FORMAT

A Magazine from INDUG
For DISCiPLE & PLUS D Users

ISSUE #9 - APRIL 1988
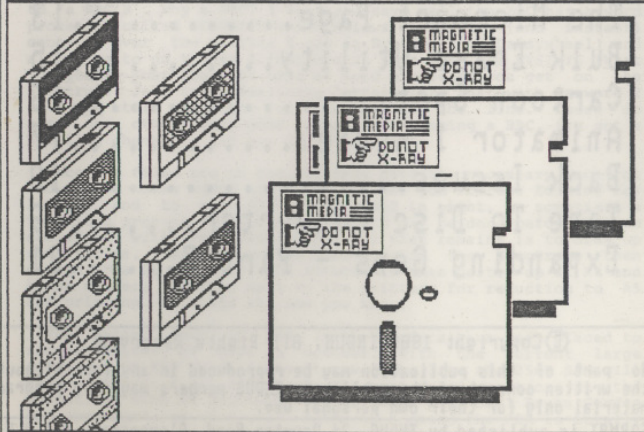
TAPE TO DISC
WE SHOW YOU HOW

INDUG.

CONTENTS

ISSUE #9 - APRIL 1988

The Editor Speaks

Issue number nine, time does fly when you're enjoying yourself. I do have great fun each month putting together an issue of FORMAT, and some people have asked just how I do it. Magic? well no, just plain hard work. Each issue takes about ten days, mostly under pressure to reach the printer by the agreed date. Articles and programs are stored on disc and filed in printed form when they come in. Each month I try to select as broad a spectrum of items as I can. Any program is tested on both the DISCiPLE and the PLUS D and some small amendments are made, if needed, to get it working on both.

The files and programs are then transferred, via an RS232 link to a BBC'B'. Why a BBC? I hear you say, well the BBC has a word processor called WORDWISE which allows me full (and instant) control over the printer I use (a Brother HR15 Daisywheel) and allows me to do things I just couldn't do with the WYSIWYG (What You See Is What You Get) type of word processor you get on the Spectrum. Just try underlining letters like that with Tasword 2 or 3 and still get right justification on the line. There is also one other very good reason for me using a BBC, its got a spelling checker - and boy do I need that.
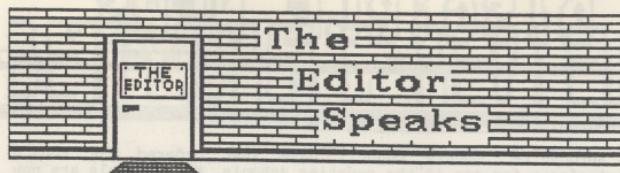
Once the files are on the BBC they have the necessary control and format codes added and then the real job starts. Editing for length, and to make things look good in print, is sometimes a very long process. Several test prints may be done before I am satisfied with the result. Now all that remains is to draw up some titles, some done on the Beeb some on the Spectrum, then design the front cover. All artwork is then pasted-up at A4 and these masters are then sent to the printers for reduction to A5 and printing. So thats it, now you know.

The latest news on SAM is that orders have been placed to secure delivery of chips in October. With the current large increase in RAM chip prices it now looks like the 256K SAM will appear at around the £130 mark. The most asked question, at the moment, is WILL MY DISCiPLE / PLUS D WORK WITH SAM? Bruce Gordon has promised me they will, although because the ROM is totally different from the Spectrums, there will need to be a new version of the DOS produced. I hope to be able to make an announcement on a priority order system for INDUG members in the next few months so keep reading.

Finally, where are those Hints & Tips? its been about three months since I last saw one. Is there anyone out there?

See you next month.   Bob Brenchley. Editor.

DISCiPLE NEWS.

SHROPSHIRE SCHOOLS GET THE PLUS D

Shropshire County Council have just ordered 250 PLUS D interfaces for use in the counties schools. Many schools are now buying Spectrums because they are so much cheaper then the competition (BBC Master computers). They recognise the importance of giving as many children as possible 'Hands On' experience with computers.

MGT believes this will be the first of many orders from around the country and would like to hear from anyone interested in, or already using the PLUS D in education.

OUTPUT FOR GAMES

Games enthusiasts may be interested in a new 'fanzine' called OUTPUT. The first issue (January '88) contained several very readable games reviews as well as an editorial and a Hints & Tips page. This first issue was well printed (if a little sparse) with good use of graphics. The editor, Simon Gardner, states that they intend to cover all aspects of the games market (arcade, adventure, simulation etc) as well as programming, hacking and hardware.

OUTPUT is published 6 times a year by Output Publications (S.Gardner), 30 Stonehouse Road, Liphook, Hampshire, GU30 7DD. Tel 0428 723042. A years subscription is only £2.40 including UK postage.

NEW TWO-WAY CONNECTOR FROM MGT

Very soon MGT will be launching a new TWO WAY expansion connector for the Spectrum. Bruce Gordon, who is designing the unit, sees it as the ideal adaptor for PLUS D owners. Priced at around £14.95 it will have a joystick port (Kempston compatible) and two edge connectors both fitted with switches to enable peripherals to be switched out. This will avoid conflicts between say the PLUS D and the VTX5000.

ON THE MOVE

MGT are about to expand by moving into new premises in South Wales. A move from the very cramped conditions they work under in Cambridge has been on the cards for some months. It seems likely that Swansea will be the new home for MGT and the expansion will enable them to produce the PLUS D in the larger numbers required to meet demand. It will also provide them with room to work on SAM. All the current staff will move in the next few weeks and the expansion will create several new jobs in the near future with more to come later this year when SAM production gets under way. We will have the new address and telephone number for you by next months issue.

BETA-BASIC-REVIEW

By: JOHN WASE.

Although Z80 machine code is assembled and run relatively easily on a Spectrum, and although it runs miraculously fast in comparison with Spectrum Basic, there are Big Snags! Its not as easy or quick to write and come next week it's difficult to see what it's doing. Basic does not suffer from this and if speed is unimportant and the program complex, it can often be better.

Of the Basics, Sinclair Spectrum Basic is particularly slow (on the PCW Benchmarks Index, Nov '87 there is only one machine listed which is slower), and although it is clear, clean and uncluttered, it is somewhat limited in respect of commands. So it's easy to end up with a "Slow Spectrum-Basic Spaghetti Program"! - Enter Beta Basic. Beta Basic feeds into part of the RAM and interacts with and enhances the existing ROM system by adding extra commands and functions, and modifying and extending many of the existing ones.

