

disk USER

**BBC MICRO
MODEL B
MODEL B+
MASTER I28**

ALL DFS
FILING
SYSTEMS

On the Disk

WE HAVE LIFT-OFF
– THE ESSENTIAL DISK ROUTINES
MAGIC WALL PUZZLE CHALLENGE
PICK UP A PRIZE
SAVE YOUR BACON IN THE ARCADE
NAME THAT THEME TUNE
INTERACTIVE TUTORIALS
ANIMATIONS
UTILITIES

In the Magazine

HOW YOUR DISK DRIVE WORKS
BEGINNER'S GUIDE TO
DISK HANDLING
DISK SOFTWARE REVIEWED

FROM THE PUBLISHERS OF

**A&B
COMPUTING**

ANOTHER FIRST FROM THE SPECIALISTS IN MAGAZINE PUBLISHING

Disk User was the first magazine to combine the power of print and the immediacy of a computer disk to provide the complete leisure and hobby magazine for owners of the BBC Microcomputer. Both the reader and the advertiser gain the benefits of a totally interactive package. Disk User opens up exciting new avenues of exploration for all its readers while introducing a new level of user friendliness into the computer magazine market.

AN ARGUS SPECIALIST PUBLICATION OCTOBER/NOVEMBER 1987 £2.95

disk USER

**BBC MICRO
MODEL B
MODEL B+
MASTER I28**

ALL DFS
FILING
SYSTEMS

**UTILITIES REVIEWED
DISCUSSION
CHOLO SOLUTION**

On the Disk

**LIFE SAVER - BACKING-UP
IN THE BACKGROUND**

**FULL SCIENTIFIC
CALCULATOR**

**INCREDIBLE SHRINKING
SOFTWARE - THE ULTIMATE
COMPACTER**

**BLAST YOUR WAY OUT
IN TAZMAN**

**LEARNING MATHS
MADE FUN**

**CHOLO EDITOR
DOUBLE CATALOGUE**

**disk
USER** For all DFS filing systems
40 tracks convertible to 80 tracks
To start press SHIFT/BREAK

ISSUE 3 OCTOBER/NOVEMBER 1987

**Software
worth
£25+ on
disk
ready to GO!**

FROM THE PUBLISHERS OF
**A&B
COMPUTING**

Argus Specialist Publications Limited,
1 Golden Square, London W1R 3AB
Telephone: 01-437 0626 Telex: 8811896
Distribution by SM Distribution Limited

disk USER

Number Five
February/March 1988

Editor: Mark Webb
Assistant Editor: Clive Grace
Software Assistance: Matthew
Fifield
Advertisement Manager: Sarah
Musgrave
Advertisement Copy Control:
Francisca Perez

IN THE MAGAZINE

Disk Instructions 4

How to get the best out of Disk User – including 40 to 80 track conversion. Make sure you get the full picture by reading these pages before using your Disk User disk

Disk News 6

Latest releases and news for disk users

Discussion 8

News and views from Disk Users everywhere

How your disk drive works 14

The inside story

Disk Data 18

Disk software reviewed

Program Transfer 20

Getting the best from your copy of Disk User

Floppy Fun 21

Dave Reeder with help on Arkenoid and Thrust plus full Ravenskull maps on disk

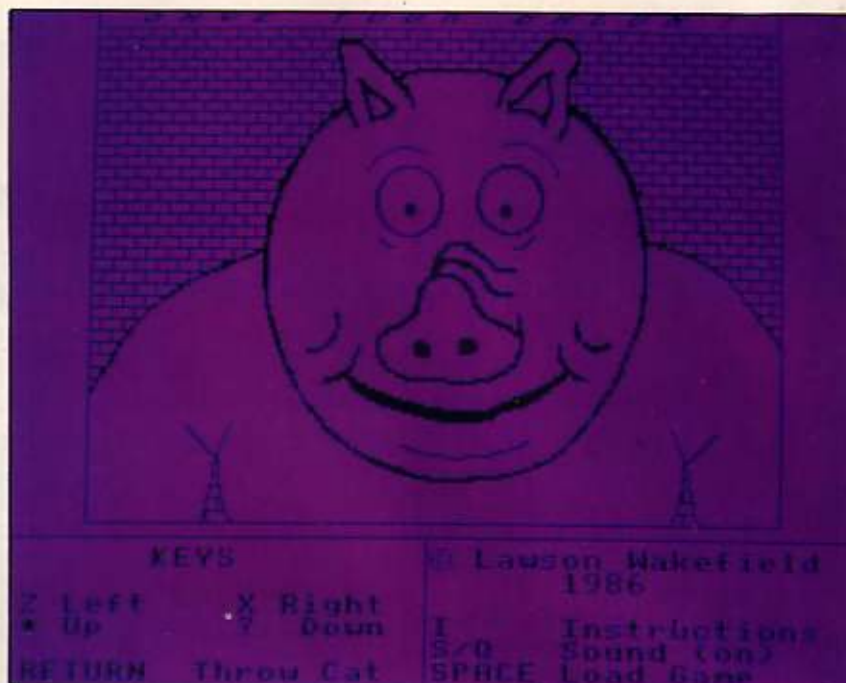
DISK DOCUMENTATION

Magic Wall 39

An infuriating puzzle for you to play. And pick up a prize if you're good enough!

Save your Bacon 10

Games time on Disk User. An amusing and colourful arcade game



Collectors Items 17

Animations for you to copy and keep. Build a collection from this entertaining A to Z of animations by computer artist Abbas

Name that Theme 17

Ian Waugh's tip for the top, the Disk User jingle you will all be humming!

Fully Functional 22

An essential set of disk programming routines. Your programming productivity will take off!

Overlay Techniques 24

Tutorial on overlay techniques, including program examples

Interceptors 27

Find and Replace for BASIC programs

Spock's Menu 30

Spock logic incorporated into this easy to use menu system

Toolkit 33

Character editing and special disk formats

Disk User is published bi-monthly on the third Friday of the month preceding cover date. Editorial and advertising enquiries to Disk User, Number One Golden Square, London W1R 3AB. Telephone: 01 437 0626. Distributed by SM Distribution Ltd., 16-18 Trinity Gardens, London SW9 8DX. Telephone: 01 274 8611. The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Ltd. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Ltd. Any reproduction requires the prior written consent of Argus Specialist Publications Ltd.

Cover design and design by the ASP Design Studio 01 631 3888
Printed by Riverside Press, Barnstaple, Devon
Disk duplication by Disk Copy Labs
ISSN 0264-4584

Copyright © Argus Specialist Publications Ltd 1987

Disk User is supplied on a 40 track disk format and can be run without conversion on a 40 track drive.

