,2
19,55
970 DATA 219,55,218,58,218,58,216,65,216,66,220,75,218,58,213,63,213,63,216,65,2
13,63,211,64
980 DATA 211,64,207,69,207,69,204,70,200,68,200,65
990 DATA 216,69,208,74,216,90,222,110,196,103,200,103,200,103,204,107,216,115,21
6,122
1000 DATA 216,122,209,133,209,133,207,145,185,163,164,155,179,160,181,169,181,16
9,181,180
1010 DATA 136,155,105,180,194,59,195,55,195,55,194,54,194,54,192,54,192,54,188,5
2
1020 DATA 188,52,180,45,172,46,170,51,178,48,185,54,185,54,194,59,193,75,192,69
1030 DATA 192,69,191,70,191,70,175,57,175,57,175,69,175,69,187,74,187,74,193,75
1040 DATA 215,119,194,120,194,120,171,125,171,125,174,128,174,128,194,120,194,12
0,213,121
1050 DATA 213,121,214,119,191,74,179,69,148,142,137,127,141,96,139,100,125,90,12
2,75
1060 DATA 122,75,122,57,122,57,125,50,139,50,144,54,144,54,147,59,147,59,143,75
1070 DATA 122,75,117,62,117,62,111,48,110,75,103,76,103,76,100,72,100,72,91,75,9
1,75,76,78,76,78,70,100
1080 DATA 70,100,28,179,103,76,90,110,90,110,56,179,110,75,104,100,104,100,91,13
0,91,130,67,179
1090 DATA 143,60,142,61,142,61,138,59,138,59,135,54,135,54,133,54,133,54,130,58
1100 DATA 130,58,128,61,128,61,127,70,129,66,130,62,130,62,133,60,133,60,136,60,
139,63,134,63
1110 DATA 134,63,132,66,132,66,131,69,131,69,131,73,131,73,134,77,0,180,180,180
1120 DATA 47,120,47,139,38,130,55,130
1130 REM ** CIRCLE DATA **
1140 DATA 67,84,59,84,51,84,43,84,35,84,67,92,59,92,51,92,43,92,35,92,27,92,19,9
2,59,100,51,100,43,100,35,100,27,100
1150 DATA 19,100,11,100,59,108,51,108,43,108,35,108,27,108,19,108,11,108,27,116,
19,116,11,116
1160 DATA 27,124,19,124,11,124,27,132,19,132,11,132,27,140,19,140,11,140,35,148,
27,148
1170 DATA 19,148,11,148,35,156,27,156,19,156,11,156,27,164,19,164,11,164,19,172,
11,172,27,172
1180 REM ** PATTERN DATA **
1190 PATTERNS#128,"0080C0F0F8FCFEE0"
1200 PATTERNS#129,"C0C0CFDFFF7F0F03"
1210 PATTERNS#130,"00000000000000C0"
1220 PATTERNS#131,"201804C2E1F0F0F9"
1230 PATTERNS#132,"201804C2E1F0F0F9"
1240 PATTERNS#133,"FF1F010000000000"
1250 PATTERNS#134,"00000000A0E0E0E0"
1260 PATTERNS#135,"E0E0E00000000000"
1270 MAG1
1280 SPRITE3,(175,57),128,13
1290 SPRITE4,(183,65),132,13
1300 PATTERNS#128,"80C0E0F8FCFEE0C0"
1310 PATTERNS#129,"C0CFDFE3F97E0F03"
1320 PATTERNS#130,"0000000000000000"
1330 PATTERNS#131,"0000C0E0F07081F1"
1340 PATTERNS#132,"0000C0E0F07081F1"
1350 PATTERNS#133,"FF1F010000000000"
1360 PATTERNS#134,"0000000000A0E0E0"
1370 PATTERNS#135,"E0E0E00000000000"
1380 PATTERNS#136,"0000000000000000"
1390 PATTERNS#137,"0000000010303C241"
1400 PATTERNS#138,"0000040C1C387838"
1410 PATTERNS#139,"1810D0B0B870E0C0"
1420 PATTERNS#140,"0000000010303C241"
1430 PATTERNS#141,"331E180000000000"
1440 PATTERNS#142,"1810D0B0B870E0C0"
1450 PATTERNS#143,"0000000000000000"
1460 MAG1
1470 SPRITE3,(175,57),128,13
1480 SPRITE4,(183,65),132,13
1490 SPRITE5,(205,56),136,13
1500 SPRITE6,(205,64),140,13
1510 FORW=0TO500:NEXTW
1520 GOTO1190
```

CONTACT PAGE

By now there are over 6000 Sega owners in New Zealand. Unfortunately most of these good people don't know that the others exist, so this page is a new regular feature to bring Sega owners out of the closet and into the open.

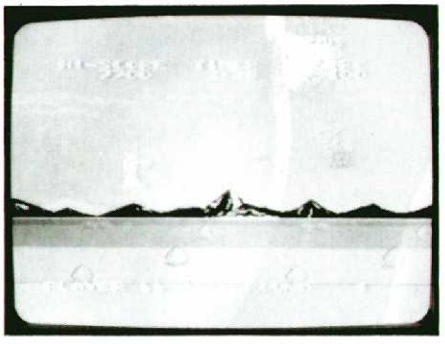
If you wish to have your name, address and a contact phone number listed, just write to:

Sega Users Club
CPO Box 2353
AUCKLAND

with the relevant information.

SOFTWARE REVIEW

ORGUS



All you Scrambler and Defender fans out there, this is the game for you. Horizontal scrolling and excellent graphics make this an addictive game.

With 3 different stages and a giant enemy base at the end that you must destroy makes this an exciting game.

You start as a robot (orgus) and can fire at the UFO, and kick the ground bases. You can also transform into the "Flyer" to gain speed, but you must be very careful as all the enemies bullets can destroy you.

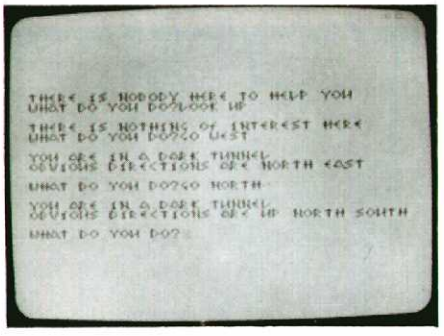
MATHS HANG UP

A delightful educational maths program based on the popular word game hang man. You must first select addition, subtraction, multiplication, division or a mixture of everything, and then the level, from beginner to senior. Its then a race against time to answer the questions before your time runs out. If you are incorrect, or your time runs out the gallows is assembled and you are eventually hung. A fast paced eductaional game that will keep you on your toes and improve mental arithmetic. A rating is given at the end of each round and you can keep track of your progress as you move into harder levels. A must for the mental arithmetic.

PYRAMID

This is another adventure game by D Van Kan (of Death Satellite fame). The idea of the game is to find the wooden staff which is a priceless artifact. There are many endless tunnels to get lost in and places in which to die!

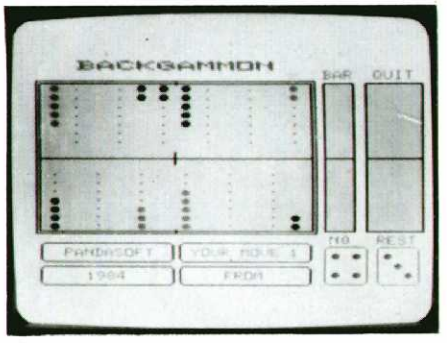
The program cleverly redefines the screen text so that it looks like an ancient Egyptian script.



SAFARI RACER

I am hooked on this one and so will you be. If you have ever seen Pole Position or Turbo in the arcades, well you will know this game. 3D racing car action driving something James Bond drives during the weekend, you can speed up to 300km/h. There is no speed limit, but watch out for the curves. You also have wild animals such as rhinos, lions, and elephants and leopards to contend with. Unfortunately all of these creatures have suicidal tendencies (or they have not learn't the road code). Dangerous boulders also litter the road. Every so often you must stop and refuel at the petrol pumps on the side of the road. If you do not your car runs out of gas and you have to walk home. The graphics and sound on this game would rival most arcade games. I am certainly going to be a buyer of this Sega game!

BACKGAMMON

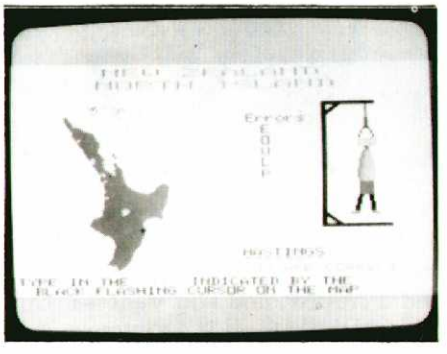


A must for all backgammon fans. Either play another person on the screen or match your wits against the computer, that is if you like losing.

GEOGRAPHY QUIZ I & II

BY IAN NICHOLSON

The graphics for these two programs have been on Ian's 1st Programme the Sega Graphic Designer. The programme produces a detailed map of an area of the world and you are asked to identify a city or geographical landmark at a certain position on the map. This is not nearly as easy as it sounds as you have to play a game of hangman at the same time. If you make 6 unsuccessful guesses you get hung!

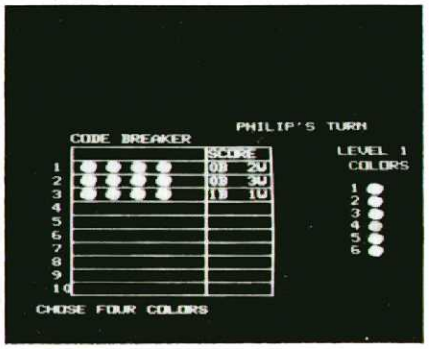


The programme is written in Machine Code so it draws the maps very rapidly. Your geography can be tested on any one of the following areas:

North Island N.Z. North America
South Island N.Z. South America
Australia and the Pacific Region
The British Isles

SOFTWARE REVIEW

CODE BREAKER

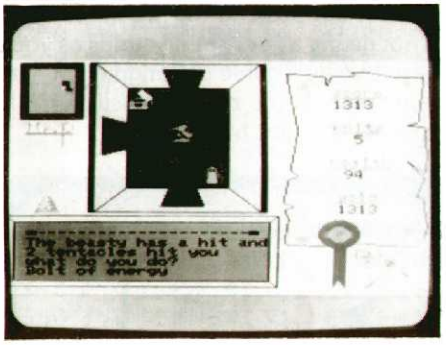


Remember good old mastermind? Well its now available for the Sega. The game has 4 different levels of difficulty making it quite hard to work out the 6 colour combination. A good solid challenging game.

THE HOUSE

BY MICHAEL HOWARD

This is the Sega Graphic adventure game (and what great graphics!)

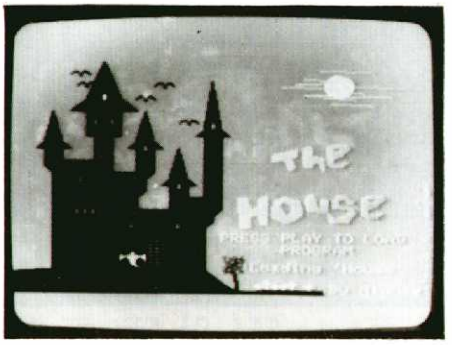


The object of the game is that you, Siddy Superspook, have to kill Vanessa the Vampire, but to do this you must find a crucifix, and find Vanessa. Unfortunately as there is an international shortage of crucifixes, there is only 1 in "The House," but you are given an occasional clue to tell you where its lying.

The game is made even harder as there are also horrible monsters in "The House." These are called purple people eaters, and have 8 tenticles, and tend to be very nasty! It's part of your job to kill the PPE'S (unless they kill you!!!)

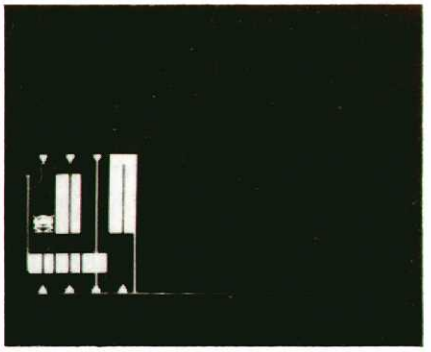
As you go from room to room searching for the crucifix, you may also come across chests of treasure and healing potions which help you regain your strength after battle with the PPE'S.

This game makes full use of the SC3000's arcade quality graphic and sound, and as the program is written in Machine Code it is very fast moving with smooth graphic movement.



TRAMPMAN

A highly original graphic game with full colour and sound. You control one of the legendary and lovable Binklehops of Bolthazar which you must manoeuvre, first through a lazer field and then bounce your way through a swarm of Bumble Bees by using eratically placed trampolines. You must then shoot another swarm of Bees and climb to safety. There are 5 more progressively difficult screens, and points are awarded as you move through the



screen and bonus points on completion. You have three lives and gain one on level completion. Features include title page, high score, and full sound effects. Joystick only. From the author of Mars Mobile.

BOXING

I have played a few boxing games on computers but this beats all. It is incredibly realistic with brilliant animation and crowds roaring in the background.

You have to see this game in action to believe it.

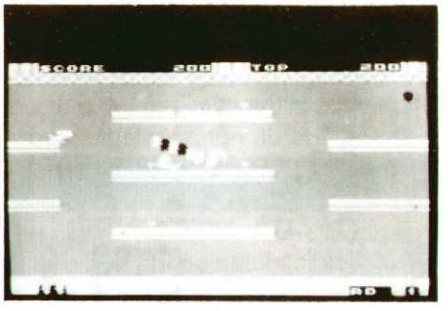
It has 5 different levels and a two player aption for a good slog out with a mate!!

You can throw a jab, use an uppercut or the good old straight punch to wittle your opponent down and knock him out. (The crowd goes wild) and the player that is knocked out gets a black eye!) Each round lasts 1½ minutes and it requires a lot of skill to be still standing after 10 rounds.

All in all an excellent game. Well worth the money.

FLICKY

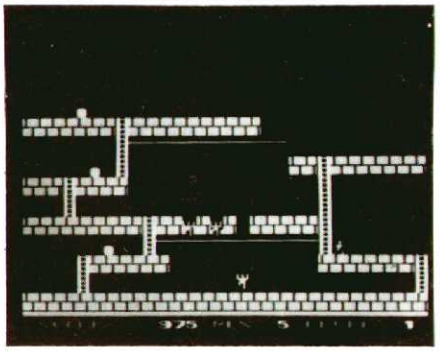
Flicky is a little blue duck who has to gather up ducklings and take them to safety at her "nest." Saucers, mugs, teacups, kettles. Gathering up guns and other objects and keeping hold of 8 ducklings is hard enough but Flicky has to contend with dreadful cats. These cats do their best to kill Flicky but don't worry she can throw hammers at them.



This game has 40 different screens with the best scrolling effects I have ever seen. It has many bonus screens and brilliant colourful graphics and arcade excellent sound. A very addictive and challenging game.

SOFTWARE REVIEW
continued

LODE RUNNER

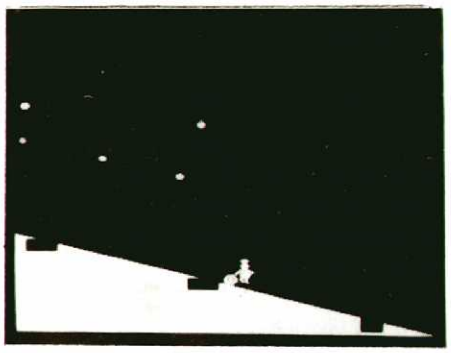


This is definitely my favourite game with 80 screens of fast action. It is brilliant. You even have the option of designing your own screens and saving on tape. You can flip between screens at will and increase the number of lives you have!

The idea of the game is to run around climbing ladders and ropes collecting the barrels. As the game progresses there are more and more barrels as well as death traps. You have to watch out for the guards as well but you can dig traps for them (being careful you do not fall in them yourself).

This will be a top seller for sure.

LUMBERJACK



You play the part of a tired Lumberjack trying to make his way home late one night after a heavy session down at the pub with the boys. Dodging the logs is not as easy as you thought it would be, especially as the levels keep on changing and getting harder. You start with three lives and gain one on completion of each level, and bonus points on completion. Features include full sound effects, title page and high score. Joystick only. A fun game that will keep you amused for hours. From the author of Mars Mobile.

SEGA®

GRAPHIC DESIGNER

Feedback from readers and users indicates that SEGA users are generally a lot more interested in using their computers for applications programs than originally anticipated by GRANDSTAND. Those who masterminded the design are not too surprised many owners are quickly enticed into wanting to find out more about the machine and its capabilities — the designers all understand what appeals to a hacker, and the SEGA hardware and software is rich with nooks and crannies just waiting to be discovered.

Programmers who start to write for the machine can fall into these enticing and carefully set traps, and may end up writing a complete 'symphony' when the original intention was simply to bash out a jaunty tune. One such victim is Ian Nicholson. Before getting stuck into some education design work, Ian thought he would run up a few utilities to make the development of the screens a little easier — amongst which is Sega Graphic Designer. What transpired is probably one of the most feature packed and accomplished drawing and illustration program yet, devised for the Sega.

The Graphic Designer is a machine code utility that provides all the tools you could possibly want to design a screen display for the SC3000. It also provides some you might never have thought of — but once discovered will become an integral part of the artist's equipment.

At the time of writing, the finishing touches are being completed — and by the time you read this, the program is likely to be just available.

The facilities available are:

Colour set selection
Pixel plotting in any colour
Line drawing
Circle drawing
Area fill
Text printing at any pixel location
Plotting with the graphic character
Complete erasing of last operation
Save/load screen and graphics characters

In all, operation can seem daunting for the newcomer, however, intuition takes over fairly rapidly and you will only need to refer to the manual for the tricky bits.

What all this adds up to is best illustrated by a picture or two. It is a shame we have not had the time to do anything more dramatic than those accompanying this piece — but it may inspire others to perform greater things — and it goes without saying that the best screens submitted each month will be published, and their originators will be rewarded in some suitable manner.

That strange whirring noise you can hear is Leonardo Da Vinci spinning in his grave. The program operates by maintaining a dummy screen image in memory (not screen memory) so that 'accidents' can be abandoned without doing any harm to the permanent image. The SEGA uses 16K for the screen display, which is why the loading screen for cassette based programs tend to be rather bland text-only affairs. However, with a disk drive loading a screen takes only a second or two, and with the Graphic Designer directly to the screen as required, and then uses these as the scenery for other programs and games. Imagine graphical adventures pulling in detailed screens as overlays from disk. The combination of a low cost disk system and the graphic precision of the SC3000 presents an unavoidable challenge to adventure publishers.

Designer unleashes a lot of creative talent. Most of those who tried it here ended up spending more time designing screens than playing games. There may not be a high score table — but the satisfaction available is of a totally different order.

DISECTION
Cavern Lander

This is a simple but challenging game. The objective of the game is to get through the cavern avoiding the stalagmites. Every cavern is different:

LINE 1Ø This creates the pattern for our space ship (see section on sprites in issue 2).

LINE 2Ø This sets the variables to S,G,L,Y & Y to Ø.

LINE 3Ø This puts us on to the graphics screen.

LINE 4Ø This changes the colour of our screen and border colour to blue and then clears the screen.

LINE 5Ø By using the LINE BF statement we fill in the screen with colour 1 (BLACK).

LINE 6Ø Sets G to a random number between Ø and 1.

LINE 7Ø The a FOR NEXT loops for I and T when drawing up. V is also set to a random number. This is the length of the stalagmites.

LINE 8Ø G which is the scope of the floor is added to 5 to give an increasing scope or climb to the cavern floor using the bline statement.

LINE 9Ø This actually digs into our covered screen and carves a cavern in random pattern.

LINE 1ØØ Next T tells Sega to go through the loop 25 times.

LINE 11Ø Asks if G<Ø, if G is then, it sets G to a new negative number, so as to change the direction of the cavern. Floor slope.

LINE 12Ø It has checked if G is already < Ø. If it is not it bypasses 11Ø and sets G to a positive number ie change direction or cavern floor slope.

LINE 13Ø Tell I to begin its loop.

LINE 14Ø This is the beginning of our joystick loop. Checks to see if the joystick is moved and makes a sound.

LINE 150 Says to GOTO 14Ø until we use the joystick.

LINE 16Ø Sets the position variables for our craft (A=5$ALONG) ie change direction for cavern floor (B=96=DOWN).

LINES 17Ø. 24Ø Corresponding to the directions of the joystick it will go to a line which moves our sprite in that direction. Then makes a rocket sound.

LINE 25Ø Add .5 to Y so we have gravitational pull.

LINE 26Ø 27Ø adds X to A and Y to B so we have drift.

LINE 28Ø Actually puts the sprite at location A by B.

LINE 29Ø By using this VPEK you can check if you are hitting anything on the screen accept sprites. If we are hitting something it will go to Line 32Ø, which is our game over routine.

LINE 3ØØ Checks if A (our X axis location) is greater than 240. In other words if you have succeeded.

LINE 31Ø Tells us to go to Line 16Ø which is the beginning of our joystick loop.

LINES 32Ø 33Ø Makes a cracking sound when you have hit a wall.

LINE 34Ø Turns the sound off.

LINE 35Ø Puts on to the text screen and clears it.

LINE 36Ø Prints up "you died", because you did.

LINE 37Ø Another for-next loop which just stops things for a while.

LINE 38Ø The beginning of the congratulations routines.

LINE 39Ø Congratulates you!

LINE 4ØØ Prints up at the bottom of the screen, to press fire for another go.

LINE 41Ø Checks if you are pressing your fire button. If you are then GOTO 1Ø which starts again.

LINE 42Ø Sets C to a random number and changes the colour accordingly. We then say keep changing until you press fire.