Describing all these in detail is well nigh impossible in the space available, with over seventy extra commands/keywords, many of them multiple (like BORDER 1, BORDER 2, etc). In addition there are some twenty-six extra functions which also appear as still further keywords, and the whole Sinclair system has been refined to make it easier to use. The program has a long pedigree (I have versions 1.8 and 2.0) leading to the current 3.0 (48K) and 4.0 (128K) versions, and now PLUS D/DISCiPLE disc versions have started to appear.

So, what makes these programs so useful? Well, try using the keyword CSIZE which changes character-size on screen (from 64 characters wide, or even more at a big pinch, to one huge letter). When you enter it, either for version 3.0 or 4.0, the editing line is a good old-fashioned Sinclair Basic 48K line at the bottom of the screen; none of this funny 128K mode with the appalling full page editor, redrawing the screen at each line entered and losing half the next line because it's too slow to keep up with your typing. But you can enter keywords either as single keystrokes, or as a combination of keywords and typed out words, or as words typed in full only. The choice is yours. Oh, and if your program line's syntactically incorrect, you get beeped at!

Once the line is entered in the program, you notice that the current line marker is much more prominent (white on black) compared with Sinclair's. It moves up and down the program quicker, too, and the screen scrolls more quickly than you were used to. There's a good RENUMber in the toolkit which enables you to renumber, copy, move or delete blocks of program, and you

can JOIN or SPLIT lines: (the latter's not really a keyword; <>
is used). If you've just entered a line, type zero and then a
line number; say 1020. The keyword EDIT appears thus: EDIT 1020
on the input line; on entering it line 1020 appears. And the
editing cursor now behaves properly, too, moving up and down as
well as side to side.

Much more important for programmers are the Basic enhancements
- enhanced loops (DO...LOOP, DO WHILE condition, DO UNTIL
condition, EXIT IF...) and IF statements (IF...THEN... ELSE); ON
(selects line number/statement) and LIST FORMAT which emphasizes
structure with "pretty" listings. You can debug using TRACE and
there is an ON ERROR command, too; very useful. But by far and
away the most useful of all is the very full implementation of
procedures (PROC, DEF PROC, END PROC), with LOCAL, REF
parameter, DEFAULT variable and ITEM function. So you can define
a procedure (DEF PROC somename), write it and finish with END
PROC: somename (or PROC somename) will call it from anywhere in
the program. Global and LOCAL variables can be used and
parameters are passed by reference; even arrays. This makes
programs much easier to read and follow and allows programmers
to develop program modules that can be inserted with ease into
new programs under developement.

There are improved SAVE, LOAD, MERGE and VERIFY, which should
deal with parts of programs or arrays (but see later) and so you
can save a library of complex procedures and use them in your
program, if necessary RENUMbering them after they have been
MERGEd; and if your disc syntax is complicated (try microdrive
syntax), then DEFAULT simplifies it. You can chop up arrays and
then join bits together without losing data, or or ALTER its
size without data loss. INARRAY and INSTRING will search arrays,
and LENGTH gives an array's dimensions. Finally, you can display
the results of your calculations beautifully with all the
enhanced Beta Basic graphic and display commands - GET a screen
area and PLOT the result back with ALTERed attributes somewhere
else. Do some advertising: a nice big coloured message can be
ROLLed round and round the screen: a longer one can be SCROLLed
right off, in the manner of the electronic displays one sees in
the shops these days.

Finally, if you want your dead serious results printed out
neatly in proper columns, then use PRINT USING - yes, it'll take
wildcards too, PRINT USING £##.##, for instance, prints out the
sums in real munney! All this lot has real syntax checking -
you've really still got your old original Spectrum, but with
extra keywords which behave just the same as the old ones -
whoopee! What's more, there are often considerable savings in
speed over Sinclair Basic: FOR-NEXT loops, GO TO's, GO-SUB's and
RETURNs are all quicker than you are used to.

The snag is that all this takes a goodly chunk of the 48K
Spectrum's memory, leaving about 22K for your Basic program. All
right; I know that you don't often write Basic programs of 22K
or more. But suppose, for instance, that you have a list of
names and addresses in an array. And you want to use, say, SORT
to find all the "Smiths" and INARRAY to locate all those living
in Oxford. If the program is very long, you haven't got room.

And that's where Beta Basic 4 comes in. Although the 128K
program appears at first to be the same, and all the extra
commands you had before are still there, it allows you to use,
in addition, a large RAMdisc file, with a special range of
commands, all ending in ! (shriek). So DIM ! joe$ (1000,60)
creates a 60K RAMdisc file which you can manipulate rapidly. For
instance, LET ! joe$(50) = "rhubarb" enables you to assign the
string "rhubarb" to element 50 of the array. PRINT INARRAY
("joe$ (a)","rhubarb") will find "rhubarb" in about two thirds
of a second per thousand strings: only about a quarter of a
second if position in the string is given. Even more important
are LIST ! and LLIST ! as this gives you the ability to LIST ! a
stream direct to a channel; i.e. to an Opentype file on
DISCiPLE/PLUS D, (or Microdrive, for that matter). This means
that you can set up files in RAM, manipulate them very quickly
(sorting a RAMdisc file using SORT ! is at a speed similar to
the SORT in version 3 in straight 48K RAM) and pour them back
into the disc: reloading is by INPUT !. Effectively this gives
you random access at a speed which many a PC owner would envy!
In addition, you have SAVE ! and LOAD !, MERGE ! and ERASE !
(same as in Spectrum 128K Basic), and finally CAT ! which also
now gives you the free RAMdisc space (78K max).

The 128K program also has some extras, apart from the RAMdisc
commands. DRAW is about 2.5 times faster than Sinclair's, and
will use the bottom two screen lines, as will the new PLOT:
CIRCLE is much quicker - by around twelve times - than
Sinclair's and you can fill an area with patterns by means of
FILL USING. The 128K mode keyword PLAY might allow you more
facilities than the old 48K BEEP (what wouldn't), but it still
stops the computer whilst the chip sounds. The new command, BEEP
!, is very versatile (it can have up to six specifiers - tone
period, duration, noise period, envelope, envelope period and
volume) and, best of all, is interrupt-driven, so nothing has to
stop whilst it is sounding.