If you have 40/80 switchable drives then make sure the drive is switched to the 40 option.

For 80 track only drive owners, a conversion program is provided – see Disk Instructions.



DISK INSTRUCTION

DISK INSTRUCTIONS

To get the best from your copy of *Disk User*, please carefully read the instructions below. We have made *Disk User* able to run on a very wide range of systems.

One point to note is that we **strongly recommend copying the disk on to a blank formatted disk before you use it**. You should use this copy as your working copy, and keep the original as the back-up. Many of the programs require to write to the disk, and doing this will diminish the usefulness of the original.

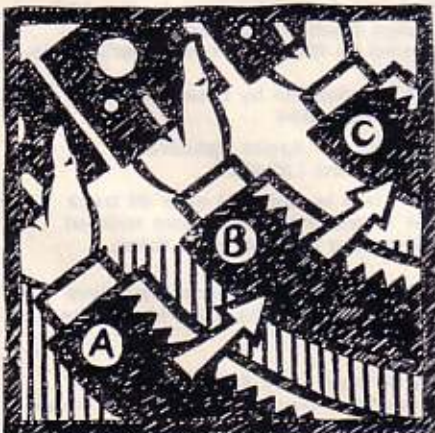
40 Track Drive Systems

Disk User is supplied on a 40 track disk so will work on any 40 track BBC micro system (at least, any that we know of!) straight away. Remember to make a working copy before use.

40/80 Switchable Drives

If you have this sort of drive, you can use *Disk User* straight away with the drive switched to the 40 track setting; don't forget to make a copy for normal use. However, you may wish to copy the disk on to 80 track format, in which case, with a single drive, you should follow the instructions for 80 track systems.

With two switchable drives, or one switchable drive set to 40 track and an 80 track drive (or even a 40 track drive and an 80 track drive), you can easily copy *Disk User* on to 80 tracks; put *Disk User* into drive 0 (40 tracks)



and a blank formatted 80 track disk into drive 1 (80 tracks) and type:

COPY 0 1 <RETURN>

Here <RETURN> means hitting the return key. You can set the boot option to drive one by typing:

***DRIVE 1** <RETURN>

***OPT 4 3** <RETURN>

80 Track Drives

Because *Disk User* is supplied as a 40 track disk, 80 track disk drives have to double-step through the disk. Probably the most convenient thing to do is to copy *Disk User* on to 80 track format. This can be done in two ways.

If your filing system allows double-stepping, we recommend using the system's own command. As a general rule, built-in 40-to-80 track converters should be used where available; the documentation for your filing system or utility ROM will give full instructions, and we give suggestions for some better-known systems further on).

Not all filing systems have facilities for double-stepping; Acorn's DFS is one such system. To overcome this, a program called CHANGE is supplied on the *Disk User* disk in a section which can be accessed by 80 track drives.

Using CHANGE

Insert *Disk User* into an 80-track drive (or 40/80 switched to 80-track) and type:

***CHANGE** <RETURN>

The program will prompt you to insert a pre-formatted blank 80 track disk when it is ready to write to it (you will have to swap back and forward between the two disks several times if you are using only one drive). Once this is completed, you can use the newly created 80-track version of *Disk User* and keep the original as the back-up.

Our suggestions on how to use *Disk User* on some popular DFSs now follow.

Master 128

This Acorn DFS has a software double stepping mode for a 80 track drive. Set it with the command

***DRIVE 0 40** <RETURN>

and then hit <BREAK>

Disk User will then work without any need for conversion. However this may not allow writing to the disk in 40 track mode; in any case, you should make a working copy, so copy to a 80 track disk.

DFS on Master Compact

The DFS is supplied as an image on some versions of the Master Compact Welcome disk (or is available from Acorn on disk) and this may be used in conjunction with a 5¼ inch 40 track disk drive to run *Disk User*. Please note that we **cannot** at present supply *Disk User* on a 3½ inch disk (if there is sufficient demand, we may be able to in the future).



Opus DDOS/Challenger 3

If you are using the Opus DDOS disk filing system or Challenger 1.0/DDOS then issue the command

***4080 AUTO** <RETURN>

or

***ENABLE 40/80** <RETURN>

and *Disk User* will work without any need for conversion.

Challenger 3

If you have the later ROM version Challenger 1.1 then issue the command

***OPT 8,1** <RETURN>

to achieve the same result. *Disk User* will work effectively from the RAM disk. Use

***COPY 0 4 ***

***CONFIG 4=0**

*OPT 4 3

to run from RAM disk

Solidisk DFS

With the Solidisk DFS 2.1 and 2.0 you can set a software double stepping mode for a 80 track drive with the command

***ENABLE 80 <RETURN>**

Disk User will then work without any need for conversion.

Watford DFS

The Watford DFSs also have a software double stepping mode for an 80 track drive. Consult your manual for the appropriate FX call or command. Disk User will then work without any need for conversion.

Disk failure

If for any reason your copy of Disk User will not work on your system then please carefully re-read the instructions given above.

If you still experience problems then:

1. If you are a subscriber, return it to:

Disk User, Infonet, Times House, 179 The Marlowes, Hemel Hempstead;

2. If you bought it from a newsagents,

return it to **Disk User**

Replacements, Discopy Labs, 20

Osyth Close, Brack Mills,

Northampton NN4 0DY before 1st

January 1988. You can make

telephone enquiries about *Disk User*

on 01 437 0626 (please ask for *Disk*

User Editorial). Enquiries in writing

will be dealt with as promptly as

possible and replacement disks sent

out immediately. Please use

appropriate packaging, cardboard

stiffener at least, when returning a

disk. Do not send back your copy of

the magazine. Only the disk please.



Disk Hints

You can speed up your disk access, speeds by issuing the software command: ***FX255,15 <RETURN>** The effect lasts until **<CTRL><BREAK>** is pressed. This extra speed is only possible with the latest disk drives - the half-height and slimline drive types.