```
10 PATTERNS#0,"081C367F7F2A1400"
20 S=0:G=0:L=0:X=0:Y=0
30 SCREEN2,2
40 COLOR4,4,,4:CLS
50 LINE(0,0)-(250,191),1,BF
60 G=RND(1)*1
70 FORI=1TO10:FORT=1TO25:L=L+1:V=RND(1)*16
80 C=RND(1)*8:S=S+G
90 BLINE(L,96-S+24)-(L,96-V-S)
100 NEXTT
110 IFG>0THENG=RND(1)*1*-1:GOTO130
120 G=RND(1)*1
130 NEXTI
140 SOUND4,3,15:IFSTICK(1)>0THEN160
150 GOTO140
160 A=5:B=96:SPRITE0,(A,B),0,15
170 ONSTICK(1)+1GOTO260,180,190,200,210,220,230,240,250
180 SOUND3,900,0:Y=Y-2:GOTO260
190 SOUND3,900,0:X=X+1:Y=Y-1:GOTO260
200 SOUND3,300,0:X=X+1:GOTO260
210 SOUND3,300,0:X=X+1:Y=Y+1:GOTO260
220 SOUND3,200,0:Y=Y+1:GOTO260
230 SOUND3,300,0:X=X-1:Y=Y+1:GOTO260
240 SOUND3,300,0:X=X-1:GOTO260
250 SOUND3,900,0:X=X-1:Y=Y-1
260 Y=Y+.5
270 A=A+X
280 B=B+Y
290 SPRITE0,(A,B)
300 IFVPEEK(INT((B+4)/8)*256+INT((A+4)/8)*8+BMOD8)>0THEN330
310 IFA>240THEN390
320 GOTO170
330 FORI=500TO110:SOUND3,I,5
340 SOUND4,3,15:NEXT
350 SOUND0
360 SCREEN1,1:CLS
370 PRINT"              YOU DIED"
380 FORI=1TO1000:NEXTI:GOTO410
390 SCREEN1,1:CLS
400 PRINT"CONGRATULATIONS YOU MADE IT ALIVE !!!"
410 CURSOR0,20:PRINT"    PRESS FIRE TO START AGAIN"
420 IFSTRIG(1)>0THEN10
430 C=RND(1)*16:COLORC:GOTO410
```

SPRITE MOVER

BY JOHN DAVIS

This program automatically moves all sprites according to the information held in the movement table, (in the example this is positioned at &HA1ØØ, although this can be moved by changing the relevant line). The routine itself has been written so as to be position independent, and can therefore be loaded anywhere in R.A.M.

To use, load the machine code, set up the movement table, (SEE NOTE), set up the sprites on the screen, and then each time you want the sprites to move, CALL the start address. (The more often, the better)

SPRITE MOVEMENT TABLE: this is a block of memory (&HØØ bytes), referred to by the routine and which specifies how each of the 32 sprites move.
The table is made up of 32 blocks (one per sprite), each of four bytes.

Each block is in the form:-

RATE; 1-255 specifies how often the sprite will move. (1 = fast to 255 = slow)

COUNTER; this is used within the routine, and should initially be set to one.

Y; specifies how far the sprite will move in the y-dimension each time. (This is in two's complement form, i.e. Ø-127 will give a positive movement while 128-255 will give a negative)
X; as for Y but in the X-direction.

```
11,Ø4,ØØ        LD DE,&HØØØ4
21,ØØ,3B        LD HL,&H3BØØ
DD,21,ØØ,A1     LD IX,&HA1ØØ    ;start of movement table,
                                   change as necessary
 ØE,DØ          LD C,&HDØ
 Ø6,2Ø          LD B,&H2Ø
F3              DI
DD,35,Ø1DEC     (IX + 1)         ;decrement counter
2Ø,28           JRNZ &H28        ;move sprite if
                                 counter = Ø
DD,7E,ØØ        LD A,(IX + Ø)
DD, 77,Ø1       LD (IX + 1),A    ;reset counter
CD,32,2C        CALL &H2C32      ;move in y dimension
DB,BE           IN &HBE,A
DD,86,Ø2        ADD A,(IX + 2)
B9              CP A,C           ;correct for jitter when
                                   y = &HDØ
2Ø,Ø3           JRNZ &HØ3
DD,86,Ø2        ADD A,(IX + 2)
CD,44,2C        CALL &H2C44
D3,BE           OUT &HBE,A
23              INC HL
CD,32,2C        CALL &H2C32      ;move in x dimension
DB,BE           IN &HBE,A
DD,86,Ø3        ADD A,(IX + 3)
CD,44,2C        CALL &H2C44
D3,BE           OUT &HBE,A
2B              DEC HL
19              ADD HL,DE        ;move to next sprite
DD,19           ADD IX,DE        ;move to next block
1Ø,CE           DJNZ &HCE        ;repeat X32
C9              RET
```

```
10 DATA 11,04,00,21,00,3B,DD,21,00,A1,0E,D0,06,20,F3,DD,35,01,20,28,DD,7E,00,DD,
77,01,CD,32,2C,DB,BE,DD,86,02,B9,20,03,DD,86,02,CD,44,2C,D3,BE,23,CD,32,2C,DB,BE
,DD,86,03,CD,44,2C,D3,BE,2B,19,DD,19,10,CE,C9
20 REM SET UP MACHINE CODE AT &HA000
30 RESTORE 10:FOR A=&HA000 TO &HA041:READ A$:POKE A,VAL("&H"+A$):NEXT
40 REM SET UP MOVEMENT TABLE AT &HF100
50 FOR A=&HA100 TO &HA180 STEP 4
60 POKE A+0,1+INT(5*RND(1)):REM RATE OF MOVEMENT
70 POKE A+1,1:REM COUNTER
80 POKE A+2,(253+INT(10*RND(1)))MOD 256:REM Y
90 POKE A+3,(253+INT(10*RND(1)))MOD 256:REM X
100 NEXT
110 REM SET UP SPRITES
120 SCREEN 2,2:COLOR1,14,,15:MAG 0:PATTERN S#0,"FF818181818181FF"
130 FOR S=0 TO 31
140 SPRITE S,(120,90),0,INT(16*RND(1))
150 NEXT
160 REM MOVE SPRITES
170 CALL &HA000:GOTO 170
180 END
```

Graphics and Trigonometry

by Michael Howard

From the Wow-This-Micro-Has-Brilliant-Graphics Department comes an army of short programs for creating silly patterns. They all contain one element in common, they use cosine, sin, and tangent (Don't shout "Boo-hiss!"). If you want better coverage of COS, SIN and TAN then read pp. 125-128 of the operators manual.

A lot of people think that trigonometry is difficult, well it's not, and hopefully these small programs will help put matters straight! (Trigonometry is the use of COS, SIN and TAN, it probably has Greek origins and probably means boring!) (Acutally it is Greek and Tri means 3 sides of a triangle, a triangle is the basis of the science). Well enough of this, let's get on with the programs.

Program 1