As a dedicated Beta Basic user I am enthusiastic about the
programs. They provide an unbelievable number of facilities, and
user support is readily available through Beta Basic's own news
letter. There are one or two snags with the Disciple/+D systems:
LOADing and SAVEing program parts (e.g. procedures) is
difficult; DEFAULT doesnt work, either. Fortunately if you are
intending to get the program, their newsletter will keep you up
to date with any further "fixes" or enhancements. The newsletter
has attracted quite a following and is a real benefit to all
Beta Basic users.

The whole lot comes in a super plastic wallet with a tape and
88 page manual for version 3, closely printed and beautifully
produced with lots of helpful examples, and with a supplementary
32 page manual for version 4. The earlier versions were Sinclair
User Classics; the current superb ones at £15.95 must be a snip.

BETA SOFT,
24, Wyche Avenue,
King's Heath,
Birmingham,
B14 6LQ.

Dragons Lair 2.
A CONVERSION TO DISC

By: Hugh McLenaghan.

Dragon's Lair II is one of that growing band of programs termed 'Multi Loaders'. This means later screens / levels in the game are loaded in from tape as the game progresses. No 'snapshot' device yet invented can cope with converting this type of program to disc, so it was down to a bit of hard hacking in order to transfer the game to my DISCiPLE system. I hope the following will help not only owners of Dragons's Lair II but also users who would like to convert other games but dont know where to start.

The first thing I did was to look at the type of loading routine the program uses. In this case it looked like a normal loading routine, i.e. border colours are yellow/blue and the speed is the same as normal. Therefore I thought it probably used the Spectrums ROM routine at 1366 decimal (0556 hex). Protection was my next problem. To see if the BASIC loader had protection I tried to MERGE it. I got the message - 0 OK, 0:1 so the loader was not protected from MERGE.

When LISTed I got the following:-

```
10 REM DRAGON'S LAIR II
20 PAPER O: INK O: BORDER O: CLEAR 32767: POKE 23624,0: CLEAR:
RESTORE
30 POKE 23659,0: POKE 23614,0
40 LOAD "" CODE 16384
50 FOR F=0 TO 18: READ A: POKE F+64512,A: NEXT F: RANDOMIZE US
R 64512
60 DATA 221,33,0,129,17,0,75,62,255,55,205,86,5,218,1,129,195,
0,252
100 SAVE "ESCAPE" LINE 10
```

Line 30 is the nasty line! These two pokes cause the computer to crash if an error occurs, e.g. breaking into the program. Line 50 puts a machine-code loader into the memory. This is an important line as it told me where the code is loaded and it's length.

The data statement in line 60 contains the machine-code which is POKEd in at address 64512. The numbers 221,33,x,x (LD IX,nn) tell the ROM loading routine where the loaded code is to be put, in the above listing it is 0,129 which is 33024. The 17,x,x (LD DE,nn) gives the number of bytes to be loaded, 0,75 as above which is 19200 bytes. The next bytes 62,x (LD A,255) tells the ROM routine to load a CODE file and the 55 (SCF) indicates that it is a LOAD and not VERIFY. Following these bytes is the actual call to the ROM tape loading routine and the jump into the

9

machine code, we must replace these, from the 218, with 210,0,252,201,0,0 now delete line 30 and line 100 then add the following line:

```
70 POKE 23624,56: PAPER 7:CLS: SAVE d1"ESCAPE_C"CODE 33024,19200
```

The program was run and the tape started. After the main block of code had loaded it was automatically saved to disc just in case I needed to reload it later.

I then typed NEW and entered the following program which will search for the CALL to the tape loading routine in the Spectrums ROM:-

```
 10 FOR A=33024 TO 52223: LET Z=PEEK A
 20 IF Z=194 OR Z=195 OR Z=204 OR Z=205 THEN GOTO 100
 30 NEXT A
 40 STOP
100 LET AD=PEEK (A+1)+256*PEEK (A+2): IF AD>1365 AND AD<1400 TH
EN PRINT A:STOP
110 GOTO 30
```

I ran the program and waited for an address. It does not need to be exactly 1366 it could be slightly higher. After a short wait the program returned the value 34457. I then looked at the bytes before this until reaching either a 201, a 195,x,x or another jump instruction which normally will be the end of the previous routine. The byte directly after this is the byte which will be called from within the game to do its loading. I then had to search for these CALL(s) by substituting the following line:-

```
100 LET AD=PEEK (A+1)+256*PEEK (A+2): IF AD=34447 THEN PRINT A:S
TOP
```

I found the address was 34447 by looking at the bytes before 34457. The returned number is where the program will go to when it wants to load something from tape.

Looking through the code, the routine starting at 34408 jumped to this routine.

```
LD DE,(34466)
LD IX,38912
LD A,255
LD HL,34423
JR 34447
```

The value loaded into the IX register pair is the place where the code is to be loaded. The register pair DE holds the length of the block of data, it get the length from the header before the block of data. I then had to search for the CALL to this routine. As I did before, putting 34408 into the BASIC program listed earlier.

I then found the number 34310 which called this routine. Examining the bytes before the call, told me about the header and how it figures out if it is loading the correct section or not. I found that the bytes from 34294 to 34312 were the important bytes to consider as the call made at 34294 is to a

routine which loads three bytes into the memory which is the header. It loads the data into address 34465 to 34467. The dissassembly for the routine between these loaders is:-

```
34294 CALL 34388
34297 LD A,(34465)
34300 LD B,A
34301 LD A,(34469)
34304 CP B
34305 JR NZ,34294
```

I now had all the information that was needed to write the loading routine. Using the control codes the following loader was written:-

```
load: LD A,(34469) ; WHERE THE COMPUTER STORES WHICH LEVEL
      ADD A,48 ; CHANGE IT INTO A NUMERIC CHARACTER
      LD (NAM+5),A ; LOAD THE NUMBER INTO THE NAME
```

The rest of the listing is the same as in the DISCiPLE manual.