Disk User files

All change - 40 track to 80 track convertor.

files:

CHANGE machine code file. *RUN

Disk User - Disk magazine title page animation.

author: Abbas files:

P. RUNDISC BASIC file A. DISC

machine code file

Menu - Easy selection of the software.

author: Matthew Fifield files:

MENU BASIC file

Disk User Music - The new loading theme.

author: Ian Waugh files:

LOADER BASIC file THEME data file



Save Your Bacon! - Simply addictive chase game.

author: C. Lawson & B. Wakefield files:

Bacon BASIC file Bacon1 BASIC file

Bacon2 machine code file Bacon3

machine code file

Ravenskull Map - Helpful colour map on-screen.

author: See Floppy Fun files:

MAP BASIC file LITTLE data file

CASTLE data file

NOTE! this program does not operate correctly on the Master 128

Disk Overlays - Tutorial on putting your disk files to good use.

author: J.Garland files:

OVERLAY BASIC file 1 BASIC file 2

BASIC file 3 BASIC file 4 BASIC file

END BASIC file

Find and Replace - Switch strings quickly and simply.

author: J. Kelk files:

S.F and R BASIC/Assembler source file

Animation - The letter F is motivated by Abbas.

author: Abbas files:

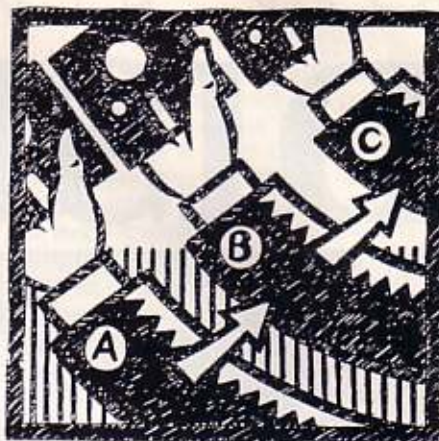
F.ALPHA machine code file P.RUNf

BASIC file

Character Editor - Change your character set.

author: Michael Spalter files:

CHEDIT BASIC file



Spock's Auto Menu - The logical solution.

author: Tom Weiss files:

SPOCK BASIC file

Magic Wall - Deceptively simple, a mind boggling puzzle.

author: D. F. Catlin files:

WALL BASIC file

Arkanoid Cheat - Help for batty people.

author: See Floppy Fun files:

LIST1 BASIC file

Thrust Cheat - Select no. of lives, level, etc.

author: See Floppy Fun files:

LIST3 BASIC file

Disk Functions Library - Essential collection of routines.

author: Dov S. Rosner files:

B.library BASIC file

Quickstep Demo - Advert for Graphics Construction Set.

author: F. Munkert files:

SAMPL2 data file SAMPL2x machine code file

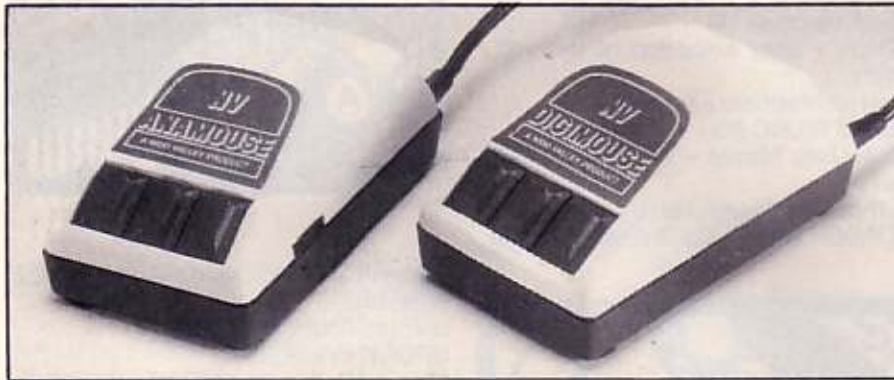
Note:- To exit from the demonstration just press the **SHIFT** key. Details on how to purchase the Graphics Construction Set are in the **SOFTWARE SERVICES** at the back of the mag.

ADFS Users

All files on this disk except SPOCK (a DFS only menu) and LIST3 work perfectly with ADFS. The other two make use of the *DRIVE and *DISC commands, which may be available to your machine on an ADFS utility package. *DRIVE can, in some cases, be replaced with a *DIR: <drive> command.

Note:- Disk User almost fills a 40 track disk. Any software that may need extra disk space to save information must be copied onto a blank disk. i.e. S.FandR.

DISK NEWS



Nidd Valley digital art

Nidd Valley have produced a disk-based art package called *The Illustrator*. A versatile package – Nidd Valley call it a "graphics development program", as it produces some very impressive drawings, illustrations and even technical diagrams.

Being disk based, there is no need to open your Beeb up and fit a ROM inside, and it works with their own Nidd Valley *Digimouse*, which is completely compatible with the vast range of Mouse driven software from Nidd Valley and other Beeb software producers such as Watford and AMX.

The *Illustrator* includes an extensive library of designs, plus the

facility to design and save your own icons, patterns, fonts and brush styles, and being window driven, it is a doddle to use as well!

Where *The Illustrator* really sets itself apart from the competition is with the quality of the dumps available – many printer dump programs look squashed, and are often very faint – *The Illustrator's* integral printer dump package creates high resolution printouts in a choice of 4 sizes with two densities.

● **Contact Nidd Valley using their Freepost address at, Wetherby, West Yorkshire, LS23 7YP. Telephone (0937) 844661 for more details.**

Talking Turtle

Logotron have recently produced an audio cassette tape called "Logo Comes of Age" for Educational Policy Makers, LEA Computer Advisers and Advisory teachers. It attempts to answer the many questions regarding this tried and tested school's language in a cleverly laid out and flashy audio presentation.

Logotron have always taken the view that educational issues take precedence over technological ones, and their tape approaches the thorny subject of educational computing from a strictly educational standpoint.

It has a lot to say for itself, and is surprisingly impartial about the advice it gives – considering they are probably the biggest suppliers of the Logo language and its associated items to the world of education.