```
10 SCREEN 2,2:CLS:PSET(128,88),8
20 FORA=0TO359STEP2
30 X=SIN(A/PI):Y=COS(A/PI)
40 X=X*X*Y*PI:Y=Y*Y*X*PI
50 LINE-(128+(X*45),88+(Y*45))
60 NEXTA
```

Prog. 1 creates a sort of wonky figure of eight in an unusual manner. The main work is done in line 3Ø, generally where there's a SIN, there's a COS! And 9 times our of 1Ø there's a PI (PI = 3.1415926536. . . . .) You may be wondering why in line 3Ø A is divided by PI. Well the reason is straightforward. Computers don't work in degrees (e.g., 90°, 127°) they work in radians (Another silly measurement that keeps mathematicians happy!). By dividing PI, radians are created from degrees (This is only rough, but it works!). Also another thing that is necessary to know, is that when using SIN and COS, no matter what number is used (e.g., SIN (23), COS(-42) etc) the result is always in the range -1 to 1. So the result must be multiplied by a fairly large number so that the function can be represented on a graph, this is what happens in line 5Ø.

Program 2

```
10 SCREEN 2,2:CLS
20 DP=2*PI/183:XI=128:YI=96:RA=90
30 FORP=0TO2*PISTEPDP
40 X=RA*COS(P):Y=RA*SIN(P)
50 X=XI+X:Y=YI+Y
60 LINE(128,96)-(X,Y),8
70 NEXTP
```

Prog. 2 creates a circle with many radii thus form a "dazzling" effect. You may have noticed that the main loop (started in line 3Ø) has a PI in it. Why? Well look at line 4Ø notice the COS and SIN, there is no PI in that! The effect of this is that P is already in radians! (If you haven't already guessed radians are expressed in terms of PI, unlike degrees which are in terms of 1Ø). RA is the radius of the circle, XI and YI the centre and DP the distance between points on the circumference (in radians).

Program 3

```
10 SCREEN 2,2:CLS
20 FORX=-100TO100STEP2
30 R=10:J=0:K=1
40 V=R*INT(SQR(90-ABS(X)/R))
50 FORY=VTO-VSTEP-R
60 Z=INT(80+30*SIN((SQR(X*X+Y*Y))/12)-.7*Y)
70 IFZ<JTHEN90
80 J=Z:PSET(128+X,191-Z-15):K=0
90 NEXTY,X
```

Prog. 3 produces a 3D graph, which looks quite spectacular! The only problem is it takes about 5 minutes to create! The crux of the program is line 6Ø, doesn't it look awful?! I'm not going to explain how it works because it is quite complicated!, and I'd only get muddled up explaining it!

Program 4

```
10 H=100:L=20:B=15:AR=-45:AL=45:S=5
20 SCREEN 2,2:CLS
30 FORA=ALTOARSTEP-S
40 F=PI/180
50 X(1)=130-H*TAN(A*F)
60 Y(1)=H
70 X(2)=X(1)+(L*COS(A*F))
80 Y(2)=Y(1)+(L*SIN(A*F))
90 X(4)=X(1)+(B*COS((90+A)*F))
100 Y(4)=Y(1)+(B*SIN((90+A)*F))
110 X(3)=X(4)+(X(2)-X(1))
120 Y(3)=Y(4)+(Y(2)-Y(1))
130 PSET(X(1),Y(1)),8:FORI=2TO4:
    LINE-(X(I),Y(I)):NEXT:
    LINE-(X(1),Y(1))
140 NEXTA
```

Prog. 4 forms a box or rectangle which moves across the screen slowly revolving. Line 5Ø determines the path along the screen. You can alter the TAN function with COS or SIN. To do so line 1Ø must be altered.

Viz:

```
10 H=100:L=20:B=15:AR=-180:AL=180:S=5

50 X(1)=130-H*COS(A*F)

50 X(1)=130-H*SIN(A*F)
```

L is the length of the box, B the breadth, AR it's finished angle, AL its starting angle and 5 the step. Try altering them. Also instead of a box, alter line 13Ø to

```
130 PSET(X(1),Y(1)),8:FORI=3TO4:LINE-
(X(I),Y(I)):NEXT:LINE-(X(1),Y(1))
```

to get triangles!

Program 5

```
10 SCREEN 2,2:CLS
20 PSET(128,96),8
30 FOR TH=0TO8*PI STEP PI/40
40 R=TH/5
50 X=R*COS(TH):Y=R*SIN(TH):X=X*5:Y=Y*5:X=X+128:Y=Y+96
60 LINE-(X,Y)
70 NEXT TH
```

Prog. 5 forms an ever increasing spiral. If you're interested try altering line 4Ø and 3Ø e.g.,

```
30 FOR TH=0TO16*PI STEP PI/40
```

would make the spiral larger.

```
40 R=TH/20
```

would alter the distance between spirals.

Another alteration which makes "flowers" is:

```
20 PSET(198,96),8
30 FOR TH=0TO2*PI STEP PI/100
40 R=1+COS(8*TH)
50 X=R*COS(TH):Y=R*SIN(TH):X=X*35:Y=Y*35:X=X+128:Y=Y+96
```

Try altering line 3Ø and 4Ø e.g.,

```
30 FOR TH=0TO2*PI STEP PI/500

40 R=1+COS(32*TH)
```

The reason for the step change is that it makes the "flower" smoother, and line 4Ø creates more "petals."

Program 6

```
10 SCREEN 2,2:CLS:PSET(12,96),8
20 FORI=0TOPI*51STEPPI/5
30 X=I*(4*PI/255):X=X*10
40 Y=SIN(X)*EXP(-X/14):Y=Y*100:Y=Y+96
50 LINE-((X*3)+12,Y)
60 NEXT I
```

Prog. 6 draws a "damped sin wave," the damping is created by the EXP function in line 4Ø try altering 4Ø to increase or decrease the damping

```
40 Y=SIN(X)*EXP(-X/10):Y=Y*100:Y=Y+96
```

Program 7

```
10 SCREEN 2,2:CLS:PSET(218,96),8
20 FORZ=0TO20*PI STEP PI/10
30 X=COS(Z/10):X=X*90:X=X+128
40 Y=SIN(Z):Y=Y*50:Y=Y+96
50 LINE-(X,Y)
60 NEXTZ
```

Prog. 7 is hard to explain, so just run it!

Program 8

```
10 SCREEN 2,2:CLS:PSET(202,96):V=9
20 FOR Z=0 TO PI*2 STEP PI/10
30 X=.125*COS(V*Z):X=X*590:X=X+128
40 Y=.125*SIN(V*Z):Y=Y*590:Y=Y+96
50 LINE-(X,Y)
60 NEXT Z
```

Prog. 8 creates multi pointed stars. Try different values of V e.g.,

Also try the following:

```
20 FORZ=0TO PI*4 STEP PI/10
```

and

```
20 FORZ=0TO PI*8 STEP PI/10
V=5.5 OR V=3.5 OR V=8.5 OR V=6.5
V=8.25 OR V=4.25 OR V=10.25 OR V=5.25 OR V=9.25
```

Oh well folks, that's the lot! I hope that you try some of your own little experiements and have realised that trigonometry is not really that awful!

BALUNE

You are in control of a Balloon that you must guide across the rugged terrain of the 6 different levels in the game. However, you must be very careful as should your 'Balune' (this is how the Japanese spell (?) Balloon) touch a wall it will burst. You have 3 balloons, use the cursor arrows to move the swinging balloon. This is rather a long program. Most of this is actually the 6 different screens (Lines 1000-2130). If you wish to leave any of the screens out to make the program quicker to type it just change line 90 so that the GOSUB is followed by as many parameters as you have typed in screens e.g. 4 screens (Lines 1000-1730).

90 ON RU GOSUB 1000, 1200, 1400, 1600

```
10 SCREEN2,2:COLOR,1,,1:CLS
20 PRINTCHR$(17):GOSUB600:HI=500:B=3:RU=1:SC=0:F=1:FF=1
30 COLOR15:CURSOR35,160:PRINT"GAME PLAY A Y/N?"
40 A$=INKEY$:IFA$="Y"THEN70
50 IFA$="N"THENCLS:END
60 GOTO 30
70 PRINTCHR$(16):SCREEN1,1:CLS:COLOR,1:GOSUB240:TIME$="00:00:00"
75 REM PATAREN PRINT
90 ON RU GOSUB 1000,1200,1400,1600,1800,2000
100 CURSOR30,1:PRINT" HI-SC";
101 CURSOR30,2:PRINTHI
102 CURSOR30,5:PRINT" SCORE"
103 CURSOR30,6:PRINTSC
104 CURSOR31,13:PRINT"LEVEL"
105 CURSOR32,15:PRINTRU
106 CURSOR31,20:PRINT"BALUNE"
107 CURSOR32,22:PRINTB
110 REM MAIN LOOP
120 GOSUB232
130 A$=INKEY$:IFA$=CHR$(28)ORA$=CHR$(29)ORA$=CHR$(30)ORA$=CHR$(31)THENA=ASC(A$)-27:GOTO200
150 M=M+E:IFM=3ORM=1THENE=-E
160 IFVPEEK(AD+V(M))<>32THEN300
170 VPOKEAD+V(M),AA(0,M):VPOKEAD+40,AA(1,M)
180 VPOKEAD+V(M),32
185 IF15523<ADANDAD<15532THEN400
187 IF5<RUANDRU9THENONRU-5GOSUB4000,4100,4200,4300
190 A=0:GOTO 130
200 VPOKEAD+40,32:X=X+X(A):Y=Y+Y(A):GOSUB230
210 IFVPEEK(AD+40)<>32THENX=X-X(A):Y=Y-Y(A):GOSUB230
220 BEEP1:BEEP0:GOTO150
225 REM SABU
230 AD=Y*40+X+&H3C00:RETURN
232 X=28:Y=21:GOSUB230:M=1:E=1:RETURN
234 FORI=0TO700:NEXT:RETURN
240 RESTORE900:FORI=1TO4:READAA(0,I),AA(1,I),X(I),Y(I),V(I)
250 DATA235,148,1,0,-1,235,128,-1,0,0,235,147,0,-1,1,0,0,0,1,0
260 PR$(1)=" GREAT!!!":PR$(2)="VERY GOOD":PR$(3)="GOOD LUCK  ":PR$(4)="GOOD WORK! "
270 PR$(5)="ARE YOU OK?":PR$(6)="DON'TDIE NOW":PR$(7)="YOU'RE DOING WELL!":PR$(8)="NOW TRY THE
290 PATTERNC#42,"FFFFFFFFFFFFFF00":NEXT:RETURN                                    NEXT ONE"
300 RESTORE910:REM EXPLOSION
310 FORI=0TO1:READP:SOUND5,2,7:SOUND1,P,15:NEXT:SOUND0
320 B=B-1:IFB=0THENCOLOR1,15:CURSOR5,12:PRINT"GAME OVER":GOSUB234:GOTO10
330 GOTO70
390 REM SCRORE SC
400 VPOKEAD+V(M),32:VPOKEAD+40,32
410 VPOKEAD-40,AA(0,2):VPOKEAD,AA(1,2)
420 SC=SC+1000-(VAL(MID$(TIME$,5,1))*100+VAL(MID$(TIME$,7,2))*5)
430 RU=RU+1:IFSC>HITHENHI=SC
440 RESTORE470:FORI=0TO3:READS
450 FORU=0TO5:SOUND1,S,15:NEXT:SOUND0:NEXT
455 GOSUB234:CLS:COLOR15:CURSOR5,13:PRINTPR$(RU)
456 GOSUB234
460 IFA$=0THENIFSC>5000THENB=B+1
465 GOTO70
470 DATA 131,139,147,165
600 PATTERNS#0,"070F1F3F3F3F3F3F"
610 PATTERNS#1,"7F3F1F0F0F0F0300"
620 PATTERNS#2,"C0E8F8F8F8F8F8F8"
630 PATTERNS#3,"F8F0F0E0E0C00000"
640 PATTERNS#4,"2040800000000000"
650 MAG1:Z=2:ZZ=2:VV=0:AA$="BALUNE"
660 FOR I=30TO230 STEP40
665 VV=VV+1
670 FORY=0TO7
675 BLINE(I+Z+5,95)-(I+5,104)
680 Z=Z+ZZ:IFZ=0ORZ=8THENZZ=-ZZ
690 SPRITE 3,(I+Z,80),0,6
695 SPRITE 2,(I+Z+6,90),4,15
697 LINE(I+Z+5,95)-(I+5,104),15
700 NEXT:BLINE(I+Z+5,95)-(I+5,104):CURSORI+3,85:PRINTMID$(AA$,VV,1):NEXT:SPRITE3,(I,80),0.0:RETURN
900 DATA235,148,1,0,-1,235,128,-1,0,0,235,147,0,-1,1,0,0,0,1,0
```

```
910 DATA120,1600
1000 CLS:COLOR2
1010 PRINT"******************************"
1015 PRINT"*          ********************"
1020 PRINT"* GOAL       **  ***************"
1025 PRINT"**            **   *** *********"
1030 PRINT"**                  **********"
1035 PRINT"******                 *******"
1040 PRINT"********      *****     *******"
1045 PRINT"*********    ***       *********"
1050 PRINT"********   *****     **********"
1055 PRINT"*****      ******         ******"
1060 PRINT"***          ***              *"
1065 PRINT"**   ***    * ***       ***   *"
1070 PRINT"*           * ***       ***   *"
1075 PRINT"**    *       ***        **   *"
1080 PRINT"*                            **"
1085 PRINT"**         *                 ***"
1090 PRINT"***         **          ********"
1095 PRINT"*****      ****       **********"
1100 PRINT"*****    *******     *********"
1105 PRINT"****   ********   ************"
1110 PRINT"**                           **"
1115 PRINT"***                          **"
1120 PRINT"****  *************          **"
1125 PRINT"******************************";
1130 RETURN
1200 CLS:COLOR2
1210 PRINT"******************************"
1215 PRINT"*          ********************"
1220 PRINT"* GOAL     ********************"
1225 PRINT"*            ********       ****"
1230 PRINT"**           ******          ***"
1235 PRINT"**   *      ******      **    ***"
1240 PRINT"*   ***   ********     ***    ***"
1245 PRINT"*    **    ******      **     ***"
1250 PRINT"**         ******     **      ***"
1255 PRINT"***         ***      ***       **"
1260 PRINT"***                  ***       **"
1265 PRINT"*****              **          **"
1270 PRINT"*********       ******         **"
1275 PRINT"************************       **"
1280 PRINT"****     *******       ****    **"
1285 PRINT"**         ****          ***    **"
1290 PRINT"*          ***             *    **"
1295 PRINT"*   ***     ***    ***     **   **"
1300 PRINT"*   ****           ****         **"
1305 PRINT"*   ****           *****        **"
1310 PRINT"*   ***************************  **"
1315 PRINT"*                               *"
1320 PRINT"*                               *"
1325 PRINT"******************************";
1330 RETURN