My next problem was where to put it. I searched and then decided to put it in one of the message areas. The most suitable area was starting at 34470. The routine "load" including the DISCiPLE loading routine was put here. I then had to search through the code to find any routines which pointed to areas within 34470 to 34605. And changed them so as not to call the print routine by making the bytes 205,68,134 equal to zero. After this the only change to the program is the message you get on the menu screen to "** DO NOT REMOVE DISC FROM DRIVE**". This message will be placed over the rewind tape message which it displays before the instructions which starts at 36630. I changed the call at 34294 to call 34471 and make all the bytes between 34297 and 34317 equal to zero. The code file was then saved.

A small program was then written to load the screens from tape and save them to disc. After running and transferring the levels I finally wrote a loader program and saved it to disc.

For those of you who want the easy way out I give below a conversion program. Type it in, then RUN and start the tape. The program loads each part in turn and saves it to disc, follow the prompts on screen to start and stop the tape.

```
10 REM     Dragon's Lair II.
20 REM Conversion to DISCiPLE/PLUS D disc systems.
30 REM By Hugh J. McLenaghan
40 REM
70 DEF FN h(h$)=((CODE h$(SGN PI)-VAL "48"-(VAL "7" AND h$(SGN
PI)>"9")))*VAL "16"+CODE h$(VAL "2")-VAL "48"-(VAL "7" AND h$(V
AL "2")>"9")
80 CLEAR VAL "32767": LOAD ""CODE VAL "16384": SAVE d*"ESCAPE_
S"SCREEN$ : LOAD ""CODE : RESTORE VAL "150": CLS : PRINT AT VAL
"10",VAL "10"; FLASH SGN PI;"PLEASE WAIT";AT VAL "5",VAL "9"; IN
VERSE SGN PI;"STOP THE TAPE"
90 READ addr
100 IF addr>VAL "65535" THEN GO TO VAL "240"
```

```
  110 READ h$: LET tot=NOT PI
  120 LET z=FN h(h$): LET h$=h$(INT PI TO ): POKE addr,z: LET add
r=addr+SGN PI: LET tot=tot+z: IF h$<>"" THEN GO TO VAL "120"
  130 READ t: IF tot<>t THEN PRINT "Typing ERROR": STOP
  140 GO TO VAL "90"
  150 DATA VAL "34117","000000000000",NOT PI
  160 DATA VAL "34124","1E",VAL "30"
  170 DATA VAL "34288","3AA5",VAL "223"
  180 DATA VAL "34292","0C87DD21EA86CF3B11F9860609CF3CJ21310FAED5
BFC86C30287000000000000",VAL "3066"
  190 DATA VAL "34325","01",SGN PI
  200 DATA VAL "34538","01000064044C6576656C312020202003001B00400
000FFFFED4BFA86CF3DC30E8600C63032F486C9",VAL "3572"
  210 DATA VAL "36632","2020444F204E4F542052454D4F564520544845204
.449534B2046524F4D205448452044524945564520202A2A202020",VAL "2817"
  220 DATA VAL "65e3","373E00DD21005B110400CD56053EFF37DD210098ED
5B015BD5CD5605C1C9",VAL "2880"
  230 DATA VAL "4E5"
  240 FOR a=VAL "34474" TO VAL "34537": POKE a,NOT PI: NEXT a
  250 FOR a=VAL "34578" TO VAL "34604": POKE a,NOT PI: NEXT a
  260 SAVE d*"ESCAPE_C"CODE VAL "33024",VAL "19200"
  270 CLS : PRINT AT VAL "10",VAL "10";"START THE TAPE"
  280 LET n$="level    ": FOR a=SGN PI TO VAL "7": LET len=USR V
AL "65e3": LET n$(VAL "6")=STR$ a: SAVE d*;n$CODE VAL "38912",le
n: NEXT a
  290 CLS : PRINT "All data saved. Now type in and save the loade
r."
```

This simple loader program should also be typed in.

```
  10 CLEAR VAL "32767": LOAD D*"ESCAPE_S"SCREEN$: LOAD D*"ESCAPE
_C"CODE: RANDOMIZE USR VAL "33025"
```

Now save it by: SAVE D1"ESCAPE"LINE 10 and the jobs done. Sit
back and enjoy the game, it will not make it any easier to play
but its much nicer not having to wait for the tape to load each
section.

GLITCH REPORT

Oh dear, I dropped a spanner in the works last month. The
excellent conversion of Art Studio by Villy Feltmann was just a
few lines too long for the space I allocated, so with a flash of
the word processor I made it shorter. But I also made it NOT
WORK... In altering lines of the hexloader, line 50 come out
wrong. It should read:-

```
  50 FOR b=1 TO len/2: LET byte=16*(CODE b$(1)-48-(7 AND b$(1)>"
9"))+(CODE b$(1)-48-(7 AND b$(1)>"9")): POKE a+b-1,byte: LET che
ck=check+byte: LET b$=b$(3 TO ): NEXT b
```

The difference being the 'len/2' in the first statement instead
of 'len STEP 2'. My apologies to everyone who tried the
conversion and failed through my error.

                                                          BOB.

THE MICRONET PAGE

A NEW MONTHLY FEATURE

                                        By: Patrick McMahon.

Over the next few months I am aiming to give you, the reader,
an in depth look at Micronet; how it functions, its facilities,
and more importantly its users and IP's (Information Providers).

Once you have got your VTX 5000 modem plugged into your
Spectrum you are all set to access Micronet. This simply
involves dialling up one of the Prestel computers (there are two
for the South east area, Enterprise and Derwent) through your
local access number and logging on by entering your 10 digit ID
number and your personal password which can be a combination of
four figures (numbers or letters). With all that over you are
presented with your first proper Micronet page which welcomes
you to the Database by name and tells when you last called.
Keying # (Enter) takes you to the main menu (fig 1) which
displays an extremely general index of what is a vast Database,
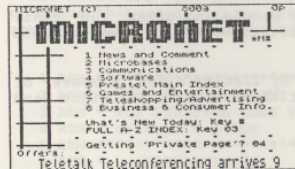the index proper takes literally hundreds of pages of text.

