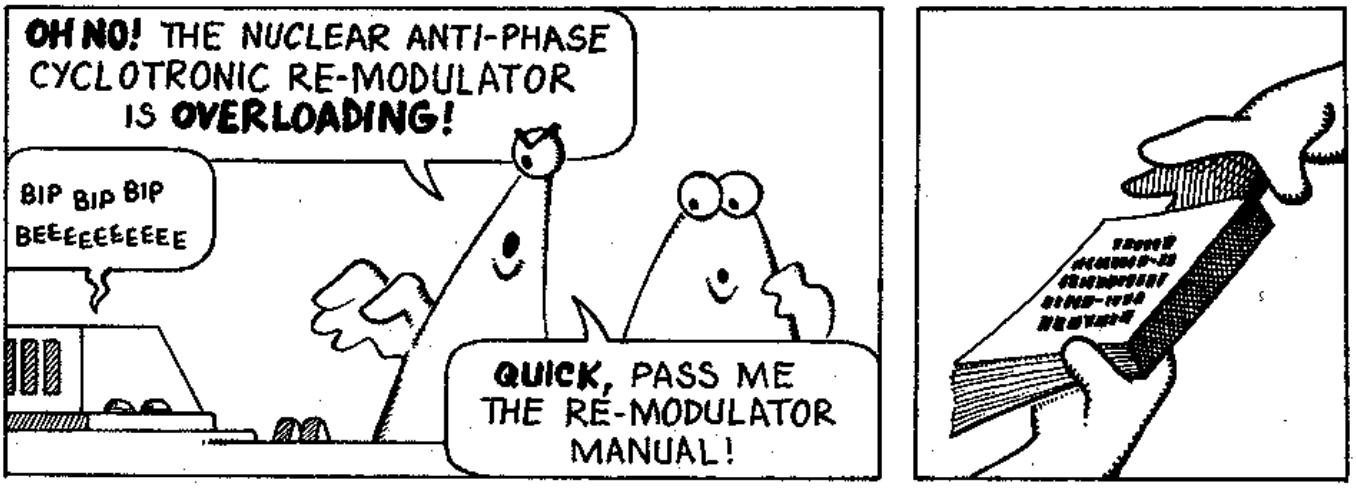
Vol 3 – No 11. July 1990.

FORMAT

FOR SPECTRUM AND SAM USERS



Manuals Are Useful Sometimes
FORMAT IS USEFUL ALL THE TIME

SPECTRUM AND SAM SOFTWARE

WITH FULL MONEY BACK GUARANTEE AND UPDATE SERVICE

SAM COUPE SOFTWARE

SAMTAPE2 MAKES SAM ACT AS A SPECTRUM - LOAD IN THEN SAVE MOST PROGRAMS TO DISK + PEEK/POKE, ETC. VERY EASY TO USE. 2ND PROGRAM INCLUDED FOR SPECCY UTILITIES - ALL KEYS WORK, LPRINT, LLIST + MUCH MORE! TASWORD, + MANY MORE PROGRAMS WORK AS NORMAL. THOUSANDS OF PROGRAMS NOW WORK ON A SAM WITH DISK. **£9.99**

SAM ASSEMBLER2 OUR Z80 ASSEMBLER HAS BEEN COMPLETELY REWRITTEN FOR SAM - 1ST RATE FULL SCREEN EDITING, 64 COLUMN, SCRATCH/HELP PAGE, MULTIPLE SOURCE FILES (64K) TOKENISED, 32K FOR OBJECT CODE, HEX/DEC/BIN, V.FAST, PRINTER ETC, ETC. TAPE OR DISK. **UNRIVALLED £8.99**

SAM ADDRESS MANAGER SAVE ADDRESSES OF ALL YOUR FRIENDS OR CUSTOMERS. PRINTS LABELS, M/CODE SEARCH+AMEND (TO DISK), UP TO 5000 ADDRESSES PER DISK. CAN INPUT CASH+PRODUCT CODE. UNSECURED. **£7.99**

SPECTRUM SOFTWARE

TAPE UTILITY D - TAPE TO TAPE BACKUP OF MOST SPECTRUM PROGRAMS - EVEN CONVERTS MANY FOR RELIABLE LOADING! UNSECURED. **£8.50** (PROGRAM INCLUDED IN TRANSPACK2)

TRANSPACK 2 - UNSECURED + COPIES TO DISK. FOR TAPE TO DISK TRANSFER - MANAGES 1000s OF PROGRAMS INCLUDING CLICKY + FIRE*, + COUNTDOWN LOADERS - NOT FOR NOVICE. **£11.99.** INFORMATION BOOKS 1,2,3 **£2.50** EACH - GIVES DETAILS OF 100s OF TRANSFERS THAT TP2 CAN'T DO AUTOMATICALLY.

Z80 TOOLKIT2 - "BRILLIANT PACKAGE" - *YS 7/88*. UNRIVALLED M/CODE PACKAGE INCLUDES SINGLE STEP TO UNDERSTAND + DEBUG M/CODE, DISASSEMBLER, A REVERSE ASSEMBLER, PLUS OUR UNIQUE ASSEMBLER WITH FULL SCREEN EDITOR, HEX/DEC, MONITOR, PRINTER, ETC. UNSECURED. **£8.99**

PLUS3MATE - 5 PROGRAMS THAT CAT YOUR DISKS, REPAIR/VERIFY, SECTOR EDITOR, MULTIPLE COPY + ERASE + UNERASE, SUPERB FILE COPIER, CLONE DISKS (INC MANY PROTECTED), BOOT PROGRAM + MUCH MORE **£12.99** ON DISK. **MICROMATE** - SIMILAR TO +3MATE BUT FOR M/DRIVE. FORMATS CARTRIDIGES TO APPROX 100K, REPAIRS FAULTY PARTS, ETC. ON CART FOR ONLY .**£11.99**

U.K POST FREE +£1 EUROPE, +£2 REST. WHEN ORDERING STATE COMPUTER.

LERM, DEPT F, 11 BEACONSFIELD CLOSE, WHITLEY BAY, TYNE AND WEAR. NE25 9UW. TEL (091)2533615. Prop L.Evans.

CONTENTS

Vol.3 No.11. July 1990.

FORMAT is published by **FORMAT PUBLICATIONS.** 34 Bourton Road, Gloucester, GL4 0LE, England. Telephone 0452-412572.

Printed by D.S.LITHO. Gloucester. Telephone 0452-23198.

NEWS ON 4

RED SPECTRUM

Rumours are floating around that a Russian based company have designed a Spectrum emulating micro. Reports that over 15,000 have been delivered to Soviet schools could not be confirmed by the Soviet embassy in London.

However New Computer Express has published a photo so there must be something in the story. More details as they emerge.

DEFENDERS OF THE EARTH

Enigma Variations have launched their first SAM Coupé title called "Defenders of the Earth". The program is an arcade game and nearly fills the full 256k of memory on a basic machine. Mode 4 screens are used so the graphics rival those of an ST and is therefore the first arcade game to really exploit the Coupé's extra colour resolution.

The program is available by mail order. It comes on tape at £11.99 or on disc at £14.99 from Enigma Variations, 13 North Park Road, Harrogate, North Yorkshire, HE1 5PD.

LERM PLEDGE CONTINUED SAM SUPPORT

LERM Software have promised continued support for the SAM Coupé and at the same time announced new software. Both of their existing SAM programs have now been upgraded with SAM Assembler 2 now having a 64 column screen, an even better full screen editor and a built-in disassembler.

Also released is their new SAM ADDRESS MANAGER for disc users which is able to store up to 200 records per file and 25 file per disc. The program should find many uses, both in the home and small business, storing addresses or other information. It costs £7.99 and their advert can be found elsewhere in this issue.

SUGER'S STORY

For anyone thats interested (if anyone is) the Amstrad story is told in full in a new book about the life and time of Alan Suger.

Called 'ALAN SUGER - THE AMSTRAD STORY' the book is written by David Thomas and is published at £14.99 by Century. The book is based on interviews with Suger and his cohorts and is said to reveal the inner secrets of Amstrads success from his early Hi-Fi days to the troubles with his IBM clones.

24 PIN FROM CITIZEN

Citizen is well known for it's excellent 120D printer which has been one of the best value for money 9 pin printers for some time. Now it looks set to do the same for 24 pin printers. Later this month the company will launch a new budget-priced 24 pin printer onto the UK market. Prices are expected to be below £250 and may even break the £200 barrier.

SHOW TARGETS EUROPEANS

The next All Formats Computer Fair is being clearly targeted at European computer users. Extensive publicity, pointing out that just one or two judicious purchases could pay for their trip to Britain, is being placed in over 20 continental magazines and newspapers. The British Tourist Authority are also sending details to all their European offices.

URGENT we need your news. Clubs, Shows, New Releases, anything you think other people should know about. If you have any news items you want to pass on then send them in. Please mark the envelope NEWS in the top corner.

DON'T PANIC, famous words to fans of the Hitchhikers Guide to the Galaxy, and very applicable at this time to SAM Coupé owners. As most of you will have heard by now Miles Gordon Technology have appointed an Administrative Receiver but things are not as black as they may seem.

The receiver has been called in to run the affairs of the company while a buyer was found for the SAM project. To make SAM the number one computer in the UK next Christmas required a large injection of capital. After long negotiations with their bankers MGT were unable to raise the extra funds (it seems to me that UK banks won't lend you money unless you can prove you don't really need it). So the directors started looking for a 'Big Brother' to come in and fund the growth nessesary to ensure the success of the Coupé. The receiver was appointed to see the company over this transition.

At the time of writing this editorial (26th June) MGT are reasonably certain that agreement (at least in principle) will be reached with a buyer in the next 7 to 10 days. At one point there were at least twelve interested parties and at the moment there seems to be two front runners. Both Alan Miles and Bruce Gordon have made it clear that they want to continue to work with any new owners. They have also expressed their faith in the long-term success of the SAM Coupé and with that I whole-heartedly concur.

Until MGT (or it's successors) are back on line FORMAT will, of course, provide what help we can to SAM users. If you know any SAM Coupé owners who are not FORMAT subscribers then point them in our direction. We will be trying to arrange a meeting, as soon as the new management is appointed, to try and clarify the position of existing users regarding guarantees. By the time I write the August issue I firmly beleive there will be better news.

Moving on, I would like to welcome Karen Kemley to the staff. Karen has bravely taken on the task of keeping the office paperwork in order - not an easy thing now there are over 2500 subscribers. This will free me for more productive work, including producing a bigger FORMAT each month. Work on the SAM projects I have mentioned in recent issues has not stopped, the first book and the printer interface are nearly finished. It is difficult to say exactly when they will appear on the market but it should be soon.

By the way, I'm sorry this months FORMAT is a little late. But with the crisis at MGT and my holiday at the start of the month things just got behind schedule. I will try to catch up by next month.

Final item for this month, UK and European readers will find a complimentary ticket to the next All Formats Show included with this issue. Its for use on the second day, Sunday 5th August. I would like to thank the organizer, Bruce Everiss, for kindly agreeing to supply the tickets. Please come along and help to make this the biggest show yet.

Until next month,

Bob Brenchley. Editor.

SHORT • SPOT

By:- John Wase.

Let's start this month with some real shorties. Les Fraser of Hackbridge, Wallington, Surrey, mentions that if you can't stop an autorun program by merging, then this might be the solution. (He doesn't mention what this is working on, but I assume it's a PLUS D).

a) Take a newly formatted disc - not even a system file!

b) SAVE program onto it. It's now on the first and following sectors.

c) Key in: LOAD @ 1,4,1,25000 : POKE 25007,255: POKE 25008, 255: SAVE @ 1,4,1,25000

And this should have removed the autorun. Mind you, if it's autorunning, you've got to stop it first.....

He also has a tip about catalogues. If, like Les, you keep lots of programs which are combinations of Basic and code on one disc, here's a way to simplify things. Put only the names of the Basic Loader-programs in CAPITAL LETTERS. Save everything else with lower case. Now use "Cat-sort" from Hugh McLenaghan's "Hack Zone", "Format", June 1989. This sorts the catalogue: upper case comes first; lower case follows. Next, use Hugh's "Hide files" routine from the same volume of "Format". Now, cataloguing the disc gives you only the loader programs, in alphabetical order. Finally, save the following program as "Autocat" LINE 10.

```
10 CAT*
20 INPUT "Load Program ","P";LINE A$
30 LET a=VAL a$
40 LOAD pa
```

Simple, innitt?