1400 CLS:COLOR2
1410 PRINT"******************************"
1415 PRINT"*        ********************"
1420 PRINT"* GOAL    ********************"
1425 PRINT"**          *********       ****"
1430 PRINT"****      ****              ****"
1435 PRINT"******      **        *       ***"
1440 PRINT"*****      ***     *******    ***"
1445 PRINT"***       ****      *****     ***"
1450 PRINT"**      **                    ****"
1455 PRINT"*        *             **     ****"
1460 PRINT"***      *           ****     ****"
1465 PRINT"***                ******     ****"
1470 PRINT"*****            *******     ****"
1475 PRINT"*****    *       *****       *****"
1480 PRINT"****    ***     ******       *****"
1485 PRINT"***    *****    ******      ******"
1490 PRINT"***    ******  ****      ********"
1495 PRINT"**      ***     ****     ********"
1500 PRINT"***      **       **       ******"
1505 PRINT"***       *        *        *****"
1510 PRINT"*                             ****"
1515 PRINT"*                               *"
1520 PRINT"*       ****     ****          **"
1525 PRINT"******************************";
1530 RETURN
1600 CLS:COLOR2
1610 PRINT"******************************"
1615 PRINT"*        *      ***        ******"
1620 PRINT"* GOAL   *                    ****"
1625 PRINT"*                      ***       **"
1630 PRINT"*  **     **    **   ***       **"
1635 PRINT"*  ***    **          ***       **"
1640 PRINT"*    ***   *           ****     ***"
1645 PRINT"*     ***    *  **     ****    ****"
1650 PRINT"*      **       ***     **     ****"
1655 PRINT"**       **      **     **       **"
1660 PRINT"***       *               *       **"
1665 PRINT"***      *****        ******     **"
1670 PRINT"**        ****     ********      **"
1675 PRINT"*          ***                    **"
1680 PRINT"*         *****                   **"
1685 PRINT"*               **     *****       *"
1690 PRINT"*    **         *     ********    *"
1695 PRINT"*       **      **************   *"
1700 PRINT"**    ****    ***     ********   *"
1705 PRINT"**     **     *         ******    *"
1710 PRINT"***     **               *****    *"
1715 PRINT"****                **            *"
1720 PRINT"*****            *****            *"
1725 PRINT"******************************";
1730 RETURN
1800 CLS:COLOR2
1810 PRINT"******************************"
1815 PRINT"*         ******                ***"
1820 PRINT"* GOAL     ***                   **"
1825 PRINT"**         ***    *****    *     **"
1830 PRINT"***     ******  *******  *      **"
1835 PRINT"***       **  *    *****        ***"
1840 PRINT"***   ******      *****        ***"
1845 PRINT"**    *******     ******        **"
1850 PRINT"*    *********    ***        *****"
1855 PRINT"**    *******    *****     ******"
1860 PRINT"***   ******       ***    *********"
1865 PRINT"**     *****        **    *********"
1870 PRINT"**       *          **    *********"
1875 PRINT"*    *    **   **    **     ********"
1880 PRINT"*   **     *   ***   **        *****"
1885 PRINT"*   ***    *    **   ****      *****"
1890 PRINT"*    *    *     **    ***      *****"
1895 PRINT"*         *     *     **         *"
1900 PRINT"***         *     *    ****        *"
1905 PRINT"****          **     ***         **"
1910 PRINT"****                        *      *"
1915 PRINT"****          *                **   *"
1920 PRINT"****            **          ****    *"
1925 PRINT"******************************";
1930 RETURN
2000 CLS:COLOR2
2010 PRINT"******************************"
2015 PRINT"*          ********************"
2020 PRINT"* GOAL   **       *****         **"
2025 PRINT"**                              **"
2030 PRINT"***           ***          **     **"
2035 PRINT"**********       ******       ***"
2040 PRINT"**              *  *******     ****"
2045 PRINT"***                             ****"
2050 PRINT"****                            ****"
2055 PRINT"**       ************************"
2060 PRINT"**          *    ****            **"
2065 PRINT"**                              **"
2070 PRINT"**           **          ****    **"
2075 PRINT"***************************     *"
2080 PRINT"**          *                 **    *"
2085 PRINT"**                              *"
2090 PRINT"*             **  **            **"
2095 PRINT"*    ***************************"
2100 PRINT"*    **                         *"
2105 PRINT"*                               *"
2110 PRINT"**          *****         ****    *"
2115 PRINT"***                             *"
2120 PRINT"***                  ****       *"
2125 PRINT"******************************";
2130 RETURN
3100 GOTO3100
4000 VPOKE15724+F,32
4005 F=F+FF:IFF=0ORF=4THENFF=-FF
4010 VPOKE15724+F,42:RETURN
```

GRANDSTAND LEISURE LIMITED

If you are having any problems obtaining any of the pieces of software please contact Granstand Leisure enclosing a cheque or postal order.

SOFTWARE

CARTRIDGES		PRICE
☐ BASIC LEVEL II		$39.95
☐ BASIC LEVEL IIIA		$130
☐ BASIC LEVEL IIIB		$230
☐ MUSIC		$39.95
☐ N/SUB		$39.95
☐ SAFARI HUNTING		$39.95
☐ BORDER LINE		$39.95
☐ CONGO BONGO		$39.95
☐ YAMATO		$39.95
☐ STAR JACKER		$39.95
☐ CHAMPION TENNIS		$39.95
☐ MONACO GP		$39.95
☐ CHAMPION BASEBALL		$39.95
☐ SINBAD MYSTERY		$39.95
☐ VIDEO FLIPPER		$39.95
☐ PACAR		$39.95
☐ POP FLAMER		$39.95
☐ CHAMPION GOLF		$49.95
☐ EXERION		$49.95
☐ SAFARI RACE	DUE JANUARY	$49.95
☐ LODE RUNNER	DUE JANUARY	$49.95
☐ BOXING	DUE JANUARY	$54.95
☐ ORGUS	DUE FEBRUARY	$49.95
☐ FLICKY	DUE FEBRUARY	$59.95

HOME/PERSONAL/FINANCE CASSETTES	PRICE
☐ FILE SYSTEM	$19.95
☐ CHEQUE BOOK RECONCILIATION	$19.95
☐ LOAN & MORTGAGE CALCULATION	$19.95
☐ ACCOUNTS RECEIVABLE	$30.00
☐ ACCOUNTS PAYABLE	$30.00
☐ 32K WORD PROCESSOR	$30.00
☐ PERSONAL RECORD KEEPER	$30.00
☐ 16K EASY WRITER	$19.95

GAME CASSETTES	PRICE
☐ TOWERS OF HANOI	$19.95
☐ HANGMAN	$19.95
☐ CITY-LANDER	$19.95
☐ CUBE-IT	$30.00
☐ BUGALOO	$19.95
☐ LASER BLAST	$19.95
☐ MUNCHMAN	$30.00
☐ ENTERPRISE ESCAPE	$19.95
☐ MARSMOBILE	$19.95
☐ MARS ADVENTURER	$19.95
☐ EMPIRE	$19.95
☐ KALAH	$19.95
☐ CHESS	$19.95
☐ DEATH SATELLITE	$19.95
☐ BIT BYTE	$19.95
☐ THE HOUSE	DUE JAN. $30.00
☐ THE PYRAMID	DUE JAN. $19.95
☐ CODE BREAKER	DUE JAN. $19.95
☐ BACKGAMMON	DUE JAN. $19.95
☐ LUMBER JACK	DUE JAN. $19.95
☐ TRAMPMAN	DUE JAN. $19.95
☐ SEGA GAMES PACK I	DUE JAN $19.95

EDUCATIONAL CASSETTES	PRICE
☐ ROCKET MATHS	$19.95
☐ LEARNING ALPHABET	$19.95
☐ WATCH ME DRAW	$19.95
☐ SPELLING	$19.95
☐ ADDITION	$19.95
☐ SUBTRACTION	$19.95
☐ MULTIPLICATION	$19.95
☐ SPRITE EDITOR	$19.95
☐ MUSIC EDITOR	$19.95
☐ TYPING TUTOR	$19.95
☐ LEARNING TO COUNT	$19.95
☐ SHAPE & COLOUR QUIZ	$19.95
☐ MUSIC CARTRIDGE DEMONSTRATOR	$19.95
☐ SEGA GRAPHIC DESIGNER	$19.95
☐ GEOGRAPHY QUIZ I	DUE $19.95
☐ GEOGRAPHY QUIZ II	JANUARY $19.95
☐ MATHS HANG UP	$19.95

I enclose payment: ✓

☐ Visa ☐ Bankcard

Cardholder signature.....

Date card expires ....

Cheque ☐ Cheque No.....

Postal Order ☐ Postal note ☐

GRANDSTAND
Box 2353
AUCKLAND

CHANGE OF ADDRESS

Name .....

Old Address .....

.....

.....

New Address .....

.....

If you are changing your address please complete this coupon and return to:

Grandstand Sign
P.O. Box 2353
Auckland, NZ.

GLOSSARY

Accessory Devices - additional equipment which attaches to the computer and extends its functions and capabilities. Included are preprogrammed cartridges* and units which send, receive or store computer data, such as printers and disks. These are often called peripherals.

Array - A collection of numeric or string variables, arranged in a list or matrix for processing by the computer. Each element in an array is referenced by a subscript* describing its position in the list.

ASCII - The American Standard Code for Information Interchange, the code structure used internally in most personal computers to represent letters, numbers, and special characters.

BASIC - an easy-to-use popular programming language used in most personal computers. The word BASIC is an acronym for "Beginners All purpose Symbolic Instruction Code."

Baud - commonly used to refer to bits per second.

Binary - a number system based on two digits, 0 and 1. The internal language and operations of the computer are based on the binary system.

Branch - a departure from the sequential performance of program statements. An unconditional branch causes the computer to jump to a specified program line every time the branching statement is encountered. A conditional branch transfers program control based on the result of some arithmetic or logical operation.

Breakpoint - a point in the program specified by the STOP command where program execution can be suspended. During a breakpoint, you can perform operations to help you to locate program errors. Program execution can be resumed with a CONT command, unless editing took place while the program was stopped.

Bug - a hardware defect or programming error which causes the intended operation to be performed incorrectly.

Byte - a string binary* digits (bits) treated as a unit, often representing one data character*. The computer's memory capacity is often expressed as the number of bytes available. For example, a computer with 16K bytes of memory has about 16,000 bytes available for storing programs and data.

Cartridges - preprogrammed ROM* modules which are easily inserted in the SEGA computer to extend its capabilities.

Character - a letter, number, punctuation symbol, or special graphics symbol.

Command - an instruction which the computer performs immediately. Commands are entered with no preceding line number.

Concatenation - linking two or more strings* to make a longer string. The "+" is the concatenation operator.

Constant - a specific numeric or string* value. A numeric constant is any real number such as 1.2 or -9054. A string constant is any combination of up to 248 characters enclosed in quotes, such as "HELLO THERE" or "275 FIRST ST."

Cursor - a symbol which indicates where the next character* will appear on the screen when you press a key.

Data - basic elements of information which are processed or produced by the computer.

Default - a standard character or value which the computer assumes if certain specifications are omitted within a statement* or a program*.

Device - (see Accessory Devices)

Disk - a mass storage device capable of random and sequential access.

Display - (noun) the video screen; (verb) to cause characters to appear on the screen.

Execute - to run a program; to perform the task specified by a statement* or command*.

Exponent - a number indicating the power to which a number or expression* is to be raised; usually written at the right and above the number. For example, $2 = 2x2x2x2x2x2x2x2$. In SEGA BASIC the exponent is entered following the letter "E" in scientific notation*. For example, $2 = 2\ 8$; $1.3 \times 10 = 1.3E25$.

Expression - a combination of constants, variables, and operators which can be evaluated to a single result. Included are numeric, string, and relational expressions.

File - a collection of related data records stored on a device; also used interchangeably with device* for input/output equipment which cannot use multiple files, such as a line printer.

Function - a feature which allows you to specify as "single" operations a variety of procedures, each of which actually contains a number of steps; for example, a procedure to produce the square root via a simple reference name.

Graphics - visual constructions on the screen, such as graphs, patterns, and drawings, both stationary and animated. SEGA BASIC has build-in subprograms which provide easy to-use colour graphic capabilities.

Hardware - the various devices which comprise a computer system, including memory. the keyboard, the screen, disk drives, line printers, etc.

Hertz(HZ) - a unit of frequency. One Hertz = one cycle per second.

Hexadecimal - a base .16 number system using 16 symbols, 0.9 and A-F. It is used as a convenient "shorthand" way to express binary* code. For example, 1010 in binary = A in hexadecimal, 11111111 = FF. Hexadecimal is used in constructing patterns for graphics characters in the PATTERN subprogram.

Increment - a positive or negative value which consistently modifies a variable*.

Input - (noun) data* to be placed in computer memory; (verb) the process of transferring data into memory.

Input Line - the amount of data* which can be entered at one time. In SEGA BASIC, this is 248 characters.

Internal date-format - data* in the form used directly by the computer. Internal numeric data is 8 bytes* long plus 1 byte which specifies the length. The length for internal string data is one byte per character in the string* plus one length-byte.

Integer - a whole number, either positive, negative or zero.

I/O - Input/Output; usually refers to a device function. I/O is used for communication between the computer and other devices (e.g. keyboard, disk)

Iteration - the technique of repeating a group of program statements; one repetition of such a group. See Loop.

Line - see input line, print line, or program line.

Loop - a group of consecutive program lines which are repeatedly performed, usually a specified number of times.

Mantissa - the base number portion of a number expressed in scientific notation*. In 3.264E + 4, the mantissa is 3.264.

Mass Storage Device - an accessory device*, such as a cassette recorder or disk drive, which stores programs and/or data* for later use by the computer. This information is usually recorded in a format readable by the computer, not people.

Memory - see RAM, and ROM, and mass storage device.

Continued on Page 19

How To Program In Machine Code Language On The Sega SC3000

by Colin Smith

Binary, Decimal and Hexadecimal.

Binary Numbers (Base 2)

The simplest logic is the Binary Digit. It has only two values Ø or 1, this can be likened to an electric light bulb it is either OFF or ON (Ø or 1). This single Binary Digit is refered to as a 'bit'.

Decimal Numbers (Base 1Ø)

Decimal Numbers is the system we are taught at school and these have 1Ø possible values Ø to 9. If we require to count higher than nine we use a second digit placed to the left of the first number. This value represents groups of 1Øs and this practice is continued for groups of 1ØØs, 1ØØØs etc.
These can be written as powers of 1Øs

e.g., The number 795 = 1ØØ 1Ø 1
7 9 5

or (7 × 1ØØ) + (9 × 1Ø) = (5 × 1)

This practice can be used for binary digits to enable us to count higher than 1. In binary if we take that Ø = Ø and that 1 = 1, to count higher we will have to generate a carry to the next column.

e.g., To represent 2 we will have to add 1 + 1

```
  1
+ 1
----
 1Ø binary
```

1 plus 1 equals zero and carry 1

Therefore the binary figure 1Ø actually represents the decimal figure 2.

To represent 3 add 2 + 1 in binary.