Fig 1
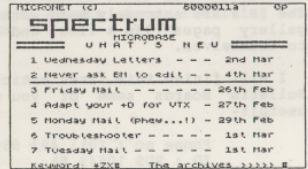
Fig 2

To get to the Spectrum Microbase you just have to key *ZX#,
you are then presented with the "What's new" index (fig 2). This
usually consists mainly of Letters from Micronetters with
problems with their system, Micronet or just airing their views
about life in general. Reviews of software and hardware are also
common, most recently there have been reviews of the DISCiPLE,
+3 and the PLUS D. There is also a spot for soldering, to get to
this you key *SOLDER ON#, which provides easy step by step
instructions on how to repair or modify your hardware. A recent
article was how to make your PLUS D work with the VTX 5000 modem
by means of a sort of inhibit button (Warning, this may
invalidate your guarantee). Talking of VTX 5000 modems; Micronet
are doing quite a good offer on them at the moment. If you
subscribe to Micronet for one year they will give you a FREE
modem, not bad eh!

A good introduction to using Micronet is the Gallery where

IP's update regularly pages of text and graphics which provide many different items such as free programs one example is the Sky software gallery pages and the Soft Option . Many just specialize in providing free telesoftware on their pages eg Phantom Viewdata. Others concentrate on producing hardware and software reviews for other people on Micronet to read and enjoy. A few IP's but not many have graphic pages on their gallery pages, one example is Simon Grant Graphics (GASP fig 2 & 3) which has a very good selection of some the best graphics available on Micronet. Many of the IP's also run Bulletin Boards (a similar idea to Micronet but on a smaller scale) Keith Burton is one example with his Phantom viewdata and Phantom gallery pages.
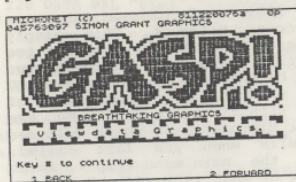
Fig 3
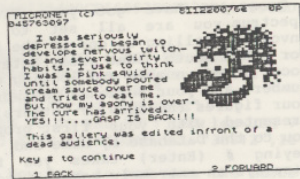
Fig 4

The IP's are extremely hard working people and often run their gallery pages (99p a time) and Bulletin Boards at great expense to themselves.

I will finish this first article by listing a few numbers of Bulletin Boards some of you out there with modems might find useful:-

        Phone no : (01) 493 9555
        ID : RUS 10549 BA

        Phone no : (061) 844 1999
        ID :647870344
        Password : 482933

        Phone no : (0800) 282301
        ID : none

I must stress that you need special software such as Firescroll or Micron to access Bulletin Boards. The normal VTX software will not work, but do not worry, this software can be downloaded free from Micronet.

For further details on Micronet membership contact:-

MICRONET,
Durrant House,
8 Herbal Hill,
London,
EC1R 5EJ.
Tel 01-278-3143

BULK-ERASE

By: John Nixon.

With up to eighty files per disc the DISCiPLE / PLUS D sometimes makes life a little difficult for us poor users. With it's powerful wildcard facility it is very easy to issue a command like ERASE d1"DATA*" only to find you have ERASEd one file too many. Using a separate ERASE command for each file is both long winded and prone to errors due to mis-typing.

This utility is designed to take the stress out of clearing up your discs. When run, it creates a catalogue of the disc in an array, it then steps through displaying each filename on screen and asking you to ERASE Y or N ?. If you reply no then the filename is removed from the array. At the end of the loop through the array, if you have said yes to any files, a list of files to be ERASEd is displayed and you are asked to confirm that the list is OK.

The files are then deleted using a Basic line, this avoids any problems that might arise through errors from rewriting an amended directory sector by sector.

```
   1 REM ***********************
   2 REM * BULK ERASE UTILITY  *
   3 REM * For DISCiPLE GDOS V3*
   4 REM * or PLUS D G+DOS      *
   5 REM ***********************
   6 REM *   (C)1988 INDUG.    *
   7 REM ***********************
   8 REM
  10 CLEAR
  20 POKE 23658,8
  30 CLS #
  40 LET P$=CHR$ 17+CHR$ 5+CHR$ 19+CHR$ 1+" DISCiPLE BULK-ERASE
PROGRAM V1 "
  50 PRINT P$''"Insert DISC into drive 1."
  60 PRINT #0;"press any key when ready."
  70 PAUSE 0
  80 CLS: PRINT P$'' FLASH 1;" CREATING CATALOGUE. "
  90 POKE 50256,0
 100 DIM f$(80,10): LET F=0
 110 FOR I=0 TO 3: FOR J=1 TO 10
 120 LOAD @1,I,J,50000
 130 FOR K=50000 TO 50256 STEP 256
 140 IF PEEK K=0 THEN GOTO 170
 150 GOSUB 460
 160 LET F=F+1: LET F$(F)=T$
 170 NEXT K
```

```
180 NEXT J: NEXT I
190 IF NOT F THEN PRINT "RUN ABORTED.  NO FILE FOUND": STOP
200 LET N=0
210 FOR I=1 TO F: CLS: LET A$=F$(I)
220 CLS: PRINT P$''' PAPER 1; INK 7;" File name :- ";
230 PRINT ;" ";A$
240 PRINT '''"      ERASE  Y OR N?"
250 LET I$=INKEY$: IF I$="" THEN GOTO 250
260 IF I$="N" THEN LET F$(I)="            ": GOTO 290
270 IF I$<>"Y" THEN GOTO 250
280 LET N=N+1
290 IF INKEY$<>"" THEN GOTO 290
300 NEXT I
310 IF N THEN PRINT ''''"END OF CATALOGUE": GOTO 330
320 CLS: PRINT P$''''''" NO FILES SELECTED"'''"   RUN ABORTED.":
GOTO 9999
330 CLS: PRINT P$'' INVERSE 1;"FILES TO BE ERASED"'''
340 FOR I=1 TO F: IF F$(I)<>"            " THEN PRINT F$(I)
350 NEXT I
360 INPUT FLASH 1;"ARE YOU SURE ?  Y or N ";I$
370 IF I$(1)="Y" THEN GOTO 400
380 IF I$(1)="N" THEN STOP
390 GOTO 360
400 CLS: PRINT FLASH 1;" ERASING FILES "''
410 FOR I=1 TO F: IF F$(I)="            " THEN GOTO 440
420 ERASE d1;F$(I)
430 PRINT "ERASED ";F$(I)
440 NEXT I
450 GOTO 9999
460 LET T$="": FOR L=K+1 TO K+10: LET T$=T$+CHR$ (PEEK L): NEXT
L: RETURN
9998 CAT 1: STOP
9999 SAVE d1"BULKERASE" LINE 20
```

"Somehow this just doesn't feel like
paradise without a computer."