Dave Marriott of Long Eaton, Nottingham, wrote, suggesting that if you load a CODE file on SAM, the size is available in DPEEK DVAR 161 and DPEEK DVAR 189. I'm not sure how this works, but my guess is that if it works, it's more by luck than judgement. DVARs go only from 0 to 7 (as in the disc operations supplementary manual). DPEEK only gives a double-length PEEK; the lengths of a file, though, would be three bytes (see page 79 of the Technical Manual [Version 3]). And finally, even if his PEEKs work, because they are not within the fixed DVAR area, they may not always work, and they may not work on all versions of DOS.

This still leaves us with a mystery, though. Just what <u>is</u> he PEEKing? And just how <u>does</u> it work? Anyone know?

Andrew McPhee of High Spen, Rowlands Gill, Tyne and Wear (what lovely names they have up there, Andrew) wrote a long letter in which he said that having read the article mentioning "Mathographics", he decided to write a program for SAM using mathematical curves and all 128 colours. He called it "fireworks", because the display looks like fireworks - all the individual pixels seem to twinkle as the colours change. Here's the program:-

```
10 REM ***SAM FIREWORKS***
20 CLS #
30 PRINT AT 1,0;" THIS PROGRAM IS FO
   R SAM COUPE"'"OWNERS EVERYWHERE.
   "'"IT SHOWS SEVERAL MATHEMATICAL
   "'"CURVES AS A DISPLAY. (THE 128"
   '"COLOURS OF THE COUPE ARE"'"EXPO
   SED.)"
40 PRINT AT 18,8;"PRESS ANY KEY"
50 PAUSE : CLS #
60 PLOT 100,30: DRAW -2,-5: DRAW 4,0
   : DRAW -2,5
```

```
 70 PLOT 140,30: DRAW -2,-5: DRAW 4,0
    : DRAW -2,5
 80 FOR a=70 TO 41 STEP -4.5
 90 FOR x=0 TO 1.2*a
100 LET r=2*a*x/40
110 PLOT x+140,-(x^2)/40+r+30
120 PLOT 140-x,-(x^2)/40+r+30
130 PLOT x+100,-(x^2)/40+r+30
140 PLOT 100-x,-(x^2)/40+r+30
150 PALETTE (RND*14)+1, RND*127
160 PEN (RND*14)+1
170 NEXT x
180 NEXT a
190 INPUT "Press e To End, r to RE-
-RUN";AM$
200 IF AM$="r" THEN RUN
210 IF AM$="e" THEN STOP
220 GOTO 190
```

Incidentally, if it gets a bit wearisome; all that starting and stopping, just insert line 185 as follows:-

185 GOTO 60 (An inelegant Wase addition).

Those who read Popular Computing Weekly might see a comment about cassette recorders which I feel is worth repeating. I received a tape to review (from LERM). It failed to load on my trusty old Evesham Market tape recorder. Irritated, I rang up LERM and demanded a disc. I also rang up MGT and got the names of their recommended cassette recorders (Sony TCM818, Lloytron V188 and Philips D6260). The nice man at King's Heath Currys (I was in Birmingham at the time) had just had a batch of the Sonys in (£24.99), and sold me one with my plastic card. I was a bit perturbed to see it had no head adjuster, but vowed that if it didn't load, SAM would return to Swansea in a hurry. The tape loaded first time. Sheepishly, I sent back the disc to LERM....

This, of course, accounts for the wad of SAMstuff in the middle. However, I'll try to spread it out a bit...

The next item concerns printers: it's regrettably anonymous because I can't find the name to add to it. More of my carelessness. The problems of dealing with graphics and with downloading characters are two of the most persistent queries - not surprising, since no two printers seem to be alike in this area. This item concerns seven-pin printers, lots of which have recently been advertised very cheaply. The writer, who has a Seikosha GP100A mentions that the impression is given in the PLUS D manual (p.18, para 3) and in FORMAT (Dec'88, p.13, para 2) that Spectrum graphics can't be printed with this type of printer; well they can. There are some limitations, but standard graphics (ASCII 129-143) can be printed in modified form, and UDGs can be created which will print correctly if the following points are noted.

a) The first pixel line of a UDG must be 255 (BIN 11111111) for the printer to operate.

b) The lines must be in upside down order.

Here is a listing for printing various symbols in sets of five (the Pi comes from page 89 of the +2 manual).

```
 10 REM "7 PIN UDGS"
100 LET a=200
110 FOR n=0 TO 7: READ d: POKE USR "A
    " +n,d: NEXT n
120 FOR n=1 TO 7: LPRINT CHR$ 144;: N
    EXT n 125 LPRINT CHR$ 27;CHR$ 15;
130 LPRINT " DATA from line ";a
140 LET a=a+10: IF a=270 THEN STOP
150 GOTO 110
200 DATA 255,0,20,20,84,60,2,0
210 DATA 255,255,0,20,20,84,60,2
220 DATA 255,60,0,60,0,60,0,60
230 DATA 255,8,20,34,65,128,65,34
240 DATA 255,240,240,240,240,15,15,15
250 DATA 255,255,255,255,255,255,255,
    255
260 DATA 255,130,124,108,84,108,124,1
    30
```

Finally he mentions that as it is a bind to add up the data for each UDG line, he gets the computer to do it for him. It is not fully foolproofed to keep it short: multiple entries of the same letter will fool it. The principle is obvious and applies equally to all printers (well, I'm shortly going to get a 24 pin, and this is a bit different, and as for the lasers...)

```
 10 REM BIN ADDER
 20 REM Adds Binary nos for UDG lines
 30 REM Enter only filled-in pixels
100 LET b=0
110 PRINT "Enter letter for filled-in
    pixel":GOTO 130
120 PRINT "Enter next letter."
130 PRINT "(SPACE to end line - X to
    break)"
```

Malcolm Goodman of Leeds (who has sent in several items) offers an ultra-fast block move in assembly language, to go with the display file clear routine in the June issue. He mentions that the entry routines could possibly have a few T-states shaved off their running time, but that he can't think of any way that the core of the routine can be made any faster. Now, I know that all FORMAT readers like to have a challenge...

```
Label Instruction T-states Comments

10      DI          4     Interrupts off
20      LD BC,6912  10    Counter
30      LD DE,16384 10    First Screen Location
40      LD HL,nn    10    Where Screen Stored
50 Loop LDI         16    Single Block Move
60      LDI         16    "
70      LDI         16    "
80      LDI         16    "
90      LDI         16    "
100     LDI         16    "
110     LDI         16    "
120     LDI         16    "
130     LDI         16    "
140     LDI         16    "
150     LDI         16    "
160     LDI         16    "
170     JP PE,Loop  10    Repeat until BC=0
180     EI          4     Interrupts on
190     RET         10    Exit
```

Malcolm points out that the routine is fairly obvious in action. The core of the routine (lines 50-170) performs the block move and test in 202 T-states, and is repeated 6912/12 times (i.e. 576) giving a total execution time of 116400 T-states. Use of a single LDIR instruction instead would result in an execution time of about 145200 T-states. So the listing above is about 20% faster - at the expense of RAM! Further LDI instructions can be added to give a little more improvement, but it is important to note that the number of LDI instructions MUST be an integral factor of 6912, otherwise the P/V flag test in line 170 will be wrong. Finally, in line 40, the "nn" must hold the address of where the SCREEN$ is held in memory.

Although this routine and the one in the June issue were written for the Spectrum, it should be possible for them to work on the SAM with a few odd adjustments.

Talking of speed, W. Ettrick Thomson (who is a frequent correpondent) writes about my "bench tests" which showed up differences in speed between SAMMY and SPECCY. I'm afraid I didn't make myself very clear in past articles and Ettrick (quite rightly) has picked me up about my comments on integer arithmetic. I really can't do better than quote directly from his letter, which applies certainly to the Spectrum ROM, and probably also to that of SAM:-

"When a number is first presented to a program, as (1) a program constant, or (2) a constant that is the whole, or part, of an INPUT expression, or (3) a constant that is the whole or part of the operand of VAL, it is converted from denary to binary to give a 5-byte number. If the number is an integer in the range plus or minus 65536, it will be in integer format (in which the first and fifth bytes are zero); if it is an integer outside these limits, or if it has a fractional part, it will be in floating-point format.

The arithmetic operations +,-,and *

first inspect the operands; if they are both in integer format, then integer arithmetic is used, and the result will be in integer format; except that if the result would be outside the integer-format limits (which will be detected in the integer-arithmetic routine), the operands are converted to floating-point format and f-p arithmetic used to give a f-p result. Thus LET a=30000, b=40000, c=a+b would store a,b in integer format but c would be in f-p; the conversion of 30000 and 40000 to f-p would not affect the format of a,b in the variables area.

If one or both operands are in f-p, then (after integer/fp conversion if necessary) f-p arithmetic is used to give an f-p result. The division operation, /, always uses f-p arithmetic, with integer/f-p conversion if necessary.

The result of a f/p operation may represent an integer within the integer-format limits, but it will be left in f-p format - with two exceptions: (1) if the result is zero, (which may arise through underflow, the real result being smaller in absolute magnitude than the limit 2 to the power of minus 128), this is put in integer format, 5 zero bytes; (2) the result of INT will be put in integer format if within the integer-format limits. This is the Spectrum situation; in BetaBasic, the result of MOD is also put in integer format if possible; presumably this will also apply to SAM's MOD, DIV.

With SAM, it is very easy to inspect the format of a variable; here is a procedure:-

```
DEF PROC formt x
 LOCAL a: LET a=LENGTH(0,x): PRINT x
 FOR a=a TO a+4: PRINT PEEK a;" ";
 NEXT a: PRINT
END PROC
```

If the first byte is zero, then you have integer format.

Here are a couple of "Bench tests" to show the speed differences:-

```
10 LET x=0: LET y=4
20 FOR i=1 TO 5000
30 LET x=x+y*(y-1)
40 NEXT i
50 PRINT x
```

Integer format, integer arithmetic, for SAM (10 LET x=0, y=4 suffices) and SAM's time is 22 seconds.

Changing lines 10 and 20 to:-

```
10 LET x=0, y=4/1
20 FOR i=1/1 to 5000/1
```

gives everything the same values (except for the initial value of x and the constant "1" of line 30), but floating-point format, floating-point arithmetic, with conversion of the constant "1" from integer to floating-point each time line 30 is obeyed: time on SAM, 33 seconds.

Finally, instead of using a stop-watch to time the operations, try the internal clock.

Start stop-watch:-

```
DPOKE SVAR 632,0: POKE SVAR 632+2,0
: REM must be done in this order
```

Read stop-watch:-

```
DEF FN lo=DPEEK SVAR 632 DEF FN hi=
PEEK (SVAR 632+2)
LET L1=FN lo, h1=FN hi L2=FN lo: IF
 L2>L1 THEN LET t=(L1+65536*h1)/50:
 ElSE LET t=(12+65536*FN hi)/50
```

't' is in seconds: essentially the same as the Spectrum routine in Chapter 18 of the Manual.

Many thanks for the clarification, Ettrick.

Please keep contributions coming in to:-

John Wase,
Green Leys Cottage,
Bishampton,
Pershore,
Worcs,
WR10 2LX.

STEVE'S SOFTWARE

SC_ASSEMBLERis a powerful Editor Assembler specially written for the SAM COUPE 256K Computer with 1 or 2 Disc Dives. SC_Assembler allows you to create upto 20K Machine code programs with the help of a friendly 64 column editor system.
The editor will except source like no other Spectrum Assembler, First there is no need to type spaces between opcodes and labels no need to tab and field source to certain parts of a line.
For example you could enter a source line like :-
10label:ldhl,16384 or 10 La BE 1 :L dH 1, 163 8 4
When you press Enter, the line is reprinted like this :-
10 label: LD HL,16384
Not only is it quicker and easier to enter source, SC_ASSEMBLER will compress the source so it takes up half the space compared to other Spectrum Assemblers,with 96K of source space availiable it's possible to store around 10,000 lines of source code.
The editor allows you to edit the whole screen of source, with the Function keys you can bring more lines of source to the screen in so many different ways allowing for faster editing.
If you happen to type somthing wrong then helpful worded error messages are displayed, you don't get messages like ERROR 02.
Assembling source on Pass 1 produces helpful info on where your code starts,ends,length and number of labels used before even going onto pass 2.
SC_ASSEMBLER with the built in Dissasembler will recognize the Full set of 798 opcodes, including the 102 Undocumented ones.
SC_ASSEMBLER is a massive 40K code program designed not to use any of the SAM ROM routines, to ensure this Quality product will be compactable with different ROMs that may appear in the Future

SC_ASSEMBLER costs £10, available from :- STEVE'S SOFTWARE
7 NARROW CLOSE, HISTON, CAMBRIDGE, CB4 4XX.
Please make cheques payable to MR S.J.NUTTING

A 512K version Assembler will be availiable around September. This will surport the extra 256K memory, useful SAM source files built in Monitor and a very large DTP produced Manual with lots of Technical information for Advanced Users, at a cost of £15.
If you purchase the 256K version now you can upgrade to the 512K version for £5, so you don't lose out buying now.