```
  1Ø
+ 1
-----
 11 binary
```

To represent 4 add 3 + 1 in binary.

```
  11
+ 1
-----
 1ØØ binary
```

One plus one in the first column equals Ø carry 1

One plus one in the second column equals Ø carry 1

One plus zero in the third column equals 1.

So the figure 1ØØ in binary represents 4 in decimal. It is necessary to convert decimal numbers to binary numbers for the computer works in binary numbers.

To convert decimal to binary divide the decimal number by 2 and keep note of your remainders.

e.g., to convert 13 to binary.

Step 1: 13 ÷ 2 = 6 and 1 remainders.
Step 2: 6 ÷ 2 = 3 and Ø remainders.
Step 3: 3 ÷ 2 = 1 and 1 remainders.
Step 4: 1 ÷ 2 = Ø and 1 remainders.

Take the remainders in reverse order and there you have binary equivalent

1101 = 13 decimal.

It is possible to write out binary numbers as powers of 2s and convert them to decimal.

1Ø11 binary = 8s 4s 2s 1s
1 Ø 1 1

or (1 × 8) + (Ø × 4) + (1 × 2) + (1 × 1)
or 8 + 2 + 1 = 11 decimal.

Hexadecimal (Base 16)

Hexadecimal is a numbering system that counts up to 15 before a carry is generated. The numbers 1Ø to 15 are represented by the letters A to F. Hexadecimal (hex) is convenient because each digit can be represented by exactly four binary digits (bits).

Table 1

Decimal	Binary	Hex
Ø	ØØØØ	Ø
1	ØØØ1	1
2	ØØ1Ø	2
3	ØØ11	3
4	Ø1ØØ	4
5	Ø1Ø1	5
6	Ø11Ø	6
7	Ø111	7
8	1ØØØ	8
9	1ØØ1	9
1Ø	1Ø1Ø	A
11	1Ø11	B
12	11ØØ	C
13	11Ø1	D
14	111Ø	E
15	1111	F

To add hexadecimal numbers remember a carry is only generated when the numbers add up to 'F'.

eg. 1.
```
  5
+ 7
---
  C
```

2.
```
  8
+ A
---
 13
```

3.
```
  13
  13
+ 26
```

4.
```
  1B
+ ØF
---
 2A
```

Now have a look at question 3 it looks like a normal decimal arithmatic, but remember it is hexadecimal. The true decimal equivalent is 38. If you saw that question by itself you could not tell whether it is hexadecimal or decimal. SO whenever a hexadecimal number is written down write an H after it so it can be identified as Hex.

All the questions above should have the letter H after each number.

e.g. 1.
```
  5H
+ 7H
----
  CH
```

2.
```
  8H
+ AH
-----
 13H
```

3.
```
  13H
+ 13H
-----
 26H
```

4.
```
  1BH
+ ØFH
-----
 2AH
```

This is a habit you must get into. Another habit is to place a stroke through every zero (e.g., Ø) this identified the figure as a zero and not the letter O.

e.g., The letters CODE can take on a completely new meaning. CØDEH.

How Do These 3 Number Systems Work Together.

The Z8Ø reads and writes data, 8 bits long in binary. It is very confusing to read 8 bits of binary and understand what the code represents (e.g., 11ØØØØ11). It is convenient to split the eight bits into 2 four bit codes and represent there value in hex.

1ØØ ØØ11 binary or C3 hex.

The Z8Ø has 16 address lines and these are also in binary. There are over 64 thousand possible combinations using these 16 address lines again it is difficult to remember a specific address if it is written down in binary (e.g.,

ØØ1Ø11ØØØØ111111). What we do is to split the 16 bits into 4 four bit codes and again use the hexadecimal equivalent.

ØØ1Ø, 11ØØ, ØØ11, 1111, = 2C3FH.

The 2CH is called the high byte, the 3FH is called the low byte.

When we use decimal numbers These must be converted to binary so the computer can store and operate on them.

The Z8Ø instruction set consists of 158 different types, which can be condensed down to the following groups.

Load exchange,
Block transfer and exchange,
Arithmetic and logical,
rotate and shift,
Bit manipulation,
Jump, call, return,
Input/output,
CPU control,

Addressing Mode.

Instructions can be one to four bytes long. The first byte of data fetched is used by the computer as the op code — i.e., tells the computer what to do with the data it already holds or data following the op code.

e.g., When the computer is first turned on it is reset and goes to the very first location in memory which is ØØØØH. If we examine ØØØØH we find that the code stored there is C3H. Now when the computer reads the code C3H, it is internally programmed to jump unconditionally to a new location. The second and sometimes third byte of data which is refered to as the operand and is the data which is to be operated upon by the first code.

e.g., If we look at the first example the code stored in ØØØØH is C3H which means jump unconditional to a new location, but were to? Its internal programming tell it that the new address is located in the following 2 memory locations. It reads the next address in memory which is ØØØ1H. The data there is used as the low byte of the new 16 bit address and then it reads the next location and uses the data there as the high byte of the new 16 bit address.

Here is step by step what happens in the computer.

Step 1.

The computer is turned on and is automatically reset (cold start), which loads the program counter with ØØØØH.

	Address	Code
Program Counter →	ØØØØH	C3H
	ØØØ1H	ØØH
	ØØØ2H	68H

The computer uses this address to fetch its op-code which is C3H. This tells the computer that it is to jump to a new address and the address is in the following 2 memory spaces.

Step 2.

	Address	Code
Program Counter →	ØØØØH	C3H
	ØØØ1H	ØØH
	ØØØ2H	68H

The program counter is automatically incremented and the data is read from this location, as the low byte of the new address.

Step 3.

	Address	Code
Program Counter →	ØØØØH	C3H
	ØØØ1H	ØØH
	ØØØ2H	68H

The program counter is automatically incremented and the data is read from this location and is the high byte of the new address.

Step 4.

	Address	Code
Program Counter →	68ØØH	C3H

The high and low byte are now placed into the program counter, and the computer is now ready to fetch its next instruction from this address.

i.e., The computer has jumped from ØØØØH to 68ØØH

Have a look at these memory locations yourself.

```
5 REM DISPLAY MEMORY IN HEX
10 A=PEEK(&H0000)
20 B$=HEX$(A)
30 PRINTB$
40 END
```

TABLE 2 contains the codes for moving 8 bits of information In and Out of the computer chip and from one register to another. To operate the table we must first decide were the information is coming from (source) and then were we want it to go (destination) the cross contains the code to achieve this.

If we wished to load Reg D with the code 6ØH we would use the "load immed" column, which would load the code straight into the register. Using the table we find the code is 16H, n. The 16H is the instruction code that tells the computer that data is to be loaded straight into Reg D and that the data is in the next memory following. The small n represents the value we wish to place into D which is 6ØH.

TABLE 2 SOURCE

DESTINATION		IMPLIED		REGISTER							REG INDIRECT			INDEXED		EXT ADDR	IMME
		I	R	A	B	C	D	E	H	L	HL	BC	DE	IX + d	IY + d	(nn)	(n)
REGISTER	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	ØA	1A	DD 7E d	FD 7E d	3A n n	3E n
	B			47	4Ø	41	42	43	44	45	46			DD 46 d	FD 46 d		Ø6 n
	C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		ØE n
	D			57	5Ø	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
	E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
	H			67	6Ø	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
	L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
REG INDIRECT	HL			77	7Ø	71	72	73	74	75							36 n
	BC			Ø2													
	DE			12													
INDEXED	IX + d			DD 77 d	DD 7Ø d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d
	IY + d			FD 77 d	FD 7Ø d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d
EXT ADDR	(nn)			32 n n													
IMPLIED	I			ED 47													
	R			ED 4F													

8 BIT LOAD GROUP

If we wished to move data from one register to another (say we now wished to load the data in Reg D into Reg A) using the table we find D (source) to A (destination) is the code 7AH.

The following program will enable you to load a register with a value move it to Reg A, and then the contents of A will be placed into memory were you can examine it to ensure you have used the correct codes.

The program puts the data into memory locations AØØØ to AØØ7 and then jumps to the data and runs the program entered and prints the result in memory &HAØ1Ø.

You can modify the program by 1. changing the value of the data, 2. which register the data is loaded into, but don't forget to change the code which loads this data into A. i.e., If data is loaded into Reg C the code must be changed to load Reg C into Reg A).

1. AØØØH 16H:
   AØØ1H 6ØH: LOADS D IMMEDIATELY WITH THE DATA 6ØH.
2. AØØ2H 7AH: MOVES THE DATA FROM REG D TO REG A.
3. AØØ3H 21H:
   AØØ4H 1ØH: PUTS THE ADDRESS AØ1ØH INTO REG HL
   AØØ5H AØH: (This code is not in the table, more on this later)
4. AØØ6H 77H: PUTS THE DATA IN A INTO THE LOCATION STORED IN HL.
   AØØ7H C9H: RETURNS THE PROGRAM BACK TO BASIC.
   (Code not in table)

Here is a short assignment.

1. How many states are there is a Hexadecimal number system.
2. Convert the following decimal numbers to binary and then to hexadecimal.

2, 6, 32, 11, 1Ø, 15, 8, 14

3. Add the following binary numbers,

a. ØØØØ11Ø1 + ØØØØØ111,
b. ØØØØØØ1Ø + ØØØØØØ1Ø,
c. ØØØØ11Ø1 + ØØØØ1ØØØ,
d. Ø11ØØ111 + ØØØ1ØØØ1

4. Check your answers to the above, by converting the original numbers and answers into decimal.
5. Convert the following hexadecimal numbers to binary

a. F14A b. Ø521, c. 49Ø2, d. CØDE
The answers to these questions will be in the next issue.

Continued from Page 16

Noise - various sounds which can be used to produce interesting sound effects. A noise, rather than a tone, is generated by the SOUND subprogram* when commands 4 and 5 are specified.

Null String - a string* which contains no characters and has zero length.

Number Mode - the mode assumed by the computer when it is automatically generating program line* numbers for entering or changing statements.

Operator - a symbol used in calculations (numeric operators) or in relationship comparisons (relational operators). The numeric operators are +,-,*,/,. The relational operators are , , = , = , ., = , = , =

Overflow - the condition which occurs when a rounded value greater than 9.9999999999999E+99 or less than -9.9999999999999E-99 is entered or computed. When this happens, a warning is displayed, and the program* stops.

Output - (noun) information supplied by the computer; (verb) the process of transferring information from the computer's memory onto a device, such as a screen, line printer, or mass storage device*.

Parameter - any of a set of values that determine or affect the output of a statement* or function*.

Print Line - a 38-position line used by the PRINT statement.

Program - a set of statements which tell the computer how to perform a complete task.

Program Line - a line containing a single statement*. The maximum length of a program line is 248 characters*.

Pseudo-random number - a number produced by a definite set of calculations (algorithm) but which is sufficiently random to be considered as such for some particular purpose. A true random number is obtained entirely by chance.

RAM - random access memory; the main memory where program statements and data* are temporarily stored during program execution*. New programs and data can be read in, accessed, and changed in RAM. Data stored in RAM is erased whenever the power is turned off of BASIC is exited.

Reserved Word - in programming languages, a special work with a predefined meaning. A reserved word lmust be spelled correctly, appear in the proper order in a statement* or command*, and cannot be used as a variable* name.

ROM - read-only memory; certain instructions for the computer are permanently stored in ROM and can be accessed but cannot be changed. Turning the power off does not erase ROM.

Run Mode - when the computer is executing* a program, it is in Run Mode. Run Mode is terminated when program execution ends normally or abnormally. You can cause the computer to leave Run Mode by pressing BREAK during program execution (see Breakpoint*).

Scientific Notation - a method of expressing very large or very small numbers by using a base number (mantissa*) times ten raised to some power (exponent*). To represent scientific notation in SEGA BASIC enter the sign, then the mantissa, the letter E, and the power of ten (preceded by a minus sign if negative). For example, 3,264; -2.47E -17.

Scroll - to move the text on the screen so that additional information can be displayed.

Software - various programs which are executed by the computer, including programs built into the computer. Cartridges* programs, and programs entered by the user.

Statement - an instruction preceded by a line number in a program. In SEGA BASIC, more than one statement is allowed in a program line*.

String - a series of letters, numbers and symbols treated as a unit.

Subprogram - a predefined general-purpose procedure accessible to the user through the statement in SEGA BASIC. Subprograms extend the capability of BASIC and cannot be easily programmed in BASIC.

Subroutine - a program segment which can be used more than once during the execution* of a program, such as a complex set of calculations of a print routine. In SEGA BASIC a subroutine is entered by a GOSUB statement and ends with a RETURN statement.

Subscript - a numeric expression which specifies a particular item in an array*. In SEGA BASIC the subscript is written in parentheses immediately following the array name.

Underflow - the condition which occurs when the computer generates a numeric value greater than -1E-100, less than 1E-100, and not zero. When an underflow occurs, the value is replaced by zero.

Variable - a name is given to a value which may vary during program execution. You can think of a variable as a memory location where values can be replaced by new values during program execution.

***See definition in Glossary**

PROGRAMMING EXPLORED

This month we will deal with the read, data and restore statements.

These commands are used to store and input information into a program.