"If anyone calls, tell them I'm not in."

"Frankly, I liked the old games better."

16

ANIMATOR 1
A REVIEW

By: Jess Sulivan.

ANIMATOR 1 is a powerful piece of software intended primarily for the design and animation of sprites. The features that make for a fine animation package also serve to set the ANIMATOR 1 apart from most other art packages available for the Spectrum range of computers. Most IBM compatible art software is also hard pressed to equal some of the stunts that are possible with ANIMATOR 1.

This review will not go into depth on all the features of ANIMATOR 1 - you'll have to see for yourself - but a quick list goes something like this:

LINE FUNCTIONS - single lines, pinned lines, rubberbanding and freehand.

CIRCLES - You can put a circle of any size anywhere on the screen. These circles can even run off of the screen if necessary.

DRAWING - Controlled from the keyboard, once you get the hang of it you will be surprised at the ease and speed with which things can be drawn accurately.

MAGNIFICATION - A particularly handy option is the magnification window. This window occupies about 1/6 of the screen but can be moved anywhere you desire. If switched off at a particular position, it will still be there the next time you switch it on. This window is perfect for close in work.

SCREEN SCROLLING - The cursor keys act to move the screen in any desired direction one row or column or row and column at a time. The screen can also be rotated by 90, 180, 270 or 360 degrees at the touch of a key.

ATTRIBUTES - The Spectrum screen attributes are fully controllable within ANIMATOR 1 even to the extent of having different paper and ink colours at every character location on the screen. You can also 'INVERSE' the screen and in effect, print text or draw on the back of the paper and then inverse it again to see the actual result.

SPRITES - This is the most interesting feature of ANIMATOR 1. You can store and animate up to 256 sprite images (not just the bit image, but the attributes as well). If you define your sprites as being full screen size, this gives you two screens that are stored in memory and be called at any time. This gives you three screen on which to work. The sprite function also

17

serves as a cut/paste facility and a library of designs or pictures can be saved to disc.

TEXT HANDLING - Brilliant. You are able to print text anywhere on the screen in any size. Using the DISCiPLE or PLUS D screen dump facilities, you can create an A4 size poster in about 1 minute if you haven't much to say or about 10 minutes if you are quite laquacious.

CHARACTER SET/UDGs - The Animator allows you to redesign any of the characters or UDGs under automatic magnification. You can also redesign the four standard pen sizes, the eight brush shapes and any one of over 65000 hatch patterns. Once defined, these can all be saved to disc for future use.

COMPATIBILITY - As the ANIMATOR 1 program uses Microdrive commands throughout it is perfectly compatible with your DISCiPLE or PLUS D. Softcat Micros has released a special version just for DISCiPLE and PLUS D users which incorporates an auto save to disc function. This version is available only from MGT.

This is only a taste of the good things that the ANIMATOR 1 program is capable of doing. On the down side, there a couple of potential problems. Firstly, the manual is not as concise as it perhaps could be. Before you fully understand the package, a wrong keypress could have disastrous results. Also, some mouse users may find the lack of a mouse or even joystick option annoying. Aside from these two things, I found ANIMATOR 1 to be fast and easy to use. It is a definite must for anyone working extensively in graphics and quite a lot of fun as well.

Title: ANIMATOR 1
Supplier: Softcat Micros
Seller: MGT - 0223 311665
Price: £9.95 (£8.95 to INDUG members, see special offer leaflet)

* - * - * - *

BACK ISSUES

For members who have missed past issues of FORMAT (or perhaps worn theirs out through constant use) we run a back-issue service.

The cost is 65p per copy (85p overseas) which includes postage. Your copies will be sent out as soon as possible but, in order to keep printing costs down, it may take up to 21 days for us to dispatch.

Available Issues

Issue 1 - August 1987.   Issue 5 - December 1987.
Issue 2 - September 1987.   Issue 6 - January 1988.
Issue 3 - October 1987.   Issue 7 - February 1988.
Issue 4 - November 1987.   Issue 8 - March 1988.

TAPE TO DISC

By: Steve Nutting.

This REM listing is to be used in conjunction with the Data Compiler see issue #7. DISCiPLE owners miss out lines 200 onwards, PLUS D users with G+DOS version 2 miss out lines 180,190,260 (version 1 users miss lines 180,190).

Once you have typed in the REM listing Save it to Disc by:-
SAVE d1"temp" then load in the Data Compiler by CLEAR 29999: LOAD d1"datacomp" CODE then type RUN. If there are no errors then type:-

RANDOMIZE USR 30829: SAVE d1"TAPE-DISC" CODE 0,6656 (DISCiPLE)
RANDOMIZE USR 30844: SAVE d1"TAPE-DISC" CODE 8192,6656 (PLUS D)

This will save a special version of the System file for future use.

With the new System File loaded we can call up the Snapshot key routine. Press the Snapshot button then key 1 (DISCiPLE) or key 0 (PLUS D). You will then see PLAY TAPE printed on the top screen So press play on your tape recorder. After the first header block loads you will see the Filename and the relevant information on the program file type. If FILE TYPE ERROR is printed you have attempted to load a file which is not Basic, Numeric Array, Character Array or Code. Once the next data block is loaded the currently used Disc Drive (D*) will start up with the tape file automatically transferring to disc. If LOADING ERROR is printed then the tape program just loaded is corrupted and will not be transferred. If when a tape program is loading you wish it not to be transferred to disc press Space or Break.