● **Contact Logotron at Dales Brewery, Gwydir Street, Cambridge, CB1 2LJ. Telephone (0223) 323656 for more information.**

Arty Reading

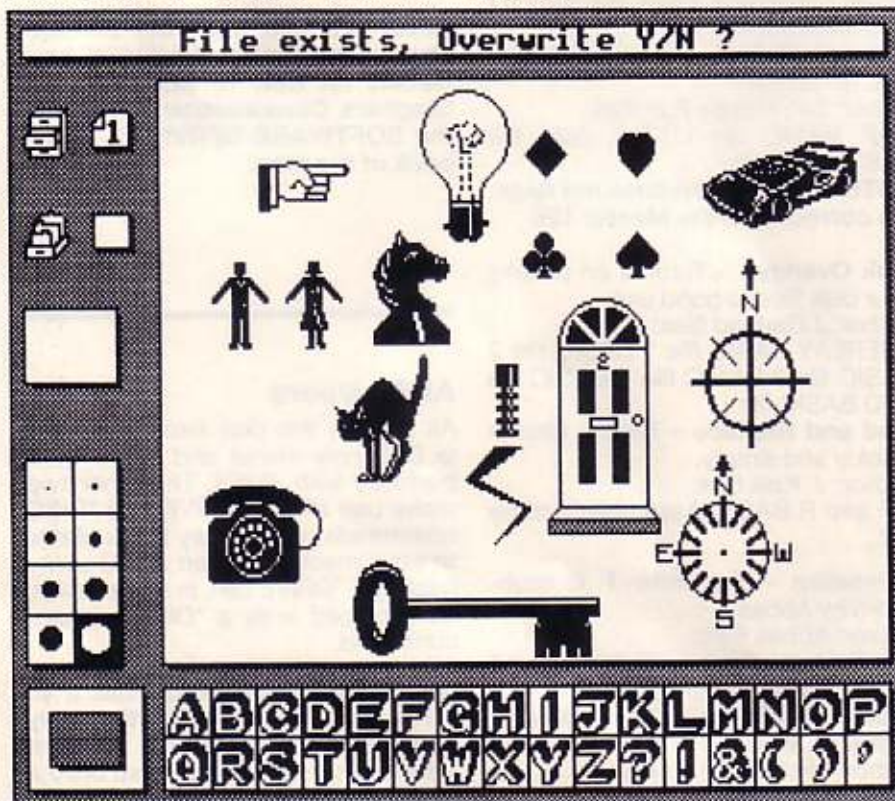
A new book called *Microcomputer Graphics*, by Prof. Michael Batty, has been released by Chapman and Hall books – Providing a readable, and enjoyable introduction to the art and science of creating pictures using the BBC Micro – although it isn't a Beeb specific book, we hasten to add.

Professor Batty skilfully combines information on technical aspects and programming principles with an "appropriate" emphasis on art and design.

As soon as "technique" is mastered, it is elaborated into a program from which the user can produce a piece of art (all be it a picture or a design) that is both well-structured and visually appealing.

Prof. Batty goes into great detail about Fractals, Computer Landscapes and Animation. There are full BBC BASIC listings given in the book as well! All in all, the book weighs in at a hefty 344 pages, and for £14.95 for the paperback version, doesn't seem too expensive either.

● **Contact: Chapman and Hall at 11 New Fetter Lane, London, EC4P 4EE.**





In Two Minds ...

Zaphod software announced the availability of a utility package for users who need to transfer existing BBC Micro files to an Archie.

Called "Beeb to Ark", this package links a BBC B with DFS to an Archimedes by a supplied data cable and a disc based program, it provides all the necessary menu driven software to transfer the files.

● **Contact Zaphod Software, Floor 1, 17 Howe Street, Edinburgh, EH3 6TE. Telephone (031) 556 4056 for more details.**

BAKing up

BAKsoft have contacted us about the DISK utility program supplied with the Acorn 512 upgrade to format disks to IBM PC standard. In the expectation that users would be able to use these disks on IBM PC's or compatibles, often such disks fail to work properly – some users thinking that the problem lies behind

the fact that the Master 512 uses 80 track disk drives, whilst the IBM PC uses 40 track drives.

The truth of the matter is that the 360K Disk format suffers a "major design defect" – in that there is insufficient space allowed after the index hole and before the first sector of the disk. The DISK program writes less than 50 bytes, whilst most PC's expect 150 or more.

BAKsoft's utility, DOScopy, formats disks on 80 track drives which are almost always readable on IBM PC 40 track drives; The utility also copies files between BBC and PC formats with full wildcard facilities, and is much faster than Acorn's GETFILE and PUTFILE routines.

● **Contact BAKsoft at 20 Leys Avenue, Cambridge, CB4 2AW.**



BBC to Archimedes conversion?

Tubelink, the popular BBC Micro database on PRESTEL, have released a version of Archimedes BBC BASIC for the BBC Micro.

Available as either a disk or a ROM, Advanced BASIC attempts to fully emulate the Archimedes' implementation of BBC BASIC which has "friendly" extensions such as CASE declarations, extended colour commands, circles, ellipses, advanced editing options, the "install" command, the "Mouse" commands (not forgetting the famous WHILE ENDWHILE structure).

Advanced BASIC costs £29.95 and is available from Tubelink, P.O. Box 641, London, NW9 8TF. Telephone 01 205 9393.

Scrambled Chips

A new utility ROM is available from **Number 3 Software**. Called *The Scrambler*, this ROM will help owners of BBC Micros and BBC Masters to comply with the security requirements of the Data Protection Act. The ROM costs £9.95.

The Scrambler is menu driven, and encodes files (these can be BASIC programs, text files, database records and so on...) Each ROM has its own password and scrambles the data so that no one can have a look at what is on your disk without them knowing the password first.

The program is intelligent enough to recognise BASIC files from text files and so on, so the decoding process is completely automatic.

● **Contact Number 3 software at 3 Dairy Farm Court, Attleborough, Norfolk, NR17 2BT. Telephone Attleborough 454902.**



Watford make the Best of both worlds

Times are changing fast for the floppy disk, and whilst the current standard for the BBC B and Master Series is the 5" inch disk, the future standard for future machines will most certainly be the smaller; faster and (possibly) more reliable 3½ inch disks – now a standard feature of both the Master Compact and the Archimedes Series.

To support the past and cope with the future, Watford Electronics have designed their own plinth-mounted disk system – called the DP35 800S, it contains both the 5½ inch and the 3½ inch drive formats as well as offer a firm and rigid support for a monitor on top. With the BBC Micro underneath, the plinth makes a surprisingly small footprint on the desk giving you more space to work on.

The 5" inch disk drive is 40/80 track switchable, and it has its very own power supply running independently of the BBC Micro or Master.

Watford's new plinth enables users to read, write and format disks in DFS and ADFS format – currently the accepted standard for the Master Compact and the Acorn Archimedes, it also enables the user to read disks written for the Archimedes, or vice versa.

The DP35 800S costs £175 and is available from Watford Electronics.

● **Contact Watford Electronics, Jessa House, 250 Lower High Street, Watford, Herts. Telephone (0923) 37774 for more details.**

DISCUSSION



Compact problem

I am writing to you in desperation hoping you can shed some light on our problem. We bought a BBC Master Compact for our children and, becoming increasingly frustrated at the lack of software on 3.5" disk, we decided to invest in a 5.25" drive. We purchased a Cumana switchable 40/80 and appropriate lead. We were given a Master 128 DFS because our Welcome disk did not contain the DFS image.