KEYIN

SAM's Program Writer

By:- Carol Brooksbank.

I have always hated writing code pokers. I write machine code in hex, using an assembler, so when I am writing a machine code program to be printed in FORMAT, I have no difficulty in producing the source code listing. The assembler does it for me.

But not everyone who reads FORMAT has an assembler or understands hex, so there has to be a code poker so that anyone can use the program. I am always terrified that somewhere, when I am translating the bytes into decimal and listing them in DATA lines, I shall make a mistake, hit a wrong key, or miss something out. Do any of those, and the reader who types the code poker in has a useless program that either doesn't do what it is supposed to do, or crashes, or both. So every code poker I write costs me hours of checking and double checking. I am quite neurotic about them. And I have never been able to bring myself to include checksums, because it just adds more places where you can make mistakes.

Musing through SAM's handbook the other day, I came across the following:-

> **KEYIN a$**
>
> enters a string as if you had typed it in yourself, so that it allows programs to write themselves!

What on earth could that mean? Then it dawned on me. It means - among other things - that SAM can write code pokers.

KEYIN lets SAM write extra lines in a program as it is running, and it is especially useful for adding or updating DATA lines. The code poker writer program illustrates how it works. Type in this program.

LISTING 1.

```
 10 LET Q=130
 20 FOR A=firstbyte TO lastbyte STEP
    8
 30 LET check=0
 40 LET A$=""
 50 FOR B=0 TO 7
 60 LET check=check+PEEK (A+B)
 70 LET A$=A$+STR$ PEEK (A+B)+","
 80 NEXT B
 90 LET B$=A$+STR$ check
100 KEYIN STR$ Q +"DATA" +B$
110 LET Q=Q+10
120 NEXT A
```

You will have to add a LINE 5, which sets up the 'firstbyte' and 'lastbyte' variables:-

```
5 LET firstbyte=.....,lastbyte=.....
```

In line 20, I have used STEP 8, but you can change this to suit yourself. It should, if possible, be some number which divides exactly into the number of bytes you want to list. If not, you will have to list a few extra, and delete the unwanted ones from the last line - and recalculate the last line checksum. For instance, if you want to list 81, bytes, STEP 8 will miss the last one off. STEP 9 would list them all, or you could list 88 using STEP 8, and delete the last 7.

The inner B loop, which must be changed to match the STEP in line 20 - e.g. B=0 TO 8 will match STEP 9 - divides the bytes into groups which will make up the DATA lines. Line 60 keeps a running total of the bytes in the line, for the checksum, and line 70 writes them into a string, each byte separated by ",". Line 90 adds the checksum (a simple running total of the value of the bytes) onto the

list of bytes.

Now comes SAM's clever part. In line 10, Q was initiated at 130, the first unused line number. Line 100 actually writes a new line into the program, consisting of the line number (STR$ Q), "DATA" and the string we have just constructed - the data bytes separated by "," plus the checksum. This line will overwrite any other line already there, so unless you want to do that, be careful that Q holds a number beyond the end of the existing program.

Line 110 gives Q the next line number, and line 120 goes back for the next set of bytes.

If you add line 5, and then RUN the program and LIST it, this is what you will find.

LISTING 1B.

```
  5 LET firstbyte=60000,lastbyte=6008
    7
 10 LET Q=130
 20 FOR A=firstbyte TO lastbyte STEP
    8
 30 LET check=0
 40 LET A$=""
 50 FOR B=0 TO 7
 60 LET check=check+PEEK (A+B)
 70 LET A$=A$+STR$ PEEK (A+B)+","
 80 NEXT B
 90 LET B$=A$+STR$ check
100 KEYIN STR$ Q +"DATA" +B$
110 LET Q=Q+10
120 NEXT A
130 DATA 42,158,234,6,32,197,229,6,90
    4
140 DATA 8,17,150,234,126,18,36,19,60
    8
150 DATA 16,250,6,8,197,33,150,234,89
    4
160 DATA 175,87,6,8,203,22,203,18,722
170 DATA 35,16,249,58,160,234,71,122,
    945
180 DATA 205,129,1,16,250,193,16,228,
    1038
190 DATA 225,193,35,16,208,201,0,0,87
    8
200 DATA 0,0,0,0,0,0,0,0,0
210 DATA 0,42,158,234,203,28,203,28,8
    96
220 DATA 203,28,1,32,0,9,203,20,496
230 DATA 203,20,203,20,34,158,234,201
    ,1073
```

SAM has added the DATA lines for the bytes at the addresses specified in line 5, which were the bytes for my £Small is beautiful' machine code - SAM version.

Now, merge the following lines into the program:-

LISTING 2.

```
 10 LET Q=130
 20 FOR N=firstbyte TO lastbyte STEP
    8
 30 LET check=0
 40 FOR B=0 TO 7
 50 READ A
 60 POKE (N+B),A
 70 LET check=check+A
 80 NEXT B
 90 READ A
100 IF check<>A THEN PRINT "ERROR IN
    LINE ";Q: STOP
110 LET Q=Q+10
120 NEXT N
```

This will give you the finished code poker:-

LISTING 2B.

```
  5 LET firstbyte=60000,lastbyte=600
    87
 10 LET Q=130
 20 FOR N=firstbyte TO lastbyte STEP
    8
 30 LET check=0
 40 FOR B=0 TO 7
 50 READ A
 60 POKE (N+B),A
 70 LET check=check+A
 80 NEXT B
 90 READ A
100 IF check<>A THEN PRINT "ERROR IN
    LINE ";Q: STOP
110 LET Q=Q+10
120 NEXT N
130 DATA 42,158,234,6,32,197,229,6,9
    04
140 DATA 8,17,150,234,126,18,36,19,6
    08
150 DATA 16,250,6,8,197,33,150,234,8
    94
160 DATA 175,87,6,8,203,22,203,18,72
```

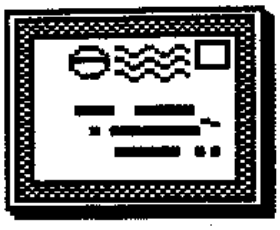
```
   2
170 DATA 35,16,249,58,160,234,71,122
   ,945
180 DATA 205,129,1,16,250,193,16,228
   ,1038
190 DATA 225,193,35,16,208,201,0,0,8
   78
200 DATA 0,0,0,0,0,0,0,0,0
210 DATA 0,42,158,234,203,28,203,28,
   896
220 DATA 203,28,1,32,0,9,203,20,496
230 DATA 203,20,203,20,34,158,234,20
   1,1073
```

We know it is accurate because SAM took the bytes directly from memory, from the working code, and SAM calculated the checksums. If anyone typing the program in makes a typing error or misses a byte out, the program will stop at line 100, telling them which line of data is incorrect.

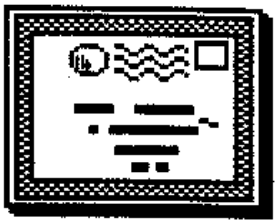
You may think that these programs are pretty useless to anyone who doesn't need to write code pokers. Not so. Suppose you want to copy a SAM ROM routine into your own machine code program - which is far better programming practice than calling a ROM address which may be changed in later versions of the computer.

Load in the first program, and specify the ROM start and end addresses in line 5. RUN the program, and the ROM bytes will be listed in the DATA lines. Merge the code poker lines, and change the line 5 addresses to those you need in your own program. RUN the code poker, and the ROM routine is copied to your program. All you need to do then is change any ROM addresses called to the correct ones for your program and you will have made an accurate copy of the ROM section you need with the minimum of checking and with no laborious copying at all.

KEYIN is a most useful command. You can use it anywhere in a BASIC program where you would like to change or update a line as the program proceeds. It can save hours of DATA line and subroutine writing - and, as a bonus, it means that from now on, you can rely on my code pokers!



YOUR LETTERS



Dear Editor,

In the January 1990 issue of FORMAT you published a program to load the Spectrum ROM into the SAM Coupé. I had already sold my Spectrum so I can't copy the ROM. Could you please copy the ROM to tape for me or tell me where I can buy a tape copy.

Yours sincerely, David Lines.

Sorry David but that's not possible. To use the Spectrum ROM you MUST own a Spectrum. If you do you have the right to use the ROM - even on another machine because the manufacturer never envisaged SAM coming along and therefore didn't impose a restriction on using the ROM image in this way.

If you don't own a Spectrum then you would be in breach of copyright if you had a copy of the ROM on tape or disc. Even if you had already copied the ROM, before you sold your Spectrum, you would have to erase ALL copies of the ROM as soon as you sold your computer. **Ed.**

Dear Editor,

After many years of using the Spectrum (following a ZX81) I have come to the conclusion that it does the job I want so there is no need to upgrade. However, I would like to get a better keyboard for my 128k (black Sinclair version). I can't find any adverts at the moment for replacement keyboards but I can use a soldering iron so it would be nice to convert a ready made keyboard for use with the 128. Can anyone help? There are lots of keyboards advertised in the mags so which should I get? What do I need to do to provide extra (single press) keys?

Yours sincerely, Mark Farmer.

I know little about keyboards but if one of our readers would like to write an article on the subject then FORMAT will be pleased to publish it. **Ed.**

Dear Editor,

I have just purchased a copy of the Tasword 2 program for the SAM Coupé. While it is better than nothing there are a few difficulties, one of which realy causes me problems. The auto-repeat rate for the keyboard is far to fast for my slow two-finger typing so I keep getting double or even triple letters appearing in the text. Can anyone help?

Yours sincerely, Alister Steel.

My version of TW2 seems to work OK so I can't reproduce your problem. Unlike the Spectrum TW2, poking the SAM system variable REPDEL(23561) doesn't affect the rate so they must be doing their own keyboad scan. If anyone knows how to slow things down then let's hear from you. **Ed.**

*** * STAR LETTER * * STAR LETTER * ***

Dear Editor,

I have been a computer enthusiast since the days of the Nascom, A name that probably means nothing to most modern users. But in those days, when the industry was young (and so was I) machines like it helped to establish the market for home computers.

OK, I know it was Sinclair, with the ZX80 and ZX81, that started the mass market but other machines produced the programmers that exploited Sinclair's excellent wares. A lot of early ZX81 and Spectrum programmers learnt their craft on the TRS-80. Back then there was a greater spirit of co-operation between computer users that seems to be missing now. Everyone used to write little games or utilities and send them off to magazines. Local clubs used to prosper. Where are these people now. I would like to hear your readers comments.

Yours sincerely, Mike Rand.

MORSE CODE TRAINER

For All Budding Radio Hams

Readers who are interested in learning Morse or improving their speed of receiving same might find this program of some use. First, type in and run this program:-

```
 20 CLEAR 49799: LET A=49800
 30 FOR L=1 TO 6 : READ D$
 40 FOR C=1 TO LEN D$ STEP 2
 50 LET X=CODE D$(C)-48-7*(D$(C)>="A"
    )
 60 LET Y=CODE D$(C+1)-48-7*(D$(C+1)>
    ="A")
 70 POKE A,16*X+Y: LET A=A+1: NEXT C:
     NEXT L
 80 DATA "DDE5DD2A4B5CDD7E06DDE5CB7F2
    00DCDA7C2DDE1DD23"
 90 DATA "FDCB016E28EADDE1C9F5F53E02C
    D0116F1D7F1FE2028"
100 DATA "372100C30100004E2346B928032
    318F7CB28C5282838"
110 DATA "02300B11A000215302CDB503180
    9113500215302CDB5"
120 DATA "0301801B0B79B020FBC118D8C50
    60318020601110051"
130 DATA "1B7BB220FB10F6C1C9"
140 LET N=65
150 FOR A=49920 TO 49970 STEP 2
160 POKE A,N: LET N=N+1: NEXT A
170 FOR A=49921 TO 49971 STEP 2
180 READ B: POKE A,B: NEXT A
190 DATA 6,17,21,9,2,20,11,16,4,30,13
    ,18,7,5,15,22,27,10,8,3,12,24,14,
    25,29,19
200 LET N=48
210 FOR A=49972 TO 49990 STEP 2
220 POKE A,N: LET N=N+1: NEXT A
230 FOR A=49973 TO 49991 STEP 2
240 READ B: POKE A,B: NEXT A
250 DATA 63,62,60,56,48,32,33,35,39,4
    7
260 FOR A=49992 TO 49999
270 READ B: POKE A,B: NEXT A
280 DATA 44,115,46,85,63,76,56,120
300 SAVE "TX CODE"CODE 49800,200
```

The BASIC should be saved in case of error and then the program run.