The routine will even copy machine code files 16384,49152 which use all Spectrum memory.

```
1 CLEAR 29999: RANDOMIZE USR 64512
5 REM s30000
10 REM 49,255,255,33,175,23,6,12,205,70,23,33,180,24,17,0,255,1
,170,0,237,176,55,62,0,221,33,122,23,17,17,0,205,0,255,210,87,23
,215,107,13,33,123,23,6,10,205,70,23,62:4419
20 REM 32,215,16,0,58,122,23,167,32,30,33,187,23,6,12,205,70,23
,237,75,133,23,33,6,1,205,150,24,237,75,135,23,33,6,2,205,150,24
,24,74,254,1,32,31,33,199,23,6,20,205,70:4003
30 REM 23,237,75,133,23,33,14,1,205,150,24,205,160,24,58,136,23
,214,64,215,16,0,24,39,254,2,194,255,23,33,219,23,6,21,205,70,23
,237,75,133,23,33,16,1,205,150,24,205,160:4686
40 REM 24,58,136,23,214,128,215,16,0,62,36,215,16,0,42,135,23,3
4,141,23,33,203,92,34,135,23,55,62,255,221,42,135,23,237,91,133,
23,205,0,255,210,87,23,33,123,23,17,103:4417
50 REM 23,1,10,0,237,176,19,1,8,0,237,176,58,122,23,50,113,23,6
```

```
0,50,102,23,61,167,40,10,254,1,40,14,254,2,40,15,24,18,42,141,23
,34,120,23,24,43,33,64,0,24,29,33,128:3213
60 REM 0,24,24,221,33,98,23,205,79,21,237,91,116,23,237,75,114,
23,205,86,21,205,129,41,195,33,22,34,118,23,33,255,255,34,120,23
,33,203,92,34,116,23,24,215,229,197,62:4704
70 REM 2,215,1,22,193,225,126,215,16,0,35,16,249,201,33,143,23,
6,15,205,70,23,195,33,22,1,0,0,100,0,32,32,32,32,32,32,32,32,32,
32,0,0,0,0,0,0,0,0,0,0,32,32,32,32,32,32:2897
80 REM 32,32,32,32,0,0,0,0,0,0,0,0,0,0,13,13,76,79,65,68,73,78,
71,32,69,82,82,79,82,13,13,70,73,76,69,32,84,89,80,69,32,69,82,8
2,79,82,22,0,11,80,76,65,89,32,84,65,80:2788
90 REM 69,13,66,65,83,73,67,13,32,65,85,84,79,13,78,85,77,69,82
,73,67,32,65,82,82,65,89,13,84,89,80,69,32,13,67,72,65,82,65,67,
84,69,82,32,65,82,82,82,65,89,13,84,89,80,69,80:3467
100 REM 69,32,13,77,65,67,72,73,78,69,32,67,79,68,69,32,254,3,4
0,11,33,158,23,6,17,205,70,23,195,33,22,33,241,23,6,14,205,70,23
,237,75,135,23,205,153,24,62,44,215,16,0:3859
110 REM 237,75,133,23,205,153,24,42,135,23,237,91,133,23,25,122
,254,2,56,65,124,167,40,5,254,255,194,209,22,21,237,83,80,24,37,
34,109,24,55,62,255,221,42,135,23,17,0,0:4787
120 REM 205,0,255,221,33,214,27,17,0,1,62,255,243,205,83,255,25
1,210,87,23,49,120,25,33,214,27,17,0,0,1,0,1,237,176,195,226,22,
42,135,23,229,62,255,55,221,33,64,156,237:5502
130 REM 91,133,23,205,0,255,33,64,156,209,237,75,133,23,237,176
,195,226,22,205,163,24,215,43,45,215,227,45,201,33,5,2,197,62,22
,215,16,0,124,215,16,0,125,215,16,0,193:5332
140 REM 201,0,20,8,21,243,62,15,211,254,33,168,255,229,219,254,
31,230,32,246,2,79,191,192,205,138,255,48,250,33,21,4,16,254,43,
124,181,32,249,205,134,255,48,235,6,156,205:6293
150 REM 134,255,48,228,62,199,184,48,224,36,32,241,6,201,205,13
8,255,48,213,120,254,212,48,244,205,138,255,208,121,238,3,79,38,
0,6,176,24,24,8,32,5,221,117,0,24,10,203:5770
160 REM 17,173,192,121,31,79,19,24,2,221,35,27,8,6,178,46,1,205
,134,255,208,62,203,184,203,21,6,176,210,109,255,124,173,103,122
,179,32,209,124,254,1,201,205,138,255:5531
170 REM 208,62,22,61,32,253,167,4,200,62,127,219,254,31,208,169
,230,32,40,243,121,47,79,230,7,246,8,211,254,55,201,251,201:4535
180 REM 243,219,187,33,172,0,34,170,0,33,33,22,34,165,0,34,164,
6,235,33,144,101,1,87,3,237,176,211,187,251,201:3416
190 REM e
200 REM 203,99,202,49,33,6,239,237,88,203,67,192,195,212,48,207
,71,62,195,50,46,33,33,17,52,34,47,33,33,48,117,17,212,48,1,132,
3,237,176,195,80,0:4252
210 REM p
220 REM 4,5,3,15,9,6,5,8,6,5,4,6,4,6,9,5,4,6,3,3,12,3,5,4,6,3,3
,13,3,6,7,4,6,3,3,14,3,4,17,3,16,7,4,9,3,6,6,22,5,3,164,5,3,3,5,
4,3,9,3,3,4,15,5,4,7,23,17,3,12,11,5,3:630
230 REM e
240 POKE 30257,161: POKE 30258,47: POKE 30268,168: POKE 30269,4
7: POKE 30271,137: POKE 30272,11
250 POKE 30566,214: POKE 30567,59: POKE 30582,44: POKE 30583,52
: POKE 30585,214: POKE 30586,59
260 POKE 30271,125
```

Coming soon, yet another System File which will give you extra
Basic commands. As usual no spectrum memory will be used, its
all held in the modified System File. Bye for now.

EXPANDING GENS

PART 3.

By: DAVE KENNEDY.

This months quota of EXPANDING GENS contains the start of the
new commands, but first a small correction to last months
listing. At line 1700 (page 18) change the 'DEFW' command to
'JR', the mods will then run correctly.

One of the commands I missed in the original GENS was a block
line mover, it makes reorganizing an assembler listing or
loading listing modules much less time consuming. My routine
acts as a replacement for the "T" dump command which I have not
converted to disc SAVE. I use the "R" run command for this block
save since I prefer to test my code under monitor control (I use
the ULTIMON monitor program rather than MONS3 since it has a Run
command with break out option). This new command needs three
arguments, FIRST line number of the move block, LAST line number
of the move block then NEW LOCATION line number. The block will
be stored in screen memory, erased from the textfile then
transferred to the move-to line number. The first part of the
listing converts these arguments to the appropriate textfile
addresses and stores them. The move-to number will be in the
assembler filename buffer. All these values are then error
tested and if the move block will fit into screen memory the
program continues with the actual transfer operation which is a
simple code block move routine. The direction of line transfer
dictates whether to use the LDIR or LDDR function. The block of
lines, safely stored in screen memory, is then transferred back
into the textfile and the whole textfile will then be renumbered
with a step interval ten.