```
10 READ X,Y,Z
20 PRINT X;Y;Z
30 DATA 1,3,5
```

When the computer sees the statement READ it looks at the first data statement in the program (in this case line 30). The computer then inputs the information after the word DATA in to the variable or variables following the read statement (in this case X,Y and Z in line 10).

After the computer has put the first piece of data into X (in this case the number 1), the computer shifts onto the next number in the list which is 3. This number is placed in Y. The computer now moves onto look at 5 which is placed in Z. Your SC3000 uses what is called a "DATA POINTER" to tell it which piece of information it should be looking at.

VARIABLE	VALUE
X	1
Y	3
Z	5

Now try changing line 30 to read

```
30 DATA HELLO, NICE, DAY
```

Doesn't work does it. A type missmatch error results because you are trying to put a STRING ($) into a VARIABLE (a number). Change the program to read:

```
10 READ X$,Y$,Z$
20 PRINT X$;Y$;Z$
30 DATA HELLO ,NICE ,DAY
```

TEST

The program above could also have been done like this:

```
10 FOR J=1 TO 3
20 READ X$
30 PRINT X$;:NEXT
40 DATA HELLO ,NICE ,DAY
```

Notice there is only 1 variable after the read statement (this is X$) each time the computer reads a piece of DATA it replaces the value in X$ with the new piece of information. The DATA POINTER is then shifted onto the next piece of data. Now type in:

```
TEST
10 READ A$
20 PRINT A$;" ";
30 BEEP:GOTO 10
40 DATA HELLO,HOW,ARE,YOU,TODAY,?
```

The program crashes as the computer does not have enough data (the data pointer is pointing at nothing). To stop this happening we must have enough data for the read statement. This can be done with a FOR-NEXT Loop.

```
10 FOR J=1 TO 6
20 READ A$:PRINT A$;" ";
30 BEEP:NEXT
40 DATA HELLO,HOW,ARE,YOU,TODAY,?
```

Now try deleting line 40 and replacing it with line 5.

```
5 DATA HELLO,HOW,ARE,YOU,TODAY,?
```

Try running the program. Notice how it still works. The computer will read DATA from any part of the program. Notice that it only READS the information from the data line. It does not send the program there or delete the information:

```
5 SCREEN 2,2:CLS
10 DATA 50,50,50,100,50,100,100,100,100,100,100,50,100,50,50,50
20 DATA 50,50,75,25,75,25,100,50
30 DATA THIS,IS,A,RATHER,SILLY,LITTLE,PROGRAMME,ISN'T,IT,'
40 FOR J=1 TO 6
50 READ X,Y,X1,Y1
60 LINE(X,Y)-(X1,Y1)
70 NEXT
80 FOR K=1 TO 10
90 READ B$:PRINT B$;" ";
100 NEXT
110 GOTO 110
```

RESTORE

This statement moves the Data Pointer to the first line of DATA or to the line you nominate:

```
10 DATA HELLO
20 READ A$
30 PRINT A$;" ";
40 RESTORE
50 GOTO 20
```

Try deleting Line 40. You get an out of data error, as there is no data left to read after the word hello.

EXAMPLES

```
10 SCREEN 2,2
20 CLS:RESTORE:FOR J=1 TO 15
30 READ C,C$
40 IF J=1 THEN COLOR 15:PRINT C$:GOTO 60
50 COLOR 1:PRINT C$
60 COLOR C,C,(0,0)-(255,191)
70 NEXT
80 GOTO 20
90 DATA1,BLACK,2,GREEN,3,LIGHT GREEN,4,DARK BLUE,5,LIGHT BLUE,6,DARK RED,7,CYAN,
8,RED,9,LIGHT RED,10,DARK YELLOW,11,LIGHT YELLOW,12,DARK GREEN,13,MAGENTA,14,GRE
Y,15,WHITE
```

```
10 READ N,D
15 IF N=0 AND D=0 THEN END
20 SOUND 1,N,14:FOR DE=1 TO D:NEXT DE
30 SOUND 0:GOTO 10
100 DATA 494,50,494,50,523,50,587,50
110 DATA 587,50,523,50,494,50,440,50
120 DATA 392,50,392,50,440,50,494,50
130 DATA 494,75,440,25,440,100
140 DATA 494,50,494,50,523,50,587,50
150 DATA 587,50,523,50,494,50,440,50
160 DATA 392,50,392,50,440,50,494,50
170 DATA 440,75,392,25,392,100
180 DATA 440,50,440,50,494,50,392,50
190 DATA 440,50,494,10,523,10,494,50,392,50
200 DATA 440,50,494,10,523,10,494,50,440,50
210 DATA 392,50,440,50,294,100
220 DATA 494,50,494,50,523,50,587,50
230 DATA 587,50,523,50,494,50,440,50
240 DATA 392,50,392,50,440,50,494,50
250 DATA 440,75,392,25,392,100
400 DATA 0,0
```

JOYSTICK OPERATION USING MACHINE CODE ROUTINE

By I.W. Nicholson

The Sega Computer is capable of being programmed to produce exceptionally good graphics for use in games, educational programmes etc.
However programmes written in BASIC suffer from the disadvantage that they are too slow for some applications such as the control of sprite movement. These may be sped up by programming so that the sprites move in large steps but precision of control is lost.
The problem is overcome by using machine code routines for those parts of the programmes where BASIC is too slow.
This article describes how a machine code routine may be used to control the movement of a sprite across the screen by the use of a joystick.
Operation of joystick control is best described by means of a flow diagram shown below.

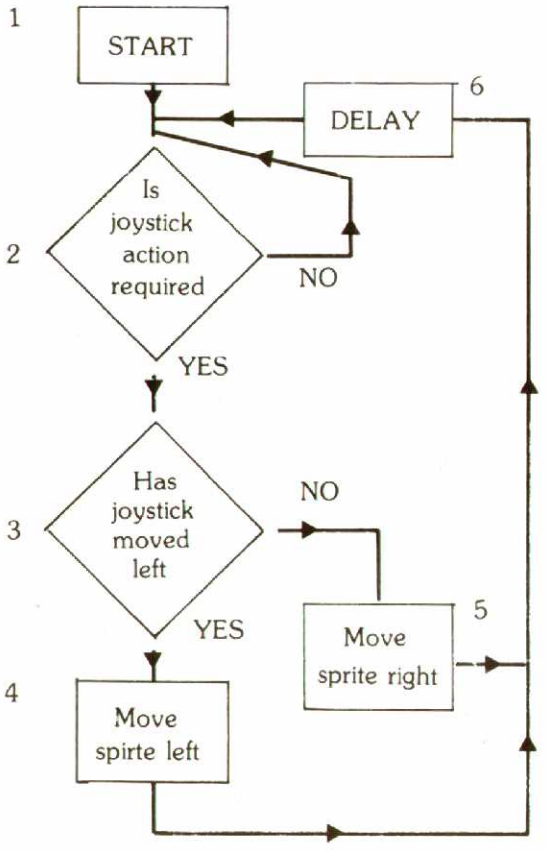


Fig 1. Flow diagram of simple joystick operation.

Operation of flow diagram broad terms is as follows:

1. START:- Computer sets up its internal registers so that the joystick port can be read. A register is another name for a variable.
2. The joystick having been read by the computer a decision is required to determine if the joystick has been moved away from its central resting position. i.e., if the joystick has not moved then a loop is set about stage 2 until the joystick has operated.
3. A decision is now made to determine the movement of the joystick. If it has moved to the left then control is passed to stage 4. However, if movement is in any other direction then control is passed to stage 5.
4 and 5. Consists of routines to move the sprite 1 pixel left or right.
6. After movement of the sprite a delay is required before the process is repeated again. This is necessary because without the delay, machine code is so fast that you will hardly see the sprite flash across the screen.

Detailed Operation of the Flow Diagram.

1. The joystick, keyboard and other input/output devices are connected to the computers processor via a programmable interface chip. The input/output device that is to recognised by the computers processor is determined by the address that has been programmed into the interface chip.

Fig 2 below shows this in pictorial form.

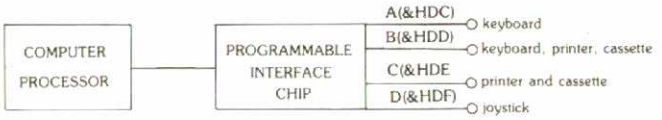


Fig 2. Interface:- Computer — input/output devices.

Note that the joysticks are connected to port address 8HDF so the interface chip needs to be programmed with the appropriate value.
Values that appear at this port for various positions of joystick No. 1 are as follows:-

Table 1:- Port values for joystick No. 1.

No action	FF
UP	FE
DOWN	FD
LEFT	FB
RIGHT	F7
FIRE L	EF
FIRE R	DF

Knowing the above facts a start to the build up of a machine programme can begin as follows:-

1. LDA,9B
   OUT (DF), A
   IN A (DC) — Sets up interface chip to read joystick No. 1. in A (DC) Reads joystick and places value into accumulator "A"
2. CP FF — Compares accumulator A with &HFF (see table 1)
   JPZ. . . . — If accumulator A is equal to &HFF then return to step 2 i.e., joystick is in its central position. If joystick has moved then programme steps to next stage 3
3. CP F7 — Compare accumulator A with F7. (Remember F7 is the value in port A if the joystick has moved right) JPNZ If accumulator A is not equal to F7 then programme control is passed to stage 5

4. LDHL 3B01

Loads HL register with 3B01 (3B01 is the address where the X value of sprite No. 0 is stored)

CALL 2C32	Routine in computer memory which reads the X value of sprite No. 0
INA,(BE)	Puts the X value of sprite No. 0 into accumulator "A"
INC A	Increases the value in accumulator "A" by 1. i.e., X value is increased by one.
CALL 2C44	Sets video display section up for a new update
OUT(BE),A	Updates video ram with new data i.e., sprite moves to the right by one pixel

5. Similar to stage 4 except that the INC A term is changed to a DEC A in order to move the sprite to the left.

Because of the process just described takes only a very minute time to process then a time delay is necessary before the process repeats itself again (refer to Fig. 1). This is described below.

LDBC ØЗØØ	Load accumulator "BC" with Ø3ØØ (time delay)
DEC BC	Decrements accumulator "BC" by 1
LDA, ØØ	Load accumulator "A" with Ø
CPB	Compares accumulator "B" with A i.e., Ø
JRNZ. . . .	If accumulator A is not equal to B then
JP. . . .	Jump to read joystick again

For different time delay the value in the BC register can be changed to a different value from ØØØØ to FFFF for maximum delay.

If further delays are required then it is necessary to use more than the single loop shown here.

Machine Code Programme in full:-

Address		Machine Code
98Ø8	LDA,9B	3E9B
98ØA	OUT(DF),A	D3DF
98ØC	INA,(DC)	DBDC
98ØE	CPFF	FEFF
98ØØ	JRZ98ØC	28FA
9812	CPF7	FEF7
9814	JRNZ9826	2Ø1Ø
9816	LDHL3BØ1	21Ø13B
9819	CALL2C32	CD322C
981C	INA,(BE)	DBBE
981E	INC A	3C
981F	CALL2C44	CD442C
9822	OUT(BE),A	D3BE
9824	JR9834	18ØE
9826	LDHL3BØ1	21Ø13B
9829	CALL2C32	CD322C
982C	INA,(BE)	DBBE
982E	DEC A	3D
982F	CALL2C44	CD442C
9832	OUT(BE),A	D3BE
9834	LD BC Ø3ØØ	Ø1ØØØ3
9837	DEC BC	OB
9838	LD A	3EØØ
983A	CPB	B8
983B	JRNZ9837	2ØFA
983D	JR 98ØC	18CD

The remaining problem is how does one enter the machine code programme into the computer.

This is achieved by poking the data into a REM statement at the beginning of a programme.

Initially the REM statement would have about 6Ø "As" in it. However when the programme is run the machine code overwrites the "As" with the appropriate information.

The memory location of the first "A" is 98Ø8 so if we use the BASIC command CALL 8H98Ø8 then control is passed from BASIC to MACHINE CODE.

Below is the BASIC programme necessary to enter the MACHINE CODE into the computer.

programme to go here.

Try this programme. You will notice that movement of the joystick to the right will move the sprite right. However movement of the joystick in any other direction will move the joystick left (ref Fig. 1). To stop the programme the RESET button needs to be pressed.

It is interesting to compare this programme with its BASIC euivalent by running the two programmes one after the other. The difference needs to be seen to be appreciated.

Below is a listing of the basic equivalent which will move the sprite across the screen at about the same speed.

This can be entered as an addition to the MACHINE CODE version.

```
5 REM AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
10 RESTORE:FOR X=&H980B TO &H983E:READ B$:C$="&H"+B$:C=VAL(C$):POKE X,C:NEXT
20 DATA 3E,98,D3,DF,DB,DC,FE,FF,28,FA,FE,F7,20,10,21,01,3B,CD,32,2C,DB,BE,3C,CD,
44,2C,D3,BE,18,0E,21,01,3B,CD,32,2C,DB,BE,3D,CD,44,2C,D3,BE,01,00,03,0B,3E,00,B8
,20,FA,18,CD
30 SCREEN 2,2:CLS:PATTERN S#0,"542A542A542A542A"
40 SPRITE 0,(120,95),0,1
50 CALL &H980B
60 NEXTZ

60 SCREEN 2,2:CLS:PATTERN S#0,"542A542A542A542A":X=120:Y=95
70 IF X<0 THEN X=250
80 IF X>250 THEN X=0
90 SPRITE 0,(X,Y),0,1:S1=STICK(1):IF S1=7 THEN X=X-1:GOTO 70
100 IF S1=3 THEN X=X+1:GOTO 70
110 GOTO 70
```

NOTE:- To run this part of the programme type in "GOTO 6Ø"

EXTRAS FOR EXPERIMENTING

(1) Try altering the delay in the MACHINE CODE programme to change the speed of the cursor. The delay is governed by bc at add. 9834, 9835, 9836

(2) Try making the cursor go up and down:- Hint: Address of Y position is at 3BØØ

(3) For the real keen try making the cursor go up and down as well as left and right:- Hint: Refer to Fig. 1. Extra decisions need to be inserted between 3 and 5

CAUTION. When experimenting with machine code extra care is needed for you will find that the entire programme that has been entered is quite likely to be erased if only the slightest error is made. It is not forgiving like BASIC which will inform you of any syntax errors.

MAZERTRON

by Alan Dobbs

This is one of the best 3-D maze game we have seen yet for the Sega, uses very little memory. The aim of the game is to escape from the maze (there are 8 of them located in lines 46 to 53) before your supply of precious oxygen is exhausted. This program makes full use of the Sega's line statements and string handling functions.