Now type in this program and save it. When run it will load in the machine code file produced above.

```
 1 CLEAR 49799
 2 LOAD "TX CODE" CODE
 3 DIM A$(360):DIM B$(36)
 5 INPUT "SPEED ?";W
10 POKE 49870,INT((315/32/W-INT(315/
   32/W))*256)
15 POKE 49871,INT(315/32/W)
20 POKE 49881,INT(840/W)
25 POKE 49891,INT(560/W)
30 POKE 49909,INT(1100/W)
35 FOR A=1 TO 36
40 LET N=65+(RND*25)
45 LET B$(A)=CHR$ N
48 IF INT(A/6)=A/6 THEN LET B$(A)=CH
   R$ 32
50 NEXT A
60 LET A$=(B$+CHR$ 255)
65 PRINT "PRESS A KEY TO TRANSMIT":
   PAUSE 0: CLS
70 RANDOMISE USR 49800
```

Lines 10 & 15 set the BEEP duration for the Dash and line 20 the BEEP duration for the Dot. Line 25 sets the gap between each dot or dash, while line 30 sets the space between each character i.e. 2 dots (this is trebled for the space between words) plus 1 dot automatically after each character. Lines 35/60 generates a string of 5 letter groups.

It is suggested that for learners 1 or 2 'space' chrs are placed after each letter for 'thinking time', and the speed set to 12 w.p.m. (changing the routine accordingly)

For those of you who are interested, here is the source code for the machine code routine.

```
        ORG 49800
START   PUSH IX ;Each character is read
        LD IX,(23627) ;into the A reg
NXT     LD A,(IX+6) ;and processing
        PUSH IX ;routine is called.
```

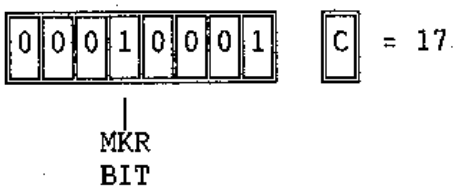
```
        BIT 7,A ;when chrs 255 is
        JR NZ,END ;received it returns
        CALL NXTCHR ;to Basic.
        POP IX
        INC IX
        BIT 5,(IY+1) ;Check for a key
        JR Z,NXT ;press and exit.
END     POP IX
        RET
NXTCHR  PUSH AF
        PUSH AF
        LD A,2 3E02 ;Open #2 and print
        CALL 5633 ;the chr. on screen.
        POP AF
        RST 16
        POP AF
        CP 32 ;If a space (CHR.32)
        JR Z,WRD ;jump to word space.
        LD HL,50000
        LD BC,0 ;Search the table for a
SEARCH  LD C,(HL) ;matching ASCII code
        INC HL ;and get the corresp.
        LD B,(HL) ;Morse Symbol from
        CP C ;adjacent address.
        JR Z,SYMB ;(see example)
        INC HL
        JR SEARCH
SYMB    SRA B ;Shift to right
        PUSH BC  ;and test each
        JR Z,LTRSP ;bit transferred
        JR C,DASH ;to carry. 0=Dot
        JR NC,DOT ;1=Dash
DASH    LD DE,160 ;If byte zero LTRSP.
        LD HL,594
        CALL BEEPER ;For dash and dot
        JR DELAY ;HL is loaded with a
DOT     LD DE,53 ;constant for a 700-
        LD HL,594 ;CPS tone. DE loading
        CALL BEEPER ;poked from Basic
DELAY   LD BC,6933 ;Duration depends
DLY     DEC BC ;on speed
        LD A,C ;A 1 dot delay after
        OR B ;each symbol.
        JR NZ,DLY
        POP BC
        JR SYMB ;Get next symbol
WRDSP   PUSH BC
        LD B,3 ;Word space=6 dots
        JR SP ;so thrice around delay
LTRSP   LD B,1 ;Ltr space=2 dots
SP      LD DE,1438 ;so once around
DLY2    DEC DE ;delay loop
        LD A,E ;DE loaded by pokes
        OR D ;from basic depending
        JR NZ;DLY2 ;on speed.
        DJNZ SP
        POP BC
        RET ;Return for next char.
```

EXAMPLE

The ASCII code for 'B' is 66 this is stored at address 49922. The required BINARY code for B is at 49923, this is arrived at by the following means:-

The Morse for B is -... this is stored in the byte in reverse order (1=Dash,0=Dot) preceded by a Marker Bit i.e.

Address 49923 =



The Bit is shifted Right and it can be seen that if a 1 is the next Bit to be shifted it results in the CARRY being set, if a 0 then the CARRY is reset. Because of the Marker bit the Byte does not reach zero until all the 0's (dots) have been processed.

NEV'S
HELP PAGE

By:- Nev Young.

Lets start off with an easy one this month. A radio ham from Sheffield would like to know if there is any way of getting programms from an IBM disc in the Spectrum using a PLUS D.

Yes there is, but it won't work.

The reason is that the IBM machines are built around the Intel ix86 chip set (8086, 80286, 80386 etc). So as you would expect the programs are written in machine code for those chips. The Spectrum, however, uses the Zilog Z80A which is about 99.5% incompatible with the Intel chip. So although you could get the program off the disc and into the Spectrum the Z80A wouldn't have a clue what to do with it.

Even getting that data is not so easy as the disc directory and track layouts are different so you have to find your way around the IBM disc using LOAD @ commands. If you realy want to use the programs I would suggest that you downgrade your computer system and get an IBM compatible.

F.Horne of Staines has a 128K Spectrum (not a sugar machine) but no manual and would like to know the connections for the RGB connector. I believe they are as follows:-

1 composite video
2 ground
3 bright
4 composite sync
5 vertical sync
6 green
7 red
8 blue

Now I notice this is the same as the 128K+2 so beware, my source may have been wrong. I don't know of any good books on the 128K machine. For some reason the publishers just didn't think the machine worth covering in the same way they did the 48K.

C.Chapman of Selkirk is having lots of fun with his new Sam. Only thing is he can't get it to print correctly. (This means he has a printer interface unlike some of us who have to struggle by without). He says he can not stop the Sam from grabbing the printer control codes going to the printer. On the PLUS D he just did POKE @6,1 but what about the Sam.

It's just as easy CLOSE #3: OPEN #3;"B". This closes the printer channel and then re-opens it as a binary channel so what you send is what the printer gets. As for the DUMP command this was not implimented in the ROM due to lack of space (this should have been stated in a correction slip in your Sam manual). However, The latest version of Sam DOS has the Dump command as an optional file that can be loaded into memory and then stays there as an extention to Basic.

Now I am getting concerned. S.Subert of Watford writes that he also gets a keyboard lock up after saving 33 files to the disc using Tasword +2 but, unlike the other three letters I've had he is using a Disciple. Now, I am using Tasword +2 to write this, and I can not get this fault to appear. I do get a crash sometimes when I type a ":" and I have to use the patch to get the keyscan to work (RANDOMIZE USR 65000 after loading Tascode). If somebody could let me know how to create this fault then I'll try and come up with a fix.

C.Martin of Baldock is having trouble with DFLIP. He can not get it to work with his DISCiPLE although his PLUS D is OK. He says that he gets the

error that it only works on 80 and 40 track drives. The problem lies somewhere in the RANDOMIZE USR 65000 this routine returns the value that is at POKE @1. This is the number of tracks. To do this, hook code 71 is used to page in the DISCiPLE ROM. But this hook code was only added on DOS version 3d. That's your problem, you state that you are still using GDOS 3b. Get an upgrade and all will be fine.

M.Bennett of Bedford South Africa is having terrible problems with Tasword +2. He has sent me a list of problems.

1. Print files do not work (page 35 of manual).

2. Suppression of the catalogue does not work correctly (page 56).

3. &number command causes a crash (page 45).

4. && crashes (page 47)

It took me some time to try this lot as I had to dig out my old microdrive to make sure the functions worked before the program was converted for the PLUS D. Of course, they did. So whoever wrote the TASCON+2 didn't include these features. If any body out there has a fix let me know and I'll pass it on to the rest of the world. (Better still send it to the editor and get paid for it).

If I manage to get some spare time then I'll try and look at it myself but don't hold your breath.

Your query about the version of Tasword +2 is that version 1.01 does not include the keyscan patch at address 65000. Apart from that they are the same.

If you want a Z80 programming manual then I can recommend 'Z80 Assembly Language Programming' written by Lance A Leventhal, and 'Z80 CPU Instruction Set' published by SGS ATES There are 'undreds of others and I daresay that what I found good may not appeal to you.

Roy Burford of Stourbridge claims not to be able to get a 128K machine to list a BASIC program to the printer connected to the RS232 port.

Now, if I assume that your printer is set for 4800 bps, then you set up the print channel with the command FORMAT "P";4800 now you can test it by LPRINT "Hello world". If that prints OK then you can print out the basic program with either the PRINT option on the main menu or by the command LLIST. If you are using a DISCiPLE or PLUS D then you must Boot-up with a system file where you have said NO to using the printer, otherwise the RS232 port is disabled.

Of course if the printer didn't print "hello world." then its no surprise that you can't print the program. If so the thing to do is try and set up your printer to the Spectrum defaults which are:-

RS232 speed = 9600 bps.
8 bits per character.
No parity.
One Stop bit.

You shouldn't need to make any changes to the RS232 output routines at 23349 as you do not want to send the data directly to the printer. If you do bypass the POUT routine then none of the programs keywords will be expanded and will instead be printed as garbage.

Well, thats all for this month. Keep those letters coming. I will answer as many queries as possible but only through the magazine. Also it is in your interest to send me as much info as possible such as program listings etc. as contrary to popular belief I do not have a vast library of programs, I only do this in my spare time!

Write ONLY to Nev Young at:-

FORMAT Help Line,
3, Mitchell Place,
Falkirk,
Stirlingshire,
Scotland, FK1 5PJ.

OTHELLO

A Game For SAM

By:- Adrian Parker.

This program was written for two reasons. First, I like the game Othello, and secondly to try out some of the features of SAM Basic.

The listing is liberally sprinkled with REMarks, so it should be easy to follow the logic of the program. I don't claim that the computer plays at a particularly high level, but it does give a reasonable game.

For those of you who are unfamiliar with Othello (also known as Reversi) the rules are quite simple. The game is played on a 8 by 8 board. At the start of the game four counters (two of each colour are already placed at the centre of the board. Each player then takes it in turn to place a counter in such a way that some of the opponents pieces are captured. To capture a piece (or more than one) you need to place your counter so that your opponents piece(s) lie in a line between your new counter and one of yours that is already on the board. These lines can be vertical, horizontal or diagonal and all directions will be tested by the computer and all of your opponents counters in the line will change to your colour. The object of the game is to have more of your coloured counters on the board, at the end of the game, than your opponent.

As well as playing against the computer you can also play against a friend, with the computer making sure you keep to the rules.