The LISTER routine from line 2660 gives an amended printer
listing which I think is much more legible. Any comments are
moved to column 36 and printed in lower case. The printer is set
to the U.S. character set ("#" instead of "£") and eight lines
per inch. I have also incorporated a printing break option as I
find that if you press the Break key with an interrupt-driven
program it will display the GDOS break message then crash.
Keeping any key pressed will halt printing when a line end is
reached and a return to GENS input mode made. If you want to
continue with the listing then just reuse the "W" command with
new arguments.

The first part of LISTER will find the start and end addresses
of the textfile block to be printed. If the start line input is
larger than the end line, then the program jumps to the page
lister routine (LIST3 at line 2850) which prints the whole
textfile in separate, numbered pages with (NUM1) lines per page
and (NUM2) newlines between each page. Line 3560 is the start of
the line number print routine using the ROM stack print routine.

Character 9, the right space key, is used as a separator within the textfile and is entered every time a space or right cursor is entered. The program searches for these separators as well as the ";" character which denotes line comments. Each part of the line is neatly printed using the TAB print command as necessary. The last routine modifies the "T" command to be a new program exit. As you will see from my set-up at Basic line 1, whenever the assembler "B" command is used I enter the Ultimon monitor program and when that is exited I return to GENS. The routine just copies the "B" command then exits with a Stop error.

All these routines are incorporated within the GENS program whilst retaining the full relocatability of the original.

In this short series we are converting HISOFT's excellent GENS assembler to use the DISCiPLE / PLUS D disc system to the full.

The first part of GENS at 27000 is the relocation routine which is why when you re-enter the program you must never recall from the load address. This takes the address table at the end of the loaded GENS and adds each address to the contents of the "BC" registers so forming an address within GENS. Here will be found an address giving the distance from the start of GENS of the routine to be called or jumped to. If this relative value is added to the "BC" value and stored back into GENS you will have formed the required call or jump address regardless of the original load position. The address table at the end of GENS is saved with the code but will be overwritten by any textfile entered. To save the code, enter both listings into the high memory GENS (PUT first to prevent undefined symbol errors when assembling) then SAVE code 26580,9645. This code can then be loaded at any memory address and called from that load address. The RELO code at line 2430 will compare locations within GENS with the "BC" load address value to decide whether relocation is required, or a warm re-entry made. The appropriate address is stacked and a RETurn made.

I hope that the above gives you some ideas on modifying the GENS program, there is no reason to use the routines exactly as listed they can be changed as required. I find it easier to test them as normal code then incorporate them into GENS with the necessary relocation symbols. The starting point for any alterations to GENS is to find the command jump table near the top of GENS(#854A in my version) and disassemble and alter the code as required.

```
2420 ;****************************************
2430           ORG 26580       ;ends 2 bytes below "gens"
2440 RELO      LD HL,DK-RELO   ;this address
2450           ADD HL,BC       ;= start of gens code
2460           LD B,H          ;alter "bc" to
2470           LD C,L          ;this value for relocation
2480           LD HL,#001E     ;1st. "jp" value in gens at #6996
2490           ADD HL,BC
2500           PUSH BC         ;save start of gens
2510           LD E,(HL)
2520           INC HL
```

```
2530           LD D,(HL)       ;"de" = contents of this value
2540           EX DE,HL
2550           AND A
2560           SBC HL,BC
2570           RET C           ;jump to start of gens ("bc" on stack)
2580                           ;if "hl"< "bc" - start of gens
2590           POP BC          ;empty stack
2600           LD HL,61
2610           ADD HL,BC       ;= warm restart location within
2620           PUSH HL         ;gens stack this value &
2630           RET             ;jump to it.
2640
2650
2660 LISTER    LD A,3          ;"w" = llist command
2670           CALL #1601      ;select line printer
2680 LIST1     LD B,1          ;initialise printer settings
2690 M1        CALL PRINTER-DK ;lprint from list
2700 M2        CALL FNDADR     ;find addrs of start & end lines
2710           JR C,LIST3      ;if start line H end line then
2720           JR Z,LIST3      ;goto single-page print llister
2730           ADD HL,DE       ;else reform addresses
2740           EX DE,HL        ;"hl"=start addr / "de"=end addr
2750 LIST2     PUSH DE         ;save end position
2760 M3        CALL LINEPRINT-DK ;print out one line
2770           POP DE          ;recover stop print address
2780           RET NZ          ;abandon print if any key pressed
2790           AND A
2800           SBC HL,DE       ;current posn. - end posn.
2810           ADD HL,DE
2820           JR C,LIST2      ;onto next line
2830           RET             ;unless curent H end posn.
2840
2850 LIST3     LD HL,(TXTSRT)  ;llist whole textfile
2860           XOR A           ;zero page number counter
2870 M5        LD (BUFFER+3),A ;page counter
2880 LIST4     CALL HLEND      ;compare "hl" & (textend)
2890           RET NC          ;exit if no textfile
2900           LD B,2          ;set to underline mode
2910 M7        CALL PRINTER-DK
2920           LD BC,#023B     ;check asm. input buffer for at
2930 M8        LD DE,BUFF2     ;least two commas (e.g. w80,2,test)
2940 LIST5     LD A,(DE)       ;in main text buffer
2950           INC DE
2960           CP 13           ;test for only one comma
2970           JR Z,LIST7      ;& therefore no heading to print
2980           CP ","
2990           JR NZ,LIST5
3000           DJNZ LIST5
3010 LIST6     LD A,(DE)       ;now at start of heading
3020           CP 13           ;newline at heading end
3030           JR Z,LIST7
3040           RST 16          ;print heading
3050           DEC C           ;spacer counter
3060           INC DE
3070           JR LIST6
```

Thats all I've got room for this month but I will be back next month with the next installment.