We bought a Model B educational disk which refused to run. We then bought Disk User but the only way I can load it is to do a "DRIVE 1 and "EXEC !BOOT. And that's as far as we got, a 'disc error 18' appearing on screen.

Would it make any difference if the DFS was the proper one for the Compact?

P.Hunt Oxfordshire

Some interesting points here. The Master 1770 DFS with which you were supplied should work correctly with the Compact. The educational disk you bought did not work on your system, nor would it on any 1770 DFS on any BBC, because of the disk protection routines used on it. There are still a good many disks being sold from back catalogues which have this type of self-defeating protection (designed for the old 8271 type disk interfaces). Naturally in your case it constitutes a disk which doesn't work and you can

return it for a replacement or, failing that, a refund.

The second drive unit is just that to begin with, treated as drives 1 and 3 by the Disk Filing System. However you can reconfigure the drives using the DIP switches, both on the 3.5" drive inside the Compact and the 5.25" external drive. The company who supplied the hardware will be able to advise you. In this way the external drive becomes drive 0 and SHIFT/BREAK will boot the Disk User menu.

Disk error 18 usually indicates that the disc format and drive format are incompatible. Disk User comes on a 40 track disk so make sure your'e drive switch is over the right way.



Graphics mystery

I have found out why the Abbas animations do not work on my machine. I have expanded my BBC B computer with a Watford ROM/RAM board. This caused overheating problems in the Signetics 6502 CPU. I therefore replaced this chip with a Rockwell 65C02 (3Mhz) CMOS CPU. Your graphics software doesn't run properly with this CPU installed. If I insert my old 6502 CPU then the graphics run perfectly.

J.Crouchley Nottingham

Very interesting. This is the first time we've come across this problem. Disk User isn't designed to be Tube compatible so any 65C02 processor

may cause problems. However many of Disk User's programs will run perfectly well across the Tube so it's worth switching on and giving them a try.



Back 'em up

Well done! Each issue of Disk User is better than the last. I particularly liked the Rescue utility, a straightforward program which enables deleted programs to be recovered, providing the instructions in the article are carefully followed.

Fortunately I did read the article before loading and running the program. If I hadn't I probably would have wiped the disk. At least you had the forethought not to include the program on the menu.

Perhaps I am being unfair, but I would suggest that you make such programs start with a brief introduction and/or warning.

B.M.Phillips Cumbria

Point taken. We felt that leaving the program off the menu was enough to deter readers from running the program without consulting the instructions. And we were... *wrong* in a number of instances! Many of the

most useful disk utilities we publish are potentially harmful to data on the disk so we must repeat again that your first action after taking Disk User out of the plastic wrapping should be to **back up** the disk. This goes for any valuable software you have. And keep the original in a safe place, only ever using the copy.

MAGIC WALL

Meet the Magic Wall puzzle challenge and win a year's subscription to your favourite magazine!



In the next issue of Disk User, issue 6, April/May out on the 18th March 1988, we will publish the solution to the Magic Wall. If, before the closing date of 25th January 1988, you have solved this *Rubik Wall* then send it in to the Disk User office. The first correct entry will receive a subscription to Disk User for the following six issues.

Send your entry to:
Magic Wall Disk User Number One
Golden Square London W1R 3AB

The following rundown of the program may allow you to understand how the code works, but it might not help your attempt to crack the puzzle.

Line 20 This line is the auto-start on <ESCAPE> and should be omitted on first entering the program. **30** SH(n) – used for initial set up and as memory store when scrolling wall. **40-60** Mode 2 – copy keys off, cursor off. **70** This line calls the menu from which the rest of the game happens. **100-280** PROCPLAY – the main play routine. **120-170** Sets up screen. **180-200** The

play loop which 'drops out' when CHK=0 ie the WALL is complete. **210-250** Reward for finishing the wall. **260-270** Program hold till keypress when wall completed. **290-500** PROCINIT – initialising routine that draws main wall and side panels and writes messages at top and bottom of the screen. **490** X=340 – required for PROCPOINT. **510-590** PROCINPUT – Input routine. **530** Flush keyboard buffer. **540** Get keypress. **550-570** Inc/decr X and move pointers. **580** Scroll colours using PROCfill and check for finish (PROCcheck). **600-810** PROCshape – which is the PROCdraw steerage routine. N is used to select which set of parameters are passed to PROCdraw for each brick in the wall. **620** Brick colour is set using SH(N). **820-920** PROCdraw – brick drawing routine. A,B,C are passed to the routine which draws a brick on row A, which starts at position B units across and is C-B units wide. The colour of the brick is set before the routine is entered. **930-1050** PROCfill – Colour scrolling routine.

950-960 Samples are taken for each brick colour and stored in SH(1)-SH(6), 'background' colours are stored in SH(7)-SH(12). **970-1040** Colours are scrolled using PLOT 77 and PLOT 73 commands. Colour information comes from SH(1)-SH(6) and SH(7)-SH(12). **1060-1150** PROCPOINT – draws arrows at top and bottom of the screen centered on X. **1160-1250** PROCscramble – wall shuffling routine. Each colour (1 to 6) is offered up three times and inserted into random 'empty' bricks. **1260** PROCcheck – finished wall check routine. 36 separate samples are taken (6 per row) to check for correct colour/row. Routine is abandoned on first incorrect colour. CHK=0 if wall is correct, else CHK=1. **1350** PROCmenu – displays front page, and waits for 'P' or 'B' input. **1500-1730** PROCfanfare – plays terrible tune when wall is completed. **1740-1840** PROCbegin – routine for beginners, allowing then to play on a finished wall. **1770** Sets up 'perfect' wall data in SH(n). **1830** Repeats until 'S' is pressed.

NEXT MONTH

NEXT ISSUE

The next issue of **Disk User** is the sixth, completing **Disk User's** first full year providing programs for **BBC Micro** users. In the **APRIL/MAY** issue of **Disk User** you can expect more of the best software mix in computer magazines. Celebrate with **Disk User APRIL/MAY**, out on the **18TH MARCH**. Reserve your copy now! **DISK USER SIX WILL FEATURE:**

Magazine Publishing Awards **Disk User** – finalist in the 1987

Disk User for the **BBC Micro** was one of the stars at the 1987 Magazine Publishing Awards. The trailblazing **Disk User** was the only representative of the computer magazine industry amongst the big guns of publishing.