I hope readers will enjoy the program.

```
10 REM
20 REM Othello - this version by A.
   Parker commenced on 16-02-90
30 REM Othello - Layout, validity c
   hecking completed 17-02-90
40 REM Othello - IQ = 1 completed 1
   8-02-90
50 REM Othello - IQ = 2 completed 1
   8-02-90
60 REM
70 :
100 REM
110 REM Define & Initialize Variable
    s.
120 REM
130 DIM bd(8,8): REM Array of Piece
    positions on board.
140 DIM take(64,2): REM Array of Pie
    ces that can be taken on current
     move.
150 LET i=0,j=0: REM General Loop Va
    riables.
160 LET k=0: REM General Loop Variab
    le.
170 LET dx=0,dy=0: REM Direction of
    Scan in Valid Move Routine.
180 LET a$="": REM General Use.
190 LET p$="": REM Current piece def
    inition.
200 LET p1$="": REM Player 1 Name.
210 LET p2$="": REM Player 2 Name.
220 LET pc1$="": REM GRABbed drawing
     of player 1 piece.
230 LET pc2$="": REM GRABbed drawing
     of player 2 piece.
240 LET emp$="": REM GRABbed drawing
     of an empty square.
250 LET col=2: REM Current Player Nu
    mber and Colour.
260 LET taken=0: REM Number of Piece
    s that can be taken on current m
    ove.
270 LET other=1: REM Identifier of c
    urrent opponent's pieces.
280 LET max=-100: REM Best number of
     computer taken pieces so far.
290 LET iq=0: REM IQ of Current Play
    er.
300 LET iq1=0: REM IQ of Player 1.
310 LET iq2=0: REM IQ of Player 2.
320 LET xmove=1: REM X Co-ordinate o
    f current move.
330 LET ymove=1: REM Y Co-ordinate o
    f current move.
340 LET xnow=1: REM Current Test Pos
```

```
    ition X Co-ordinate.
350 LET ynow=1: REM Current Test Pos
    ition Y Co-ordinate.
360 LET xput=1: REM X Co-ordinate of
     best computer try so far.
370 LET yput=1: REM Y Co-ordinate of
     best computer try so far.
380 LET sc1=2: REM Score of Player 1
390 LET sc2=2: REM Score of Player 2
400 LET byhere=0: REM Number of near
    est neighbours.
410 LET newgame=0: REM Another Game
    Flag.
420 LET endgame=0: REM End of Game F
    lag.
430 LET invalid=0: REM Invalid Move
    Flag.
440 LET nomove=0: REM No Move Flag.
450 REM
460 REM Main Program.
470 REM
480 GOSUB 780: REM Initial Interroga
    tion.
490 GOSUB 1160: REM Setup Starter Sc
    reen.
500 REM
510 REM Main Loop.
520 REM
530 DO WHILE NOT endgame
540 PRINT AT 0,19;sc1;" ": REM Displ
    ay score of Player 1.
550 PRINT AT 10,19;sc2;" ": REM Disp
    lay score of Player 2.
560 GOSUB 1670: REM Decide whose tur
    n, adjust that window.
570 IF iq THEN GOSUB 3030: REM Make
    computer move.
580 IF NOT iq THEN GOSUB 1780: REM G
    et human move.
590 IF nomove THEN GOTO 640: REM If
    no move is possible, skip checks
600 GOSUB 1960: REM Valid Move ?
610 IF invalid THEN GOSUB 2140: REM
    Say if invalid move.
620 IF invalid THEN GOTO 570: REM Re
    -enter move if invalid.
630 GOSUB 2410: REM Make Move.
640 GOSUB 2640: REM End of game ?
650 IF endgame THEN GOSUB 2690: REM
    Game Over.
660 IF endgame THEN GOSUB 2840: REM
    Another Game ?
670 IF newgame THEN RUN: REM Start A
    gain.
680 LOOP
690 REM
700 REM Return to System.
710 REM
720 CLS #
730 MODE 3
740 STOP
750 REM
760 REM Initial Interrogation.
770 REM
780 CLS #: CSIZE 6,8
790 MODE 3
800 PRINT "Do I move first? (y/n) ";
810 GET a$
820 IF a$="N" OR a$="n" THEN PRINT a
    $: LET iq1=0: GOTO 850
830 IF a$="Y" OR a$="y" THEN PRINT a
    $: GOTO 880
840 BEEP 1,.1: GOTO 810
850 PRINT: PRINT "What is the name o
    f player 1 ? ";
860 INPUT #2; LINE p1$
870 GOTO 940
880 PRINT: PRINT "What IQ level shou
    ld I play (1-2) ";
890 GET a$
900 IF a$="1" THEN LET iq1=1
910 IF a$="2" THEN LET iq1=2
920 IF iq1=0 THEN BEEP .1,1: GOTO 89
    0: ELSE PRINT a$
930 LET p1$="SAM (IQ="+a$+")"
940 PRINT: PRINT "Do I move second ?
     (y/n) ";
950 GET a$
960 IF a$="N" OR a$="n" THEN PRINT a
    $: LET iq2=0: GOTO 990
970 IF a$="Y" OR a$="y" THEN PRINT a
    $: GOTO 1020
980 BEEP 1,.1: GOTO 950
990 PRINT: PRINT "What is the name o
    f player 2 ? ";
1000 INPUT #2; LINE p2$
1010 GOTO 1100
1020 PRINT: PRINT "What IQ level shou
     ld I play (1-2) ";
1030 GET a$
1040 IF a$="1" THEN LET iq2=1
1050 IF a$="2" THEN LET iq2=2
1060 IF iq2=0 THEN BEEP 1,.2: GOTO 10
     20: ELSE PRINT a$
1070 LET p2$="SAM (IQ="+a$+")"
1080 IF LEN p1$ > 10 THEN LET p1$=p1$
     ( TO 10)
1090 IF LEN p2$ > 10 THEN LET p2$=p2$
     ( TO 10)
1100 LET p1$=p1$+" "
1110 LET p2$=p2$+" "
1120 RETURN
1130 REM
1140 REM Setup Initial Screen.
1150 REM
1160 MODE 4
```

```
1170 PALETTE 1,89
1180 PALETTE 2,68
1190 PEN 15
1200 PRINT AT 0,6;"Othello."
1210 PLOT 38,164
1220 DRAW 80,0
1230 PRINT AT 2,2;"1 2 3 4 5 6 7 8"
1240 FOR j = 1 TO 8
1250 PRINT AT (j-1)*2+4,0;CHR$ (64+j)
1260 NEXT j
1270 FOR j = 0 TO 144 STEP 18
1280 PLOT 12,j
1290 DRAW 127,0
1300 NEXT j
1310 FOR j=12 TO 146 STEP 16
1320 PLOT j,144
1330 DRAW 0,-144
1340 NEXT j
1350 GRAB emp$,79,69,13,13
1360 CIRCLE 68,63,6
1370 FILL PEN 1;68,63
1380 CIRCLE 84,63,6
1390 FILL PEN 2;84,63
1400 CIRCLE 68,81,6
1410 FILL PEN 2;68,81
1420 CIRCLE 84,81,6
1430 FILL PEN 1;84,81
1440 GRAB pc1$,79,69,13,13
1450 GRAB pc2$,62,69,13,13
1460 PEN 2: PRINT AT 0,22;p1$( TO 10)
1470 PEN 1: PRINT AT 10,22;p2$( TO 10
     )
1480 PEN 15
1490 PLOT 175,164
1500 DRAW 10+(LEN (p1$)-11)*7,0
1510 PLOT 175,74
1520 DRAW 10+(LEN (p2$)-11)*7,0
1530 LET bd(4,4)=1,bd(5,5)=1
1540 LET bd(4,5)=2,bd(5,4)=2
1550 PRINT AT 2,23;"Othello"
1560 PRINT AT 3,23;"  by"
1570 PRINT AT 4,23;"A.Parker"
1580 PRINT AT 5,23;"  1990"
1590 PRINT AT 12,23;"Othello"
1600 PRINT AT 13,23;"  by"
1610 PRINT AT 14,23;"A.Parker"
1620 PRINT AT 15,23;"  1990"
1630 RETURN
1640 REM
1650 REM Decide whose turn and Adjust
      that window.
1660 REM
1670 LET other=col
1680 IF col=2 THEN LET col=1: ELSE LE
     T col=2
1690 IF col=1 THEN LET iq=iq1: ELSE L
     ET iq=iq2
1700 FOR j=1 TO 17
1710 SCROLL 2,1,170,155-(90*(col=2)),
     87,60
1720 FOR k=1 TO 20: NEXT k
1730 NEXT j
1740 RETURN
1750 REM
1760 REM Human Move Input.
1770 REM
1780 LET xmove=0,ymove=0
1790 PRINT AT 5+(11*(col=2)),23;"Your
      Move"
1800 PRINT AT 6+(11*(col=2)),23;">";
1810 GET a$
1820 IF a$="0" THEN PRINT "No Move!":
      LET nomove=nomove+1: RETURN
1830 LET nomove=0
1840 IF (CODE a$) >96 THEN LET a$ = C
     HR$ ((CODE a$)-32)
1850 LET ymove = (CODE a$)-64
1860 IF ymove>0 AND ymove<9 THEN PRIN
     T a$;: GOTO 1880
1870 BEEP 1,.2: GOTO 1810
1880 GET a$
1890 LET xmove = (CODE a$)-48
1900 IF xmove>0 AND xmove<9 THEN PRIN
     T a$;: GOTO 1920
1910 BEEP 1,.2: GOTO 1880
1920 RETURN
1930 REM
1940 REM Valid Move ?
1950 REM
1960 LET invalid=0,taken=0,byhere=0
1970 IF bd(xmove,ymove) <>0 THEN LET
     invalid=1: RETURN
1980 FOR k=1 TO 8
1990 LET xnow=xmove,ynow=ymove
2000 IF k=1 THEN LET dx=1,dy=0: GOSUB
      2270
2010 IF k=2 THEN LET dx=1,dy=1: GOSUB
      2270
2020 IF k=3 THEN LET dx=0,dy=1: GOSUB
      2270
2030 IF k=4 THEN LET dx=-1,dy=1: GOSU
     B 2270
2040 IF k=5 THEN LET dx=-1,dy=0: GOSU
     B 2270
2050 IF k=6 THEN LET dx=-1,dy=-1: GOS
     UB 2270
2060 IF k=7 THEN LET dx=0,dy=-1: GOSU
     B 2270
2070 IF k=8 THEN LET dx=1,dy=-1: GOSU
     B 2270
2080 NEXT k
2090 IF taken=0 THEN LET invalid=1
2100 RETURN
2110 REM
2120 REM Invalid Move.
2130 REM
```

```
2140 FOR j=1 TO 17
2150 SCROLL 2,1,170,155-(90*(col=2)),
     87,60
2160 NEXT j
2170 PRINT AT 5+(11*(col=2)),23;" Ill
     egal"
2180 PRINT AT 6+(11*(col=2)),23;" Mo
     ve!"
2190 FOR j=1 TO 18
2200 SCROLL 2,1,170,155-(90*(col=2)),
     87,60
2210 NEXT j
2220 FOR j=1 TO 5: BOOM: NEXT j
2230 RETURN
2240 REM
2250 REM Search one direction from he
     re for pieces to take.
2260 REM
2270 LET count=0
2280 LET xnow = xnow + dx
2290 LET ynow = ynow + dy
2300 IF xnow>8 OR xnow<1 OR ynow>8 OR
      ynow<1 THEN LET taken=taken-cou
     nt: RETURN
2310 IF bd(xnow,ynow)=col AND count=0
      THEN LET byhere=byhere+1: RETUR
     N
2320 IF bd(xnow,ynow)=0 THEN LET take
     n=taken-count: RETURN
2330 IF bd(xnow,ynow)=col THEN RETURN
2340 LET count=count+1,taken=taken+1
2350 LET take(taken,1)=xnow
2360 LET take(taken,2)=ynow
2370 GOTO 2280
2380 REM
2390 REM Make Move.
2400 REM
2410 IF col=1 THEN LET p$=pc1$: ELSE
     LET p$=pc2$
2420 FOR j=1 TO 3
2430 PUT 15+(xmove-1)*16,33+(7-ymove)
     *18,p$
2440 ZOOM
2450 FOR k=1 TO 100: NEXT k
2460 PUT 15+(xmove-1)*16,33+(7-ymove)
     *18,emp$
2470 FOR k=1 TO 100: NEXT k
2480 NEXT j
2490 LET bd(xmove,ymove)=col
2500 PUT 15+(xmove-1)*16,33+(7-ymove)
     *18,p$
2510 FOR k=1 TO taken
2520 LET xnow=take(k,1),ynow=take(k,2
     )
2530 LET bd(xnow,ynow)=col
2540 PUT 15+(take(k,1)-1)*16,33+(7-ta
     ke(k,2))*18,p$
2550 ZAP
2560 FOR j=1 TO 50: NEXT j
2570 NEXT k
2580 IF col=1 THEN LET sc1=sc1+1+take
     n,sc2=sc2-taken
2590 IF col=2 THEN LET sc2=sc2+1+take
     n,sc1=sc1-taken
2600 RETURN
2610 REM
2620 REM End of Game Test.
2630 REM
2640 IF sc1+sc2=64 OR nomove=2 OR sc1
     =0 OR sc2=0 THEN LET endgame=1
2650 RETURN
2660 REM
2670 REM End of Game.
2680 REM
2690 PAUSE 50
2700 CLS
2710 MODE 3
2720 LET p1$=p1$( TO 10),p2$=p2$( TO
     10)
2730 PRINT AT 5,10;"Congratulations "
     ;
2740 IF sc1>sc2 THEN PRINT p1$;: ELSE
      PRINT p2$;
2750 PRINT "!"
2760 PRINT AT 7,10;"You beat ";
2770 IF sc1>sc2 THEN PRINT p2$;: ELSE
      PRINT p1$;
2780 PRINT " by ";
2790 IF sc1>sc2 THEN PRINT sc1;" poin
     ts to ";sc2: RETURN
2800 PRINT sc2;" points to ";sc1: RET
     URN
2810 REM
2820 REM Another Game ?
2830 REM
2840 PRINT AT 18,28;"Another Game ( Y
     /N )"
2850 GET a$
2860 IF a$="y" OR a$="Y" THEN LET new
     game=1: RETURN
2870 IF a$="n" OR a$="N" THEN LET new
     game=0: RETURN
2880 BOOM: GOTO 2850
2890 :
2940 REM
2950 REM Computer Calculated Moves.
2960 REM
2970 REM This is the basis for all of
      the computer's moves and is the
2980 REM the sole determining factor
     for the computer IQ=1. It merely
2990 REM places it's piece so as to t
     ake the maximum number of pieces
3000 REM of it's opponent.
3010 REM A.Parker 18-02-90
3020 REM
```

```
3030 LET max=-100
3040 PRINT AT 5+(11*(col=2)),23;"My M
     ove"
3050 PRINT AT 6+(11*(col=2)),23;">";
3060 FOR xmove=1 TO 8
3070 FOR ymove=1 TO 8
3080 LET invalid=0
3090 PRINT AT 6+(11*(col=2)),24;CHR$
     (xmove+64);",";ymove;"  "
3100 GOSUB 1960: REM Validity test.
3110 IF invalid THEN GOTO 3150
3120 IF iq>1 THEN GOSUB 3290
3130 LET taken=taken+byhere
3140 IF taken>max THEN LET max=taken,
     xput=xmove,yput=ymove
3150 NEXT ymove
3160 NEXT xmove
3170 IF max=-100 THEN PRINT AT 6+(11*
     (col=2)),24;"No move!": LET nomo
     ve=nomove+1
3180 LET xmove=xput,ymove=yput
3190 IF NOT nomove THEN PRINT AT 6+(1
     1*(col=2)),24;CHR$ (ymove+64);ST
     R$ (xmove) ;"  "
3200 RETURN
3210 REM
3220 REM The computer IQ level of 2 i
     s identical to that of IQ = 1, w
     ith the
3230 REM addition of the following su
     broutine.  It weights the number
      of pieces
3240 REM taken with the position on t
     he board.  ie those pieces on t
     he corners
3250 REM or edges are given priority,
     whereas those moves on penultima
     te columns
3260 REM or rows are to be avoided.
3270 REM A.Parker 18-02-90
3280 REM
3290 IF xmove=2 OR xmove=7 THEN LET t
     aken=taken-5
3300 IF ymove=2 OR ymove=7 THEN LET t
     aken=taken-5
3310 IF xmove=1 OR xmove=8 THEN LET t
     aken=taken+5
3320 IF ymove=1 OR ymove=8 THEN LET t
     aken=taken+5
3330 RETURN
```