```
1 SCREEN2,2:CLS:COLOR13,12,(0,0)-(255,191),12:LINE(0,0)-(255,191),1,B:LINE(36,76
)-(210,110),1,B:CURSOR40,80:PRINTCHR$(17);"WOULD YOU LIKE":CURSOR46,90:PRINT"INS
TRUCTIONS?":CURSOR76,100:PRINT"(Y OR N)"
2 C$=INKEY$:IFC$="N"THENBEEP:GOTO9
3 IFC$<>"Y"ANDC$<>"N"THEN2
4 BEEP:CLS:COLOR4,1,(0,0)-(255,191),1:FORF=0TO6STEP2:LINE(F,F)-(255-F,191-F),15,
B:NEXT:CIRCLE(128,30),80,13,.2,0,1:LINE(78,24)-(176,34),,B:CURSOR80,26:PRINT"MAZ
ATRON":PAINT(72,20),5
5 LINE(10,50)-(245,181),15,B:CURSOR56,60:PRINT"INSTRUCTIONS":LINE(56,70)-(198,70
),15:LINE(14,76)-(241,177),15,B:PRINTCHR$(16)
6 CURSOR18,80:PRINT"Your aim is to escape from the maze.":CURSOR18,90:PRINT"The
exit is indicated by a ";CHR$(249);" when it":CURSOR18,100:PRINT"is in sight.":C
URSOR18,110:PRINT"Your oxygen supply is dwindling,and"
7 CURSOR18,120:PRINT"you will die if it runs out.":CURSOR18,130:PRINT"Use the cu
rsor keys for movement.":CURSOR18,140:PRINT"The down key turns you around.":CURS
OR86,160:PRINT"Press any key"
8 IFINKEY$=""THEN8
9 BEEP:GOSUB46
10 A1=1:SCREEN2,2:CLS:COLOR10,15,(0,0)-(255,191),15:PRINTCHR$(16):B$=CHR$(31):GO
TO17
11 B$=INKEY$:IFB$=CHR$(30)ANDMID$(Q$,A+A1,1)="0"THENA=A+A1
12 IFB$=CHR$(31)THENA1=-A1
13 IFB$=CHR$(29)THENA1=(A1=12)+(-1*(A1=-12))+(-12*(A1=1))+(12*(A1=-1))
14 IFB$=CHR$(28)THENA1=(A1=-12)+(-1*(A1=12))+(-12*(A1=-1))+(12*(A1=1))
15 IFB$=CHR$(28)ORB$=CHR$(29)ORB$=CHR$(30)ORB$=CHR$(31)THEN17
16 GOTO11
17 O=O-.5:IFINT(O)<0THEN40
18 N=2*A1:BEEP:CLS:LINE(53,21)-(203,171),1,B:LINE(53,173)-(203,187),1,B:IFA=ZTHE
N34
19 IFA=ZTHEN34
20 COLOR6:CURSOR58,176:PRINT"OXYGEN~";LEFT$(O$,O);" "
21 IFMID$(Q$,A+A1,1)="1"THENLINE(99,92)-(150,102),1,B:CURSOR102,94:PRINTCHR$(17)
;"WALL":BEEP:BEEP:PRINTCHR$(16):GOTO11
22 IFMID$(Q$,(A+(A1=12)+(-1*(A1=-12))+(-12*(A1=1))+(12*(A1=-1))),1)="1"THENLINE(
54,22)-(92,60),1:LINE-(92,132),1:LINE-(54,170),1
23 IFMID$(Q$,(A+(A1+(A1=12)+(-1*(A1=-12))+(-12*(A1=1))+(12*(A1=-1)))),1)="1"THEN
LINE(92,60)-(118,86),1:LINE-(118,106),1:LINE-(92,132),1:LINE-(92,60),1
24 IFMID$(Q$,A+N,1)="1"THENLINE(118,86)-(138,106),1,B
25 IFMID$(Q$,(A+(A1+(A1=-12)+(-1*(A1=12))+(-12*(A1=-1))+(12*(A1=1)))),1)="1"THEN
LINE(138,86)-(164,60),1:LINE-(164,132),1:LINE-(138,106),1:LINE-(138,86),1
26 IFMID$(Q$,(A+(A1=-12)+(-1*(A1=12))+(12*(A1=1))+(-12*(A1=-1))),1)="1"THENLINE(
202,22)-(164,60),1:LINE-(164,132),1:LINE-(202,170),1
27 IFMID$(Q$,(A+(A1=12)+(-1*(A1=-12))+(-12*(A1=1))+(12*(A1=-1))),1)="0"THENLINE(
53,60)-(92,132),1,B
28 IFMID$(Q$,(A+(A1+(A1=12)+(-1*(A1=-12))+(-12*(A1=1))+(12*(A1=-1)))),1)="0"THEN
LINE(92,86)-(118,106),1,B
29 IFMID$(Q$,A+N,1)="0"THENLINE(118,86)-(138,106),1:LINE(138,86)-(118,106),1
30 IFMID$(Q$,(A+(A1+(A1=-12)+(-1*(A1=12))+(-12*(A1=-1))+(12*(A1=1)))),1)="0"THEN
LINE(138,86)-(164,106),1,B
31 IFMID$(Q$,(A+(A1=-12)+(-1*(A1=12))+(-12*(A1=-1))+(12*(A1=1))),1)="0"THENLINE(
164,60)-(203,132),1,B
32 IFA+N=ZORA+A1=ZANDMID$(Q$,A+A1,1)="0"THENCURSOR126,94:PRINTCHR$(249)
33 GOTO11
34 IFO<0THENO=0
35 CLS:COLOR1,15,(0,0)-(255,191),15:FORF=0TO10STEP2:LINE(F,F)-(255-F,191-F),1,B:
NEXT
36 LINE(40,14)-(216,96),1,B:CURSOR70,20:PRINTCHR$(17);"WELL DONE":BEEP:BEEP:FORF
=0TO100:NEXT:CURSOR58,40:PRINT"YOU ESCAPED":CURSOR46,60:PRINT"WITH";INT(O);" OXY
GEN":CURSOR64,84:PRINT"UNITS LEFT"
37 FORF=1TO14:LINE(40,136)-(216,150),F,B:CURSOR46,140:PRINT"PRESS ANY KEY"
```

```
38 IFINKEY$>""THENBEEP:GOTO9
39 NEXT:GOTO37
40 CLS:COLOR15,1,(0,0)-(255,191),1:FORF=0TO10STEP2:LINE(F,F)-(255-F,191-F),15,B:
NEXT
41 LINE(20,20)-(235,64),15,B:CURSOR26,30:PRINTCHR$(17);"YOU HAVE SUFFERED":CURSO
R50,40:PRINT"A BAD CASE OF":CURSOR43,50:PRINT"TERMINAL DEATH":CIRCLE(128,120),10
0,8,.5,0,1
42 CURSOR106,80:PRINT"This":CURSOR78,90:PRINT"is mainly":CURSOR55,100:PRINT"due
to a lack":CURSOR118,110:PRINT"of":LINE(90,126)-(172,140),4,B:CURSOR96,130:PRINT
"OXYGEN"
43 COLOR15:PRINTCHR$(16):CURSOR92,170:PRINT"Press any key"
44 IFINKEY$=""THEN44
45 BEEP:GOTO9
46 V=INT(RND(1)*8)+1:IFV=1THENQ$="1111111111111000000011011010101010011011001001
0110001010110110100010000110001011011110101000000110001011011110110001010110010
0000001111111111111"
47 IFV=2THENQ$="1111111111111011000010011000101000111110111110011000000000111010
1110110110000010000110101000101110000010000110101011010110000000001001111111111
"
48 IFV=3THENQ$="1111111111111000000000001101011010101101000000000110001011010110
0000000110001101010110101101000011000000000101101010110101100000000000011111111111
"
49 IFV=4THENQ$="1111111111111000001000011011100101011010001000011010101010111001
1010000110100001110110101010000110100010111110111110101110000000000011111111111
"
50 IFV=5THENQ$="1111111111111000000100011010110101011010010101011001010100011011
0101010110000100010111111101010110000001010110101111010110000000000011111111111
"
51 IFV=6THENQ$="1111111111111000001000001101101101101101001001001101011010101101
0000000110101010110110100000000011001110111011110101000101100001010100111111111111
"
52 IFV=7THENQ$="1111111111111100001100001101101001011100101011001101101010101100
0011100011111000101011100010101011101101111011000000100011101010001011111111111
1"
53 IFV=8THENQ$="1111111111111001000000011010011110111011000100011000011110111011
0000000110110111011110100001010110110101000110111111111011000000000001111111111
"
54 O$="00000000000000000":O=18
55 A=INT(RND(1)*132)+1:IFMID$(Q$,A,1)<>"0"THEN55
56 Z=132:IFA=ZORA+3=ZORA-3=ZORA+12=ZORA-12=ZTHEN55
57 RETURN
```

SEGA SUPERSTAR

HIGH SCORE TABLE

The person recording the highest points tally on each of the Sega games cartridges will receive a free cassette program, and have their name published in the Magazine, along with their score.

To have your score entered, we must have some form of verification as to its validity, ideally a photograph of the screen, showing the recorded high score. In games where the high score may be flashing, the screen action can be frozen, by depressing the Reset key.

High Score Table

This month we have 2 degrees in space invaderology to present. The awards with honours, go to:

Mark Culter
Palmerston North

For his outstanding performances producing these scores:

Star Jack 726725
Popflamer 226000

If we were holding a Sega Champion Golf open, there is one player who would be at the top of the leader board.

Hank Hoitianga of Henderson, Auckland, obviously had no trouble putting while scoring an amazing 8 under the card for round 1.

HOLE	SCORE
1	3
2	5
3	2
4	3
5	3
6	3
7	3
8	4
9	2

Star Jacker 726725
Pop Flamer 226000
Golf -8

Monaco GP 999999
Pacar 1,526,400
Video Flipper 110,600

SERPENT

BY COSTA KERDEMELIDIS

This program is being awarded the program of the month prize for its superb use of the Sega's sprites, sound (capability). The writer of this program has received a cartridge game of his choice. The idea of the game is that you are a snake that has to move it way around the garden eating apples and lemons and magic mushrooms. The game gets harder and harder as the screen is progressively fulled as you move on to the next round.

You are awarded points eating apples and lemons and also as your length increases, however, be aware of the magic mushrooms as if you eat one of these the snake is reversed so that its old tail becomes the head.

The serpent also leaves droppings around the garden which you must not eat otherwise severe stomach ache and death will occur. Your must also be very careful not to crash into the walls or yourself.

The control of the snake is either with the joystick or the left and right arrow keys. You can go left or right according to the direction of travel. What this means is that you must think of yourself as the serpent and decide if you must turn to the serpents left or the serpents right. This can make things rather confusing!

```
5 DIM A(500):DIM B(500):GOSUB 5100:GOSUB 4000:GOSUB 4500:HS=0
10 SCL=0:LE=1
20 GOSUB 2100:GOSUB 2000:COLOR2,1,(0,0)-(255,191),1:KF=0
50 SPRITE 0,(X,Y-2),0,2
51 E$=INKEY$:P1=STICK(1)
52 ON DI GOSUB 200,203,206,209
53 GOSUB 400
54 IF FL=1 THEN FL=0:GOTO 20
55 IF FL=2 THEN FL=0:GOTO 10
56 GOTO 51
200 IF P1=7ORE$=CHR$(29) THEN X=X-8:DI=4:RETURN
201 IF P1=3ORE$=CHR$(28) THEN X=X+8:DI=3:RETURN
202 Y=Y-8:RETURN
203 IF P1=7ORE$=CHR$(29) THEN X=X+8:DI=3:RETURN
204 IF P1=3ORE$=CHR$(28) THEN X=X-8:DI=4:RETURN
205 Y=Y+8:RETURN
206 IF P1=7ORE$=CHR$(29) THEN Y=Y-8:DI=1:RETURN
207 IF P1=3ORE$=CHR$(28) THEN Y=Y+8:DI=2:RETURN
208 X=X+8:RETURN
209 IF P1=7ORE$=CHR$(29) THEN Y=Y+8:DI=2:RETURN
210 IF P1=3ORE$=CHR$(28) THEN Y=Y-8:DI=1:RETURN
211 X=X-8:RETURN
270 IF P1=3ORE$=CHR$(28) THEN Y=Y+8:DI=2:RETURN
280 X=X+8:RETURN
290 IF P1=7ORE$=CHR$(29) THEN Y=Y+8:DI=2:RETURN
300 IF P1=3ORE$=CHR$(28) THEN Y=Y-8:DI=1:RETURN
310 X=X-8:RETURN
400 SPRITE 0,(X,Y-2),0,2:IF CH=1 THEN CH=0:GOSUB 1800:GOTO 411
401 AD=VPEEK(INT((Y+3)/8)*256+INT(X/8)*8+(Y+3)MOD8)
402 IF C=500 THEN C=0
403 IF X<=16ORX>=248ORY<=8ORY>=176 THEN GOSUB 800
404 IF AD=28 THEN SC=SC+10:GOSUB 1500:CURSOR220,184:PRINT CHR$(5);SC:KF=KF+1:IF
KF=LE*8 THEN 1900
405 IF AD=24 THEN SC=SC+20:GOSUB 1600:CURSOR220,184:PRINT CHR$(5);SC:KF=KF+1:IF
KF=LE*8 GOTO  1900
407 IF AD=252 THEN 800
409 IF AD=248 THEN 800
410 IF AD=126 THEN GOTO 1700
411 C=C+1:A(C)=X:B(C)=Y:R=R+1
412 IF R=10 THEN R=0:CURSOR 220,184:SC=SC+5:PRINT CHR$(5);SC:TL=TL+2:TU=1:GOTO 4
16
413 IF TU=0 THEN 416
414 IF TU=1 THEN CURSOR A(C-1),B(C-1):PRINT CHR$(236):TK=TK+1:IF TK=2 THEN TK=0:
TU=0:RETURN
415 RETURN
```