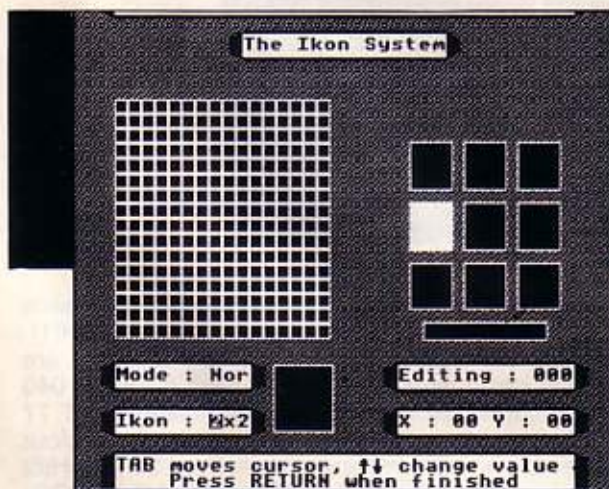
The Publisher magazine each year chooses a distinguished panel of industry experts to judge nominated magazines in 17 different categories. **Disk User** for the **BBC Micro** was nominated in the category of most successful spinoff magazine because it was launched out of sister magazine **A&B Computing**.

Although only two issues of **Disk User** had been published at the time of judging, the magazine was

runner up in the category. At the Magazine Publishing Awards luncheon on November 11th, guest presenter Sue Arnold of the Observer, handed over the finalist's certificate to **Disk User** editor Mark Webb.

The judges praised the speed at which the launch was accomplished and were impressed by the "sheer value of the cover-mounted disk that formed part of the new publication's concept".

Disk User would like to thank readers for their support and pledges to bring the **Disk User** concept continuously up to date and into line with what readers ask for from us.



Iconophile.

An icon orientated control system. You can use this system to "professionalise" any software, providing a highly attractive visual "front end" for non-technical users. Machine code provides high performance but clever design makes the icon system as easy to use as the graphic environments it helps to create.

Two screens are better than one.

Instantly switch between two screen windows, carrying out independent operations in each, eg cataloguing a disk and listing a program side by side.



Larger than life.

Zooming in on some of the more interesting screens around, and giving you the tools to do it yourself.



Anywhere, Anytime.

A hotkey utility for screendumps, disk handling, indeed any machine operation. Hops out of any program and back again!

Psychebrot.

Careware from CODIL Language Systems. Stunning special effects on magnificent mandelbrot graphics. PLUS a chance to donate help to the mental health charity MIND. Find out how you can help in the next issue of **Disk User**.

PLUS: ● animations ● games ● utilities
● graphics ● cheats!



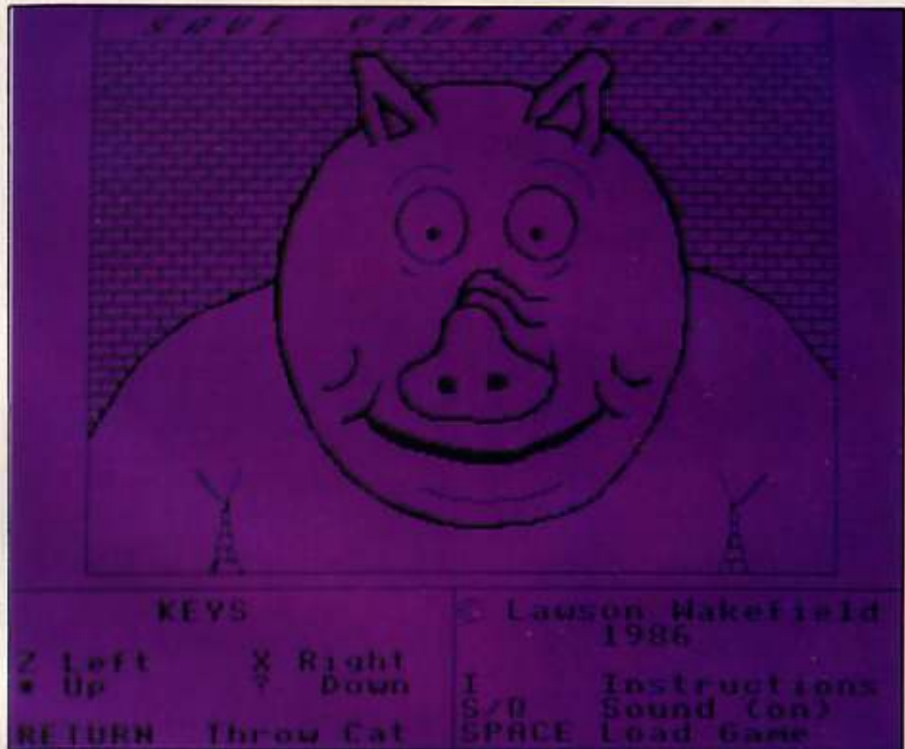
SAVE YOUR BACON

'Save Your Bacon' – an original arcade-style game with an off-beat story line



In *Save Your Bacon* you control a pig who must save his bacon literally. The pig must run around the houses and collect the pork chops which move randomly round the screen. He is in competition with a dog who will eat the pork chops given half a chance. Later a second dog appears who is more inclined to chase the pig, but who is also not averse to eating the odd pork chop. The pig must collect ten chops to clear a screen.

If life gets too hectic, the pig has several friendly cats at his disposal which can be thrown at the dogs. If a cat hits a dog, a big bonus is added to the score and the dog is held up for several seconds in a fight with the cat. Whilst this fight goes on the pig can dash round collecting extra chops and can even run straight through the dog who is of course otherwise occupied. The pig loses a life if the dogs eat too many pork chops or he is caught by one of the dogs.



Program documentation

The program is written mainly in BASIC, but uses a machine code sprites routine for improved presentation and speed. The usual features such as sound effects, jingles, high score, sound on/off, extra lives, selectable start screen, on-screen instructions, loader screen, etc are all included. It can be run on either tape or disc and is split into four parts. It was written on a 32K BBC Model B with Acorn DNFS and BASIC I, but the program runs without modification on the BBC B+, Acorn Electron and Master 128. The program will not run on a second processor/Turbo board or with shadow RAM activated.



File: Bacon

This is the loader/instruction program. User characters and sound envelopes are defined in this program. A cartoon picture of a pig is loaded into memory above the program, but below the Mode 4 screen memory. The picture is POKE'd into the screen memory and the game keys are displayed. The user is offered the option of switching sound on/off, seeing the instructions or loading the main game. If instructions are required, these are displayed on a separate Mode 4 screen and the Pig screen is redisplayed afterwards.