K.A.

Find Out What's On Your DISCiPLE & PLUS D Discs

By:- Nev Young.

KA is a disc utility designed to produce a complete sector usage map of a DISCiPLE or PLUS D disc. It is written completely in BASIC and should be fairly easy to follow. Briefly it works as follows:-

Lines 10 - 60 create and initialise the data arrays.

Lines 70 - 130 ask which disc and whether you want to use a printer.

Lines 150 - 240 is the main loop and reads each directory sector in turn.

Lines 250 - 310 store the file name, length and start sector in array f$.

Lines 320 - 490 Read every sector of the file just found by lines 250-310 and compares its length with the length held in the file header. Also every sector used is put into the array m$. Checks are also made to see if the file collides with any other file.

Lines 500 - end give a printout of the disc sector map with a summary of how fragmented the disc is and of any missing or duplicated sectors.

The {GS 8} on lines 60, 400, 600 and 610 are produced by using the Graphics Mode and pressing Shift and 8. Line 60 requires 20 of them.

Perhaps one day I'll do a program to unscramble any discs that have become corrupted. (Then again maybe not).

By the way if you are wondering why the program is called KA its because where I work the disc map utility is called KA. It gets it's name from: disK Analyser. (Bob doesn't like that he says it should be CA!) But KA is the name I use so KA it is.

KA THE PROGRAM

```
 10 REM KA By Nev Young.
 20 CLEAR 59999
 30 LET TRACKS=80
 40 DIM F$(80,15)
 50 DIM M$(1600)
 60 FOR N=1 TO 1600 STEP 20: LET M$(N
    TO N+19)="{20 GS 8}": NEXT N
 70 INPUT "Disc - 1/2 ";DISC
 80 INPUT "Hard copy y/n ";P$
 90 LET ASS=0: LET MIS=0
100 LET P$=CHR$ CODE P$: LET CH=2
110 IF P$="y" OR P$="Y" THEN LET CH=3
    : GOTO 130
120 IF P$<>"n" AND P$<>"N" THEN GOTO
    80
130 IF DISC<1 OR DISC>2 THEN GOTO 70
140 LET F=1: LET TCNT= 0
150 FOR T=0 TO 3: FOR S=1 TO 10
160 LOAD @DISC,T,S,60000
170 FOR H=0 TO 256 STEP 256
180 IF NOT PEEK (60000+H) THEN GOTO 2
    10
190 GOSUB 250
200 LET F=F+1
210 NEXT H
220 NEXT S
230 NEXT T
240 GOTO 500
250 FOR N=1 TO 14
260 LET F$(F,N)=CHR$ PEEK (60000+N+H)
270 NEXT N: LET F$(F,15)=CHR$ 0
280 PRINT #CH; "File #";F;TAB 8;": ";
    F$(F)( TO 10);" ";
290 INPUT ,
300 GOSUB 320
310 RETURN
320 REM Load sector map
330 LET CNT=0
340 LET SIZE=CODE F$(F,12)+256*CODE F
    $(F,11)
350 LET TR=CODE F$(F,13): LET SE=CODE
     F$(F,14)
360 IF TR=0 OR SE=0 THEN GOTO 460
370 LOAD @DISC,TR,SE,61000
380 LET O=SE+(TR-128 AND (TR>80))*10
390 IF O>1600 THEN PRINT #CH;TAB 2;"e
    xceeds max disc address": RETURN
400 IF M$(O)<>"{GS 8}" THEN PRINT #CH
```

```
    ;TAB 2;"collides @ ";TR;"/";SE;"
    with ";F$(CODE M$(O), TO 10): LET
     ASS=ASS+SIZE: RETURN
410 LET M$(O)=CHR$ F
420 LET TR=PEEK 61510: LET SE=PEEK 61
    511
430 LET CNT=CNT+1
440 LET TCNT=TCNT+1
450 GOTO 360
460 IF CNT=SIZE THEN PRINT #CH;"OK":
    RETURN
470 PRINT #CH;TAB 2;"size was ";CNT;"
     should be ";SIZE
480 LET MIS=MIS+SIZE-CNT
490 RETURN
500 REM printout
510 CLS : IF CH=3 THEN LPRINT '''''
520 PRINT #CH;"Start";TAB 16;"In use
    by"'"tr/se";TAB 8;"length";TAB 16
    ;"  file";TAB 28;"area"''
530 LET FREE=0: LET FRA=0
540 PRINT #CH;"0/1";TAB 8;40;TAB 16;"
    Directory";TAB 28;1
550 LET N=41
560 LET SE=N: GOSUB 700: PRINT #CH;TR
    ;"/";SE;TAB 8;
570 LET L=0: LET X$=M$(N)
580 IF M$(N)=X$ THEN LET N=N+1: LET L
    =L+1: IF N<1600 THEN GOTO 580
590 INPUT ,: PRINT #CH;L;TAB 16;
600 IF X$="{GS 8}" THEN PRINT #CH; "F
    ree space";TAB 28;: LET FREE=FREE
    +L: LET FRA=FRA+1: PRINT #CH; FRA
610 IF X$<>"{GA 8}" THEN PRINT #CH;F$
    (CODE X$)( TO 10);TAB 28;: LET F$
    (CODE X$,15)=CHR$ (1+CODE F$(CODE
     X$,15)): PRINT #CH;CODE F$(CODE
    X$,15)
620 IF N<1600 THEN GOTO 560
630 PRINT #CH;''''"Total free sectors
          = ";FREE
640 PRINT #CH;"Total free areas
       = ";FRA
650 PRINT #CH;"Total missing sectors
       = ";MIS
660 PRINT #CH;"Total duplicated secto
    rs = ";ASS
670 PRINT #CH; '''''''''': STOP
680 SAVE d1"KA" LINE 10
690 RUN
700 LET TR=INT (SE/10)
710 LET SE=SE-10*TR
720 IF TR>=TRACKS THEN LET TR=TR+128-
    TRACKS
730 RETURN
```

* - * - *

BEYOND SIMPLE UDGs

By:- Clyde Bish.

A while ago there was a plea from a subscriber for some "good old fashioned listings like magazines used to carry!" Well, its your User Group so "Ask And Ye Shall Have". But please don't look on it as just a game. You'll find you'll learn many programming techniques on the way.

The listing is for the first part of the 3D maze game I promised you last issue. It uses the picture elements we have been working on for a couple of months (but in a slightly different form so don't worry if you're new to FORMAT). Unlike the original 3D Maze of early ZX days this one doesn't have a raging dinosaur charging up on you, but later it will have a rather unpleasant Green Goblin. He'll be following you around the maze, sending indecisive players back to the start. He may (reluctantly) award you with a pot of gold IF you ever reach the middle. But more of that in the next issue. Lets concentrate this time on the basic maze.

Let's begin by picking out a few points of interest before we look at the listing for Program 1 in detail.

1. There's one main addition to the basic maze elements idea that we've looked at before. You get two views of each element. The first is a distant view, which changes to a closer one as you move forwards. Some examples of what you'll see on screen are given at the end.

2. Don't search this issue of FORMAT for a plan of the maze. You won't find one. No, Bob has run out of space or, horror of horrors, missed it out! It doesn't exist until you begin to play. You see it all depends on the values generated by RND in line 210 coupled with the actual moves a player makes. This means that the person typing it in is just as in the dark as someone coming to it fresh, and will find it just as difficult to play. A rather unusual state of affairs I think you will agree. Anyway, I think a map makes it all too easy! You will be pleased to know, however, that once an element is generated it does stay in the same place throughout the game. This is ensured by strings X$ and Y$.

3. The program does not use machine code. I feel that the speed of the Spectrum's PLOT and DRAW commands are fast enough. Also be avoiding machine code the structure of the program is more easily seen by those wishing to learn or extend their experience of programming techniques (and those who just like to "dissect" programs!)

4. The elements which are used to construct the views of the maze are placed at the front of the listing as this increases their speed of access.

O.K. Enough of this introductory chat. Let's start on the serious work of entering the lines of Program 1. When all is entered save with:-

```
SAVE "MAZE" LINE 6000
```

Now its safe on tape let's give the typing fingers a rest for a moment as we see how its going to operate. The easiest way to do this is to look at what each group of lines does:-

190-280:- This is the logic controlling your path through the maze (which itself is set in line 6020 as you'll see later). The basis is this:-

a) 210:- if the next-but-one element is a dead end (element 7) then the

program generates a random number. If this is less than .2 then the routine skips the dead end, otherwise the sequence in X$ is followed, the dead-end is reached and the player is sent back to the beginning.

b) 270:- When the player reaches this decision point again if he/she remembers and chooses an alternative pathway (the original choice being stored in Y$) then the dead-end is avoided.

c) 260:- If during a further attempt a player takes a different turning to one made the previous time (excepting (b) above) then a dead-end is generated and the player is sent back to the beginning.

This is what I meant in the introduction by the layout of the maze being designed as play proceeds.

1000-1710:- The elements of the maze are displayed on screen as if you are looking down the corridoor. There are three passage elements - 1000 = left turn, 1100 = right turn and 1200 = straight on. Also there are four junction elements - 1300 = forward or right, 1400 = forward or left, 1500 = left or right and 1600 = forward, left or right. There is also the dead end at 1700. Each element, except those displayed by lines 1200 and 1700 are two part giving first a distant view followed by a closer view with the player in a position to turn if necessary.

2000-2020:- If the player tries to walk into the passage wall this rouine prevents him/her from doing so.