```
416 CURSOR A(C-1),B(C-1):PRINT CHR$(236):CURSOR A((C-TL)-1),B((C-TL)-1):PRINT CH
R$(8):RETURN
550 AD=INT(Q/8)*256+INT(Z/8)*8+QMOD8
560 RETURN
600 VPOKEAD+1,&H18:VPOKEAD+1+&H2000,&H81
610 VPOKEAD+2,&H3C:VPOKEAD+1+&H2000,&H81
620 VPOKEAD+3,&H7E:VPOKEAD+2+&H2000,&H81
630 VPOKEAD+4,&H7E:VPOKEAD+3+&H2000,&H81
640 VPOKEAD+5,&H18:VPOKEAD+4+&H2000,&HF1
650 VPOKEAD+6,&H18:VPOKEAD+5+&H2000,&HF1
660 RETURN
700 CIRCLE(X+4,Y+4),3,4,1,0,1,BF
710 RETURN
800 OUT &H7F,&HE4:FOR X=&HF0 TO &HFF:OUT&H7F,X
810 FOR DE=1 TO 20:NEXT DE
820 NEXT X
840 CIRCLE (A(C)+2,B(C)+3),3,13,1,0,1
850 CIRCLE (A(C)+2,B(C)+3),5,13,1,0,1
860 FOR I=0 TO TL
870 CIRCLE (A(C-I)+2,B(C-I)+3),2,13,1,0,1,BF:NEXT I:IF SC>HS THEN HS=SC
880 SCREEN 2,2:CLS
890 CURSOR 100,80:PRINT"YOUR SCORE WAS";:PRINT SC
900 CURSOR100,50:PRINT "TRY AGAIN Y/N?":GOSUB 3000:FL=2:RETURN
1000 VPOKEAD+1,&H04:VPOKEAD+2+&H2000,&HE1
1010 VPOKEAD+2,&H08:VPOKEAD+2+&H2000,&HE1
1020 VPOKEAD+3,&H18:VPOKEAD+3+&H2000,&H81
1030 VPOKEAD+4,&H3C:VPOKEAD+4+&H2000,&H81
1040 VPOKEAD+5,&H3C:VPOKEAD+5+&H2000,&H81
1050 VPOKEAD+6,&H18:VPOKEAD+6+&H2000,&H81
1060 RETURN
1070 VPOKEAD+1,&H08:VPOKEAD+1+&H2000,&H21
1080 VPOKEAD+2,&H1C:VPOKEAD+2+&H2000,&HA1
1090 VPOKEAD+3,&H1C:VPOKEAD+3+&H2000,&HA1
1100 VPOKEAD+4,&H1C:VPOKEAD+4+&H2000,&HA1
1110 VPOKEAD+5,&H1C:VPOKEAD+5+&H2000,&HA1
1120 VPOKEAD+6,&H08:VPOKEAD+6+&H2000,&HA1
1130 RETURN
1500 SOUND1,698,15:SOUND1,831,15:SOUND1,880,15:SOUND1,698,15:SOUND0
1510 RETURN
1600 SOUND1,440,15:SOUND1,415,15:SOUND1,349,15:SOUND1,440,15:SOUND0
1610 RETURN
1700 X=A(C-TL):Y=B(C-TL):FOR I=0 TO TL
1710 A(C-I)=A(C-TL+I)
1720 NEXT I
1730 FOR I=0 TO TL:B(C-I)=B(C-TL+I):NEXT I
1740 CH=1:GOTO 400
1800 IF DI=1 THEN DI=2:RETURN
1810 IF DI=2 THEN DI=1:RETURN
1820 IF DI=4 THEN DI=3:RETURN
1830 IF DI=3 THEN DI=4:RETURN
1900 SCREEN 2,2:CLS:CURSOR 20,80
1901 SOUND1,392,15:GOSUB 4210:SOUND 0:SOUND1,392,15:GOSUB 4210:SOUND1,523,15:GOS
UB 4210:SOUND1,392,15:GOSUB 4210:SOUND1,523,15
1902 GOSUB 4210:SOUND1,659,15:GOSUB 4210:SOUND1,523,15:GOSUB 4210:GOSUB 4210:SOU
ND0:SOUND1,523,15:GOSUB 4210:SOUND 0:SOUND1,523,15:GOSUB 4210
1903 SOUND1,659,15:GOSUB 4210:SOUND1,523,15:GOSUB 4210:SOUND1,659,15:GOSUB 4210:
SOUND1,784,15:GOSUB 4210:SOUND1,659,15:GOSUB 4210:GOSUB 4210:SOUND0
1906 IF LE>5 THEN MES$(LE)=MES$(5)
1907 PRINT MES$(LE);:PRINT "-PREPARE FOR ROUND";:LE=LE+1:PRINT LE
1908 FOR I=1 TO 500:NEXT I:IF SC>HS THEN HS=SC
1910 FL=1:RETURN
2000 CV=1:X=128:Y=96:DI=4:KT=0:A(0)=128:B(0)=96:C=0:R=0:TL=0:TU=0:SC=SCL
2010 MAG0:PATTERNS#0,"81423C5AFF7E663C"
2020 RETURN
2100 SCREEN 2,2:CLS:COLOR10,1,(0,0)-(255,191),1:CURSOR104,8:PRINT "SERPENT":CURS
OR190,184:PRINT "SCORE":CURSOR170,176:PRINT "HI-SCORE";:PRINT HS
2110 Y=8:FOR X=16 TO 88 STEP 8:GOSUB 700:NEXT X
2120 Y=8:FOR X=152 TO 248 STEP 8:GOSUB 700:NEXT X
2130 Y=176:FOR X=16 TO 162 STEP 8:GOSUB 700:NEXT X
2150 X=16:FOR Y=8 TO 168 STEP 8:GOSUB 700:NEXT Y
2160 X=248:FOR Y=8 TO 168 STEP 8:GOSUB 700:NEXT Y
```

```
2170 FOR I=1 TO LE*4:GOSUB 2200:GOSUB 2220:GOSUB 2200:GOSUB 2230:GOSUB 2200:GOSU
B 2240:NEXT I:RETURN
2200 DEF FNS(G)=INT(RND(1)*G)*8
2210 Z=FNS(30):Q=FNS(22)
2211 CP=VPEEK(INT((Q+3)/8)*256+INT(Z/8)*8+(Q+3)MOD8)
2212 IF CP=24 THEN 2200
2213 IF CP=126 THEN 2200
2215 IF CP=28 THEN 2200
2216 IF Z<240RZ>240 THEN 2200
2217 IF Q<160RQ>168 THEN 2200
2218 IF Z=1280RQ=96 THEN 2200
2219 RETURN
2220 GOSUB 550:GOSUB 600:RETURN
2230 GOSUB 550:GOSUB 1000:RETURN
2240 GOSUB 550:GOSUB 1070:RETURN
2280 RETURN
3000 A$=INKEY$:IF A$="" THEN 3000
3010 IF A$="Y" THEN RETURN
3020 IF A$="N" THEN END
3030 GOTO 3000
4000 SCREEN 2,2:CLS:COLOR1,9,(0,0)-(255,191),9
4010 Y=20:X=60:MAG1:PATTERNS#0,"81423C5AFF7E663C":CL=1:SV=1:MC=1
4020 READ A,B,C,D
4030 LINE (A,B)-(C,D),CL
4040 IF A=1 THEN 5000
4050 GOTO 4010
4060 IF SV=1 THEN SPRITE 0,(X,Y),0,4:X=X+20:IF X=200 THEN SV=2:Y=110
4070 IF SV=2 THEN SPRITE 0,(X,Y),0,4:X=X-20:IF X=60 THEN SV=1:Y=20
4080 A$=INKEY$:IF A$=CHR$(32) THEN  SOUND0:RETURN
4090 IF MC=1 THEN 4130
4100 IF MC=2 THEN 4140
4110 IF MC=3 THEN 4150
4120 IF MC=4 THEN 4160
4130 MC=2:SOUND1,147,15:SOUND2,880,15:GOSUB 4200:SOUND2,831,15:GOSUB 4200:SOUND2
,698,15:GOSUB 4200:SOUND1,175,15:SOUND2,880,15:GOSUB 4200
4135 GOSUB 4200:SOUND2,880,0:GOSUB 200:SOUND1,220,15:GOSUB 4200:GOSUB 4200:GOTO
4060
4140 MC=3:SOUND1,147,15:SOUND2,831,15:GOSUB 4200:SOUND2,880,15:GOSUB 4200:SOUND2
,698,15:GOSUB 4200:SOUND1,175,15:SOUND2,831,15:GOSUB 4200:GOSUB 4200
4145 SOUND1,220,15:SOUND2,880,15:GOSUB 4200:GOSUB 4200:GOTO 4060
4150 MC=4:SOUND1,131,15:SOUND2,784,15:GOSUB 4200:SOUND2,698,15:GOSUB 4200:SOUND2
,659,15:GOSUB 4200:SOUND1,165,15:SOUND2,784,15:GOSUB 4200
4155 SOUND2,698,15:GOSUB 4200:SOUND1,196,15:SOUND2,659,15:GOSUB 4200:SOUND2,523,
15:GOSUB 4200:GOTO 4060
4160 MC=1:SOUND1,147,15:SOUND2,587,15:GOSUB 4200:GOSUB 4200:GOSUB 4200:SOUND1,17
5,15:GOSUB 4200:GOSUB 4200:GOSUB 4200:SOUND0:SOUND1,220,15:GOSUB 4200GOSUB 4200:
GOSUB 4200:SOUND0:GOTO 4060
4200 FOR I=1 TO 25:NEXT I:RETURN
4210 FOR I=1 TO 20:NEXT I:RETURN
4300 DATA 40,40,45,45,45,45,38,53,38,53,48,62,48,62,39,73,39,73,34,69,34,69,41,6
2,41,62,30,50,30,50,40,40
4310 DATA 67,42,79,50,79,50,76,54,76,54,70,50,70,50,67,54,67,54,70,52,70,52,68,6
0,68,60,65,58,65,58,62,62,62,62,72,68,72,68,70,72,70,72,55,62,55,62,67,42
4320 DATA 85,45,103,39,103,39,100,53,100,53,110,58,110,58,108,62,108,62,97,60,97
,60,102,68,102,68,99,71,99,71,85,45,97,45,95,52,95,52,90,47,90,47,97,45
4330 DATA 131,43,144,59,144,59,126,64,126,64,118,74,118,74,114,71,114,71,131,43,
131,51,136,57,136,57,128,58,128,58,131,51
4340 DATA 160,38,164,42,164,42,155,49,155,49,159,52,159,52,163,49,163,49,165,52,
165,52,160,57,160,57,165,63,165,63,172,58,172,58,176,60,176,60,164,71,164,71,147
,50,147,50,160,38
4350 DATA 180,70,192,53,192,53,199,60,199,60,210,51,210,51,199,69,199,69,191,62,
191,62,183,73,183,73,180,70
4360 DATA 217,45,235,44,235,44,235,50,235,50,229,51,229,51,229,70,229,70,224,70,
224,70,224,51,224,51,217,51,217,51,217,45,1,1,1,1
4500 SCREEN 1,1:CLS
4505 PRINT "YOU CONTROL A HUNGRY SERPENT":PRINT
4510 PRINT "YOU CAN ONLY TURN LEFT OR RIGHT"
4520 PRINT "ACCORDING TO YOUR DIRECTION OF"
4530 PRINT "TRAVEL":PRINT
4540 PRINT "USE THE ";CHR$(142);" KEYS TO GO LEFT AND RIGHT"
```

```
4560 PRINT "TO GO LEFT OR RIGHT ":PRINT
4570 PRINT "EAT FRUITS FOR POINTS BUT AVOID"
4580 PRINT "EATING YOUR OWN DROPPINGS OR HITTING"
4590 PRINT "THE WALLS OR YOU WILL EXPLODE":PRINT
4595 PRINT"YOU ALSO SCORE POINTS EACH TIME YOU"
4596 PRINT "GROW":PRINT
4600 PRINT "HITTING A MAGIC MUSHROOM CAUSES"
4610 PRINT "STRANGE THINGS TO HAPPEN-YOU WILL"
4620 PRINT "CHANGE DIRECTION AND A PART OF YOUR"
4630 PRINT "BODY WILL SNAP OFF-BEWARE"
4635 PRINT :PRINT "[ HOLD DOWN SPACEBAR TO PLAY ]"
4640 A$=INKEY$:IF A$=CHR$(32) THEN RETURN
4650 GOTO 4640
5000 PAINT(40,45),1:PAINT(65,50),1:PAINT(95,55),1:PAINT(130,50),1:PAINT(155,45),
1:PAINT(190,60),1:PAINT(225,50),1
5010 CURSOR30,154:PRINT"HOLD DOWN SPACE BAR TO PLAY":GOTO 4060
5100 MES$(1)="GOOD":MES$(2)="VERY GOOD":MES$(3)="EXCELLENT"
5110 MES$(4)="FANTASTIC":MES$(5)="INCREDIBLE":RETURN
```

PROGRAM OF THE MONTH

Each issue, we would like to invite Sega owners to send in their programs for evaluation and possible entry into the Magazine. Not only games programs, but utility programs, and machine code or basic routines. Each program will be judged on originality, usefullness, or play value, graphic content or design and layout, individually depending on its use or function.

Any programs which are considered to be commercially viable, separate contact will be made to agree terms. All other programs accepted for inclusion into the magazine, will receive a cassette program of the writers choice.

One program each issue will be singled out as being the "Star Program of the Month", entitling the writer to select any one of Sega's game cartridges as their prize.

ADDENDUM

The following two lines should put right any problems you have been having with last issues INDY GP program due to a typesetting error. Some of program was missed and we apologise for any inconvenience caused.

```
290 PRINTCHR$(16):FORI=15TO0STEP-.5:SPRITE0,,0,6:SOUND1,110,I:SPRITE0,,1,10:SOUND4,0,I:NEXTI

380 COLOR13:CURSOR85,90:PRINT"STRIKE A KEY TO                    START"
```

Teach Yourself Basic Games Programming

A comprehensive guide to the fascinating world of games programming in Basic. This useful package contains a 60 page booklet crammed with information, demonstration programs and helpful hints. Comes complete with cassette program to run on 16 or 32K SC3000.

NOW AVAILABLE

SEGA® Beginners Guide

Take the frustration out of getting to know your Sega computer. Contains practical advice on all aspects of using the SC3000 and its peripheral attachments. Written in New Zealand, this book is a must for Sega owners unfamiliar with the basic computer language and explains in simple terms how to get the best from your favourite micro.

The Hard Word from Sega

SEGA'S new hard keyboard and new hard disk drive, make Sega even harder to beat.

The very latest industry standard 3'' diskettes store a massive 328K of information each desk, and operate at speeds far in excess of their older 5½ inch counterparts.

Sega's super control station will also enable you to connect to many other attachments such as business printers electronic typewriters even telephone moderns, through it's two additional 1/0 ports, for RS232C and parrallel interfaces.

Sega now boasts not only the most sophisticated and easy to learn basic language, the most incredible graphic capabilities, a sensational range of cartridge and cassette games, superb education and applications software written by and for New Zealanders, but also a very sophisticated, fast, small business system, with the state of the art Sega control station.
Feel the difference, see the difference, compare the difference, and you too can enjoy the difference of Sega superiority.
Call at any Sega stockist for a demonstration or contact Grandstand Leisure at: Box 2353 AUCKLAND

GRANDSTAND
SEGA®