PROC chars defines characters for House, Kennel, Sty and copyright symbol

PROC env defines envelopes.

PROC intro prints instructions in white on blue background

PROC keys prints keys and draws frame on main loading screen

PROC pig POKE's cartoon picture of pig (*LOADed at start of the program) into top two-thirds of screen

PROC print formatted string printing routine, used to print instructions

PROC space prompts user to press SPACE and waits for it

PROC window1 sets window to bottom third of screen and clears it

PROC window2 sets window to bottom right corner of screen and clears it

File: 1.Bacon

Mode 4 screen file, containing a cartoon picture of a Pig drawn using AMX Art. This file is *LOADed by 'Bacon' to &3900.

File: 2.Bacon

Main game program. First *RUNs 3.Bacon, which is the sprite handling machine code. The main loop checks the keys and controls the movement of the pig. If no key is pressed, the pig will carry on in the last direction selected. The meat and dogs are updated next. If the cat is on-screen it is updated, otherwise the RETURN key is checked for throwing the cat. The main loop terminates when the pig has collected enough chops or a life has been lost. If the screen has been successfully completed the next screen is initialised. A new life is given if the score has crossed a multiple of 10000 during the last screen. A game ends when all lives are lost.

A% - Z% reserved for use by sprite code

bon% remaining bonus for current screen

(Starts at 1000 on screens 1 to 5)

(Starts at 1500 on screens 6 to 9)

(Starts at 2000 on screens 10 and later)

col% colour of border and dog kennels for current screen

(Screen 1 = red, 2 = green, 3 = yellow, ... going back to red every six screens)

ct% TRUE if there is a cat currently on screen, FALSE otherwise

del% length of time a dog is delayed by a fight with the cat. Calculated as $600 + 40 * (\text{screen number})$

d% dead flag. TRUE if a life has been lost

du% note duration for tunes/jingles

eat% amount of bonus lost when a dog eats a pork chop

ex% number of extra lives given so far

g% number of pork chops that the pig must get to clear a screen

hi% hi-score

h% sprite number of dog currently 'hit' by the cat

=0 - neither dog hit by a cat

=4 - dog allocated to sprite 4 hit by cat

=5 - dog allocated to sprite 5 hit by cat

htim% time when a dog was hit by a cat

i% general purpose temporary variable

lv% number of lives remaining (initially three but extra life every 10000 points)

m% number of pork chops collected by pig so far

nc% number of cats left (initially three, but extra cat on completion of every screen)

pi% note pitch for tunes/jingles

s1% skill factor controlling the 'intelligence' of the dogs. The lower this value the more likely the dogs are to home in on their target. The dog allocated to sprite number 5 homes in on the pork chop whilst the dog allocated to sprite 4 (which does not appear until screen 6) homes in on the pig. The value of s1 varies between 19 and 12 depending on which stage of the game has been reached

s2% skill factor controlling the variability in the movement of the pork chop. The lower this value, the more the chop keeps changing direction. The value of s2 varies between 15 and 12 depending on which stage of the game has been reached.

scn% screen number. Lowest screen number is 1

sc% score

sw% sprite currently being displayed in the cat/dog fight. Alternates between cat (6) and dog (4 or 5)

t% TRUE if this is a 2-dog screen, FALSE otherwise. Second dog appears on 6th screen

tim% start time in delay procedure
PROCwt

x% general purpose sprite x-coordinate variable

x2% number of sprite positions by which cat moves in the x-direction (twice that of other sprites)

xs% number of sprite positions by which the pig, dogs and pork chops move in the x-direction. Increases during the game.

x graphics x-coordinate used in drawing screen

y% general purpose sprite y-coordinate variable

y2% number of sprite positions by which cat moves in the y-direction (twice that of other sprites)

ys% number of sprite positions by which the pig, dogs and pork chops move in the y-direction. Increases during the game

y graphics y-coordinate used in drawing screen

z% LOCAL direction variable used by **FNc2**

z1% direction of movement of pig (sprite 1)

z4% direction of movement of 2nd dog (sprite 4)

z5% direction of movement of 1st dog (sprite 5)

z6% direction of movement of cat (sprite 6)

z7% direction of movement of pork chop (sprite 7)

(Note: directions are 1=Right, 2=Down, 3=Left, 4=Up for dog/pig/chop)

(5= Right 6= Down 7= Left 8= Up for the cat)

FN c1 TRUE if moving a dog or pig in direction defined by **Z%** is allowable. FALSE otherwise

FN c2 TRUE if moving a cat in direction **z%** from position (x%,y%) is allowable. FALSE otherwise

FN d1 returns direction number, for

first dog, which leads most directly towards the pork chop

FN d2 returns direction number, for second dog, which leads most directly towards the pig

FN x used to determine whether a sprite is in a vertical 'corridor' or not

FN y used to determine whether a sprite is in a horizontal 'corridor' or not

PROC bon reduces bonus by i% and prints remaining bonus, in green

PROC cat updates the position of the cat. If either of the dogs have been hit by the cat, a 2000 point bonus is added to the score and h% is set to the sprite number of the dog which has been hit

PROC cats prints number of cats remaining, in white

PROC chop prints number of chops collected by pig, in cyan

PROC del if previous direction of pig does not match new direction, delete current pig sprite

PROC dg move one or both of the dogs. If on a one-dog screen, only move sprite number 5. If a dog is involved in a fight with the cat don't check that sprite. Change direction if dog is in a 'corridor' intersection and a random value is above preset skill threshold. Check for collisions with the pig and with the chop

PROC eat a dog has eaten a chop, so make an appropriate noise, delete the chop sprite, initialise the position of a new chop, reduce the bonus and check if the bonus has reached zero

PROC end prints game-over message. If it's a high-score say so and play a tune

PROC exlife give an extra life. Update count of extra lives given. Tell the user they've got an extra life and update screen display of number of lives

PROC fgt animate the dog/cat fight. Alternately print either the cat or dog and make the appropriate noise. If the time limit for the fight is up, delete the cat and clear the cat flag

PROC hitune 'High score' tune

PROC inisp initialise the machine code sprite routines and set the initial positions for the pig/dog/chop sprites

PROC kill lose a life. Sets d% (dead flag) to TRUE, reduces lives by one, plays appropriate jingle, and spins the pig round on the spot

PROC mt move pork chop. If the chop collides with the pig, increment number of chops collected, delete chop, initialise a new chop, add 50 to the score