3000-3060:- This section invites a choice to be made (allowing only keys 5, 7 or 8). You can by the way stop play at any time by pressing 0. (If you do you can start again with RUN.

6000-6015:- On loading this is where the program starts. Instructions are given and the difficulty level (the number of elements in the maze) is also selected here.

6020-6040:- Once the difficulty level is selected the maze is set up as a string of numbers (representing the elements to be called) and held in X$. They are alternately 0 - 2, followed by 3 - 7, there not being two 7's following each other (or a junction would only have two dead-end exits!) As the maze is computer generated no player, not even you, can know the "true path".

8000:- The winning sequence. We'll improve on this next time.

Now I'll let you play the game! Normally you would LOAD from tape and the program would autostart with the instructions but assuming you still have it on board start it up with RUN. Choose your difficulty level to determine the one true path to the centre. The cursor keys control your movements , but remember not to turn too early of you'll walk into the wall.

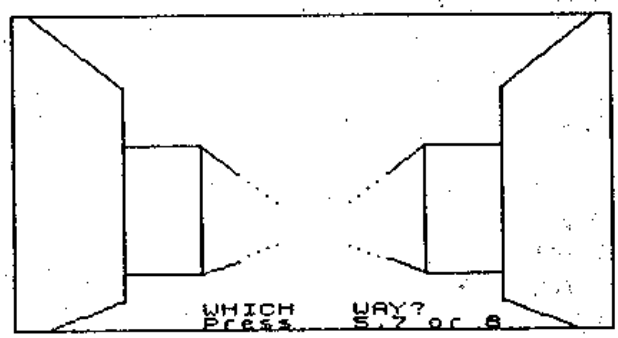
Next time we'll add the Green Goblin - and You! Until then, can you find the "true path"?

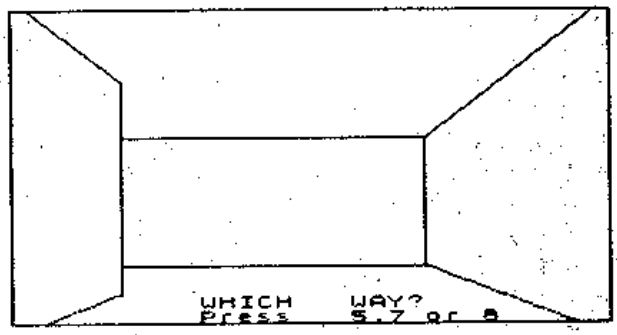
Program 1.

```
  1 RUN 6000
190 LET Y$="
       ": REM 28 spaces
200 FOR I=1 TO Y: GOSUB VAL X$(I)*100
    +1000
205 LET I=I+1: GOSUB VAL X$(I)*100+10
    00: IF X$(I)="7" THEN GOTO 200
210 IF Y$(I)=" " AND X$(I+2)="7" THEN
     LET Y$(I)=B$: LET R=RND: LET X$(
    I+2)=("8" AND R<.3)+("7" AND R>=.
    3): LET I=I+(2 AND R<.3): GOTO 28
    0
220 IF Y$(I)=" " AND X$(I+2)<>"7" THE
    N LET Y$(I)=B$: GOTO 280
230 IF Y$(I)<>" " AND X$(I+2)="8" AND
     Y$(I)=B$ THEN LET I=I+2: GOTO 28
    0
260 IF Y$(I)<>" " AND Y$(I)<>B$ AND X
    $(I+2)<>"7" THEN GOSUB INT (RND*3
    )*100+1000: GOSUB 1700: GOTO 200
270 IF Y$(I)<>" " AND Y$(I)<>B$ AND X
    $(I+2)="7" THEN LET Y$(I)=B$: LET
     I=I+2: LET X$(I)="8"
```

```
 280 NEXT I: GOTO 8000
1001 CLS : PLOT 47,32: DRAW 128,0: PLO
     T 47,103: DRAW 128,0: PLOT 15,0:
     DRAW 32,16: DRAW 0,119: DRAW -40,
     40: PLOT 240,0: DRAW -64,32: DRAW
      0,71: DRAW 72,72
1010 GOSUB 3000
1015 IF B$<>"7" THEN GOSUB 2000: GOTO
     1000
1051 CLS : PLOT 16,32: DRAW 159,0: PLO
     T 16,103: DRAW 159,0: PLOT 15,0:
     DRAW 0,167: DRAW -8,8: PLOT 240,0
     : DRAW -64,32: DRAW 0,71: DRAW 72
     ,72
1060 GOSUB 3000
1065 IF B$<>"5" THEN GOSUB 2000: GOTO
     1050
1070 RETURN
1101 CLS : PLOT 79,32: DRAW 128,0: PLO
     T 79,103: DRAW 128,0: PLOT 15,0:
     DRAW 64,32: DRAW 0,71: DRAW -72,7
     2: PLOT 240,0: DRAW -32,16: DRAW
     0,119: DRAW 40,40
1110 GOSUB 3000
1115 IF B$<>"7" THEN GOSUB 2000: GOTO
     1100
1151 CLS : PLOT 80,32: DRAW 159,0: PLO
     T 80,103: DRAW 159,0: PLOT 15,0:
     DRAW 64,32: DRAW 0,71: DRAW -72,7
     2: PLOT 240,0: DRAW 0,167: DRAW 8
     ,8
1160 GOSUB 3000
1165 IF B$<>"8" THEN GOSUB 2000: GOTO
     1150
1170 RETURN
1201 CLS : PLOT 15,0: DRAW 64,32: FOR
     N=2 TO 8 STEP 2: PLOT 2*N+80,N+32
     : NEXT N: PLOT 239,0: DRAW -64,32
     : FOR N=2 TO 8 STEP 2: PLOT -2*N+
     175,N+32: NEXT N: PLOT 8,175: DRA
     W 72,-72: FOR N=4 TO 16 STEP 4: P
     LOT 80+N,103-N: NEXT N: PLOT 248,
     175: DRAW -72,-72: FOR N=4 TO 16
     STEP 4: PLOT 175-N,103-N: NEXT N
1205 PRINT AT 12,12;"CONTINUE";AT 14,1
     2;"FORWARDS"
1210 GOSUB 3000
1215 IF B$<>"7" THEN GOSUB 2000: GOTO
     1201
1230 RETURN
1301 CLS : PLOT 240,0: DRAW -32,16: DR
     AW 0,119: DRAW 40,40: PLOT 207,32
     : DRAW -32,0: DRAW 0,71: DRAW 32,
     0: PLOT 175,32: DRAW -16,8: FOR N
     =2 TO 8 STEP 2: PLOT -2*N+159,N+3
     9: NEXT N: PLOT 175,103: DRAW -16
     ,-16: FOR N=4 TO 16 STEP 4: PLOT
     159-N,87-N: NEXT N: IF X$(I)="6"
     THEN GOTO 1402
1310 PLOT 15,0: DRAW 80,40: FOR N=2 TO
      8 STEP 2: PLOT 2*N+96,N+40: NEXT
      N: PLOT 8,175: DRAW 88,-88: FOR
     N=4 TO 16 STEP 4: PLOT N+96,87-N:
      NEXT N
1320 GOSUB 3000
1325 IF B$<>"7" THEN GOSUB 2000: GOTO
     1300
1350 CLS : PLOT 240,0: DRAW 0,167: DRA
     W 8,8: PLOT 239,32: DRAW -64,0: D
     RAW 0,71: DRAW 64,0: PLOT 175,32:
      DRAW -16,8: FOR N=2 TO 6 STEP 2:
      PLOT -2*N+159,N+40: NEXT N: PLOT
      175,103: DRAW -16,-16: FOR N=4 T
     O 16 STEP 4: PLOT 159-N,87-N: NEX
     T N: IF X$(I)="6" THEN  GOTO 1451
1360 PLOT 15,0: DRAW 80,40: FOR N=2 TO
      8 STEP 2: PLOT 2*N+96,N+40: NEXT
      N: PLOT 8,175: DRAW 88,-88: FOR
     N=4 TO 16 STEP 4: PLOT 96+N,87-N:
      NEXT N
1370 GOSUB 3000
1372 IF B$<>"8" AND B$<>"7" THEN GOSUB
      2000: GOTO 1350
1380 RETURN
1401 CLS
1402 PLOT 15,0: DRAW 32,16: DRAW 0,119
     : DRAW -40,40: PLOT 48,32: DRAW 3
     2,0: DRAW 0,71: DRAW -32,0: PLOT
     80,32: DRAW 16,6: FOR N=2 TO 8 ST
     EP 2: PLOT 2*N+96,N+40: NEXT N: P
     LOT 80,103: DRAW 16,-16: FOR N=4
     TO 16 STEP 4: PLOT N+96,87-N: NEX
     T N: IF X$(I)="6" THEN  GOTO 1610
1410 PLOT 240,0: DRAW -80,40: FOR N=2
     TO 8 STEP 2: PLOT -2*N+159,N+40:
     NEXT N: PLOT 248,175: DRAW -88,-8
     8: FOR N=4 TO 16 STEP 4: PLOT 159
     -N,87-N: NEXT N
1420 GOSUB 3000
1425 IF B$<>"7" THEN GOSUB 2000: GOTO
     1400
1450 CLS
1451 PLOT 15,0: DRAW 0,167: DRAW -8,8:
      PLOT 16,32: DRAW 64,0: DRAW 0,71
     : DRAW -64,0: PLOT 80,32: DRAW 16
     ,8: FOR N=2 TO 8 STEP 2: PLOT 2*N
     +96,N+40: NEXT N: PLOT 80,103: DR
     AW 16,-16: FOR N=4 TO 16 STEP 4:
     PLOT 96+N,87-N: NEXT N: IF X$(I)=
     "6" THEN GOTO 1660
1460 PLOT 239,0: DRAW -80,40: FOR N=2
     TO 8 STEP 2: PLOT -2*N+159,N+40:
     NEXT N: PLOT 247,175: DRAW -88,-8
     8: FOR N=4 TO 16 STEP 4: PLOT 159
     -N,87-N: NEXT N
1470 GOSUB 3000
```

```
1472 IF B$<>"5" AND B$<>"7" THEN GOSUB
     2000: GOTO 1450
1480 RETURN
1501 CLS : PLOT 48,32: DRAW 160,0: PLO
     T 48,103: DRAW 160,0: PLOT 15,0:
     DRAW 32,16: DRAW 0,119: DRAW -40,
     40: PLOT 240,0: DRAW -32,16: DRAW
      0,119: DRAW 40,40
1510 GOSUB 3000
1515 IF B$<>"7" THEN GOSUB 2000: GOTO
     1500
1551 CLS : PLOT 16,32: DRAW 224,0: PLO
     T 16,135: DRAW 224,0: PLOT 15,0:
     DRAW 0,167: DRAW -8,8: PLOT 240,0
     : DRAW 0,167: DRAW 8,8
1560 GOSUB 3000
1565 IF B$<>"8" AND B$<>"5" THEN GOSUB
      2000: GOTO 1550
1570 RETURN
1601 GOTO 1300
1610 GOSUB 3000
1615 IF B$<>"7" THEN GOSUB 2000: GOTO
     1600
1650 GOTO 1350
1660 GOSUB 3000
1680 RETURN
1701 CLS : PLOT 64,24: DRAW 127,0: PLO
     T 64,119: DRAW 127,0: PLOT 15,0:
     DRAW 48,24: DRAW 0,95: DRAW -56,5
     6: PLOT 240,0: DRAW -48,24: DRAW
     0,95: DRAW 56,56: IF I>Y THEN RET
     URN
1710 PRINT AT 10,13;"DEAD END";AT 14,1
     1;"Back to the";AT 17,14;"START":
      PAUSE 160: CLS : RETURN
2001 CLS : PLOT 0,32: DRAW 255,0: PLOT
      0,135: DRAW 255,0: PRINT AT 10,1
     3;"PASSAGE";AT 12,14;"WALL"
2008 IF B$="7" THEN GOTO 2020
2010 PRINT AT 20,11;"TURN";AT 20,18;"B
     ACK";AT 21,13;"Press ";("8" AND B
     $="5")+("5" AND B$="8"): PAUSE 0:
      IF INKEY$<>"5" AND B$="8" OR INK
     EY$<>"8" AND B$="5" THEN CLS : GO
     TO 2001
2015 PRINT AT 20,11;"     ";AT 20,18;"
        ";AT 21,13;"       ": PAUSE 30
     : CLS : RETURN
2020 PRINT AT 21,13;"TRY AGAIN": PAUSE
      30: CLS : RETURN
3000 PRINT AT 20,10;"WHICH";AT 20,18;"
     WAY?";AT 21,10;"Press";AT 21,18;"
     5,7 or 8"
3010 PAUSE 0
3050 LET B$=INKEY$: IF B$="0" THEN STO
     P
3055 IF B$<>"7" AND B$<>"8" AND B$<>"5
     " THEN GOTO 3000
3060 PRINT AT 20,10;"     ";AT 20,18;"
        ";AT 21,10;"     ";AT 21,18;"
           ": PAUSE 20: RETURN
6000 BORDER 4: PRINT AT 1,5; INK 4;"%%
     %%%%%%%%%%%%%%%%%%%%%%";AT 2,5;
     ;"%                          %";AT
     3,5;"%%%%%%%%%%%%%%%%%%%%%%%%%%";
     AT 2,6; INK 0; PAPER 6;"LEPRECHAU
     N'S GOLD";AT 5,0; PAPER 7;"Can yo
     u make your way to the     centre
      and find the GOLD?";AT 8,5;"Pres
     s 5 to turn left";AT 9,11;"7 to m
     ove forwards";AT 10,11;"8 to turn
      right";AT 11,10;"(0 to exit";AT
     12,0;"Don't turn too early - you'
     ll   walk into the wall!"
6015 PRINT  BRIGHT 1;"     PRESS ANY K
     EY TO BEGIN     ": PAUSE 0: CLS :
      INPUT "Enter difficulty level (1
      to 10) ";Z: PRINT "PREPARING MAZ
     E": LET Y=Z*2+8
6020 LET X$="
          ": FOR I=1 TO Y STEP 2: LET
     X$(I)=STR$ INT (RND*3): LET X$(I+
     1)=STR$ INT (RND*4+3): IF I<3 THE
     N GOTO 6040
6030 IF X$(I-1)<>"7" AND RND<.4 THEN L
     ET X$(I+1)="7"
6040 NEXT I
6050 GOTO 190
8000 GOSUB 1700: PRINT AT 10,8; BRIGHT
      1;"You've made it!": PAUSE 200
8025 PRINT AT 1,5;"PRESS 1 TO PLAY AGA
     IN": PAUSE 0: IF INKEY$="1" THEN
     CLS : RUN 6015
8999 STOP
```





FORMAT READERS SERVICE. For those important back-issues you need to complete your set of FORMATs and conversion software you can't get elsewhere.

FORMAT BACK-ISSUES

For readers who have missed past issues of FORMAT (or perhaps worn theirs out through constant use).

All issues from Volume 1 Number 1 (August 1987) right up to the latest issue are available. To order Just quote the Volume number and the Issue number.

Price £1.00 per copy

TAPE SOFTWARE

All software is supplied on tape for use with any format/size of disc. To use conversion software you must have the original tape copy of the software to be converted.

ARTCON 48

Converts the 48k OCP ART STUDIO to full DISCiPLE / PLUS D operation, including printer.

Order Code FST-02 **Price** £3.95

TASCON 128

Converts TASWORD 128 for use with DISCiPLE or PLUS D. All features retained, includes amendments for TW2 file converter.

Order Code FST-05 **Price** £3.95

TASCON +2

As for TASCON 128 but converts TASWORD +2 to disc.

Order Code FST-06 **Price** £3.95

WRITER 48

Converts the WRITER 48K for use with DISCiPLE or PLUS D. All features work including Help files.

Order Code FST-07 Price £3.95

FORMAT READERS SERVICES

G+DOS 2a

Latest version of the PLUS D system file. Works with ROM version 1a.

Order Code FST-90 **Price** £3.95

GDOS 3d

Up to date system file for DISCiPLEs with ROM version 3. Includes new set-up program, far easier to use.

Order Code FST-91 **Price** £3.95

ODD BITS

KEYBOARD STICKERS

Spectrum (48k) keyboard stickers to give all the single key keywords. Colour printed on clear plastic.

Order Code ODD-01 **Price** £1.50

POSTAGE/PACKING and ORDERING

Add the following charges, for postage and packing, to your total order.

	UK	OVERSEAS	AIR
Under £10	£1.00	£1.50	£2.50
£10 to £25	£1.50	£2.25	£3.50
Over £25	£FREE	£3.00	£4.50

Please **WRITE YOUR ORDER ON A PIECE OF PLAIN PAPER**, and mark envelope ORDER. Clearly state Order Code; Title; Quantity required; price & add any postage as above. Payment in STERLING by Cheque (UK bank), P.O. Euro Cheques or Cash. Please make cheques payable to FORMAT Publications. Payment MUST accompany order, do not send by different route.

Remember to quote your membership number or we wont be able to send out your order. DO NOT include letters with your order as this will cause delays. Goods will be sent out at the same time as the next available issue.

SMALL ADS

SWIFT DISC USER wants contact with other Swift Disc users. Also wants programs and utilities for Swift Disc especially Artist II 128k. Phone (evenings) 081-440-2602.

WANTED DISCiPLE (V3 ROM) or PLUS D. Also 3½ 80 track double sided disc drive with power supply. Write to M.Perry, 216 Marlpool Lane, Kidderminster, DY11 5DL.

WANTED The Rocky Horror Show (CRL) and Games Designer (Quicksilva. Will pay good price. Originals only. Also SAM Coupé PD software. Phone 0344-882718.

AMSTRAD 1512 with 30 Megabyte Internal Hard Disc, Dot Matrix 4000 Printer, Video Digitiser, Mouse, and over £1000 worth of software (mostly working packages - Wordcraft etc, but a few games). Whole package to first cash offer £800 for the lot. Phone 0452-715145.

48K SPECTRUM fitted with DK'Tronic keyboard, DISCiPLE interface, Single 3½ drive £140. Also DISCiPLE interface £50. Buyer collects. Please ring John Webb on Tunbridge Wells 21854.

YOUR ADVERT

Selling, Buying, Pen Friends, etc.
Any PRIVATE advert, up to 30 words (subject to acceptance), will be printed FREE in the next available issue. Any software sold must be original copies, in working order and with full instructions. The publishers will not be held, in any way, responsible for adverts in this column. Trade advertisers contact the publisher for rates.

Please Note:- If your advert remains unpublished after two issues please send it again.

** STOP PRESS **

Just a couple of things I had to squeeze in before going to press. First the good news is that a buy-out for the SAM project now looks set, so by the August issue we should have full details.

Next, I owe an appology to Juan B.Guillen Serra, last month we printed the Opentype file copier and I left his name off the article. John Wase had rewritten the article and added a few bits to it but the guts of the Copier were Juan's own work. I should have credited both of them for their work but forgot. Sorry Juan.

Carol Brooksbank has been in contact to clear up a few points on the SAM 'Small Is Beautiful' article that appeared in Volume 3 Issue 9. I'll leave you with her words.

Since I wrote 'SAM's Small Is Beautiful' I've upgraded my SAM Coupé to 512K. I realise now that the screen addresses I gave in the listings apply only to 256K machines with DOS. So, to make the listings run on any version of SAM (with or without DOS), the following changes should be made:-

Listing 2

```
 90 LET A=IN(252) BAND 31, SC1=(A+1)*
    16384: LOAD S$ CODE SC1
100 LET P$=MEM$(SC1 TO SC1+6912)
```

Listing 5

```
 70 LET A=IN(252) BAND 31, SC2=(A+1)*
    16384: LOAD "FAX1" CODE SC2
 90 LET A=IN(252) BAND 31, SC3=(A+1)*
    16384: LOAD "FAX2" CODE SC3
110 LET A=IN(252) BAND 31, SC4=(A+1)*
    16384: LOAD "FAX3" CODE SC4
170 LET P$=MEM$(SC2 TO SC2+6912)
190 LET P$=MEM$(SC3 TO SC3+6912)
210 LET P$=MEM$(SC4 TO SC4+6912)
```

If anyone has difficulty getting FLASH! to load MODE 1 SCREEN$, change lines 90, 170 & 260 of listing 4 to:-

```
LET A=IN(252) BAND 31: SC1=(A+1)*16384
: SAVE ... CODE SC1
```

GET THE MOST FROM YOUR DRIVE FACILITIES. *STEP UP TO*

NOT SO SOFT-WARE

SPECIFICALLY WRITTEN FOR PLUS D & DISCIPLE

SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE · SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE

FASTWORD PLUS

Here is a great programme that rejuvenates an old favourite. How often when using TASWORD TWO, have you been stuck for the right word? FASTWORD PLUS offers a complete reference book of words, waiting, ready for use at the press of a key!

This is the Plus D and Disciple Disc version of the sucessful program FASTWORD, a THESAURUS for the Spectrum. Features include ● Full alphabetical index of over 6,000 words ● A display of almost 600 screens ● Equivalent to a 120,000 word printed Thesaurus ● Sense identifaction for each word ● Select a word from the index using the first 2 letters and displayed on the screen is an average of 20 synonyms and associated words ● 2 way scrolling index ● Fast response ● Overlays Tasword 2 ● Words can be looked up in the Thesaurus as you are working on a text file in Tasword 2!

Also supplied is a NEW TASWORD 2 BASIC PROGRAM, specially written for Disc drive operation, offering new features such as selective catalogue, Wordcount etc. A CONVERTER program tailors your Tasword 2 code to give any combination of screen colours plus a new highly readable Tasword 2 character set.

Complete with instructions these 4 programs represent amazing value. Supplied on Disc. ONLY **£9.95**

FILE MASTER

At last! A comprehensive DataBase system that is VERY EASY to use. You will never forget the correct keys to press when using **FILE MASTER**, we guarantee it!

This versatile and powerful program can be used to store virtually any type of data. File Master enables you to create and maintain sophisticated data files, the type normally associated with more expensive computers. For example a file called 'DIARY' could be created. After typing the relevant data, the program can inform you which of your friends have a birthday in the next month, if the TV licence is due and the date of your next dental appointment. Records of money paid into the bank and cheques drawn could also be kept in the diary, and for an up-to-date total you simply press a key! The variety and scope of the files you can create is almost limitless and each file you create can have an infinite number of records (using segmented files).

Send for your **FILE MASTER** disc today. ONLY **£8.95**

THE ORGANISER

Features include ● Organisation of programs on any disc in any order you wish ● Makes finding programs in the directory much easier ● Selected programs load faster ● Checks for faulty sectors ● Automatic or manual Sector Repair ● Recovers erased files ... PLUS many more features to make file handling and disc 'housekeeping' so much EASIER.

This is what the experts said:-

"The ORGANISER has a very friendly user interface.... it's rare to find a package so easy to use.... brings out the best features in GDOS.... MGT should bundle it with SAM.... value for money...."
SID MARTIN in COMPUTER SHOPPER.

"Beautifully constructed and presented.... at £5 a bargain.... worth £10 or more just for the Disc Doctor aspects alone.... clearly to be recommended...."
Dr. JOHN WASE in POPULAR COMPUTING WEEKLY.

The ORGANISER is an essential program for ALL Plus D and Disciple users. Supplied on Cassette. ONLY **£5**

DISC MANAGER

THE DISC MANAGER is the most powerful program ever written for the DISCIPLE/PLUS D.

Designed to take advantage of Disc Drive ownership, the Manager keeps track of all the programs on all your discs, ● Storage of up to 27,000 records on one Disc, or 79,920 total. Random File Access. ● Name & Number Discs with fast Autonumber and user pre-defined titles features. No typing in of Data. Press a key and Discs are automatically added to appropriate catalogue. ● Fastest ever M/Code Search. 2 modes – Search and Load or Search and List all occurances, then select program to load. ● Plus many other unique features. ● Comes with 16 page manual and demonstration catalogues. ● Operates with 48K or 128K Spectrums. Send for the **DISC MANAGER** today... and you'll soon wonder how you ever Managed without it! NORMALLY £14.95

SPECIAL INDUG PRICE **£12.95**

BETTERBYTES

SPECIAL OFFER!
FREE ORGANISER TAPE
With All Orders
Over £20

10 SPITTAL TERRACE · GOSFORTH
NEWCASTLE UPON TYNE NE3 1UT · TEL: (091) 285 6185

IF OUTSIDE EUROPE SEND £1.50 p&p FOR EACH ORDER

SOFTWARE THAT IS SPECIFICALLY WRITTEN FOR PLUS D & DISCIPLE

PLEASE STATE YOUR DRIVE TYPE, SIZE, ETC, INDUG MEMBERSHIP No. AND IF REQUIRED FOR DISCIPLE OR PLUS D