PROC newm select a random start direction and (x,y) position for a new pork chop

PROC next screen complete, get ready for next screen. Add remaining bonus to the score, increment and display the number of cats, play a tune and increment the screen number

PROC pgs prints number of remaining lives in magenta

PROC screen draws a new screen and initialises counters/flags

PROC scr increments score by i% and prints score in yellow

PROC skl sets skill level for current screen

PROC start start of a new game. Prompts for SPACE TO START. If keys 1 to 9 are pressed set start screen to corresponding number. If SPACE pressed, start from screen number 1. Initialise number of extra lives (0), score (0), number of cats (3) and lives (3)

PROC thr throw a cat, if allowed. If there are no cats left or the pig is not facing along a 'corridor', then the cat cannot be thrown. If cat can be thrown, set direction (z6%) to same as that of the pig, set cat flag (ct%) to TRUE, set cat sprite co-ordinate to the same as that of the pig, print the cat on the screen and update screen display of number cats left

PROC tune 'Screen complete' tune

PROC wt waits for del% hundredths of a second.

File: 3.Bacon

Sprite handling machine code.

Beginners start here... how discs are made

Disk Drives at V

Dr Solomon Axlegrease

A floppy disk starts life as a 26" wide, six mile long roll of polyester film. The roll is coated with a film of magnetic oxide – exactly the same coating as the layers on a magnetic tape found on video tape or cassette tape.

The coating is sprayed onto the surface of the roll to a depth of "exactly" 100 micro inches on both sides; this is possibly the most exacting of all processes in the manufacture of a disc. If there is an uneven surface due to too much coating being sprayed on, the chances are that disk drive heads will clog up and even wear out your drive heads would

This is why the coating process is undergone in a laboratory with clean machines sealed from the harmful vestiges of the outside world; some companies have perfected ways of coating the disk surface so that the dispersal of magnetic particles are

random (thus making the chances of disk errors less likely)

The roll of film is then stretched out and holes are literally "punched" into the roll turning the sheet of film into disks. With the extra "holes" in the middle, the disk is electronically inserted into a plastic envelope where it is either glued together or (as is the case with modern disks) it is heat welded together making sure that the edges are rigid and tight.

The Jacket

The disk jacket is very important. Because data is recorded *onto* the surface of a disk, the disk must be kept very clean so that grit and dust particles present in the air don't get caught in the jacket and "gouge" out a chunk.

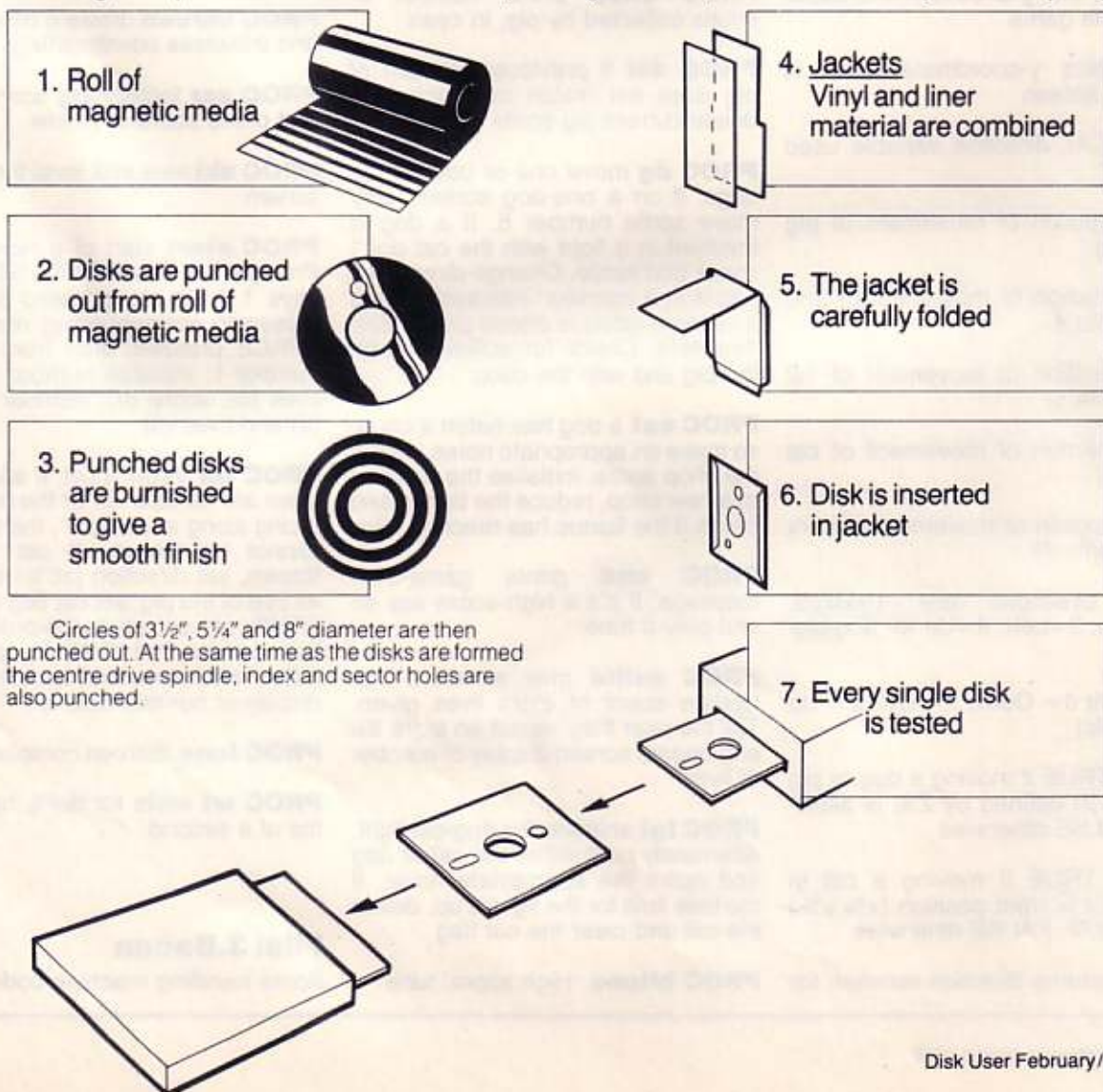
To combat this (and to stop them wearing out at the same time from casual wear!), the jackets are insulated with a padding material; this

also cleans the disks and makes them rotate in their sleeves easily thus ensuring their long life – and the combination of a modern disc drive and a modern disc means that a disc should last a lifetime!

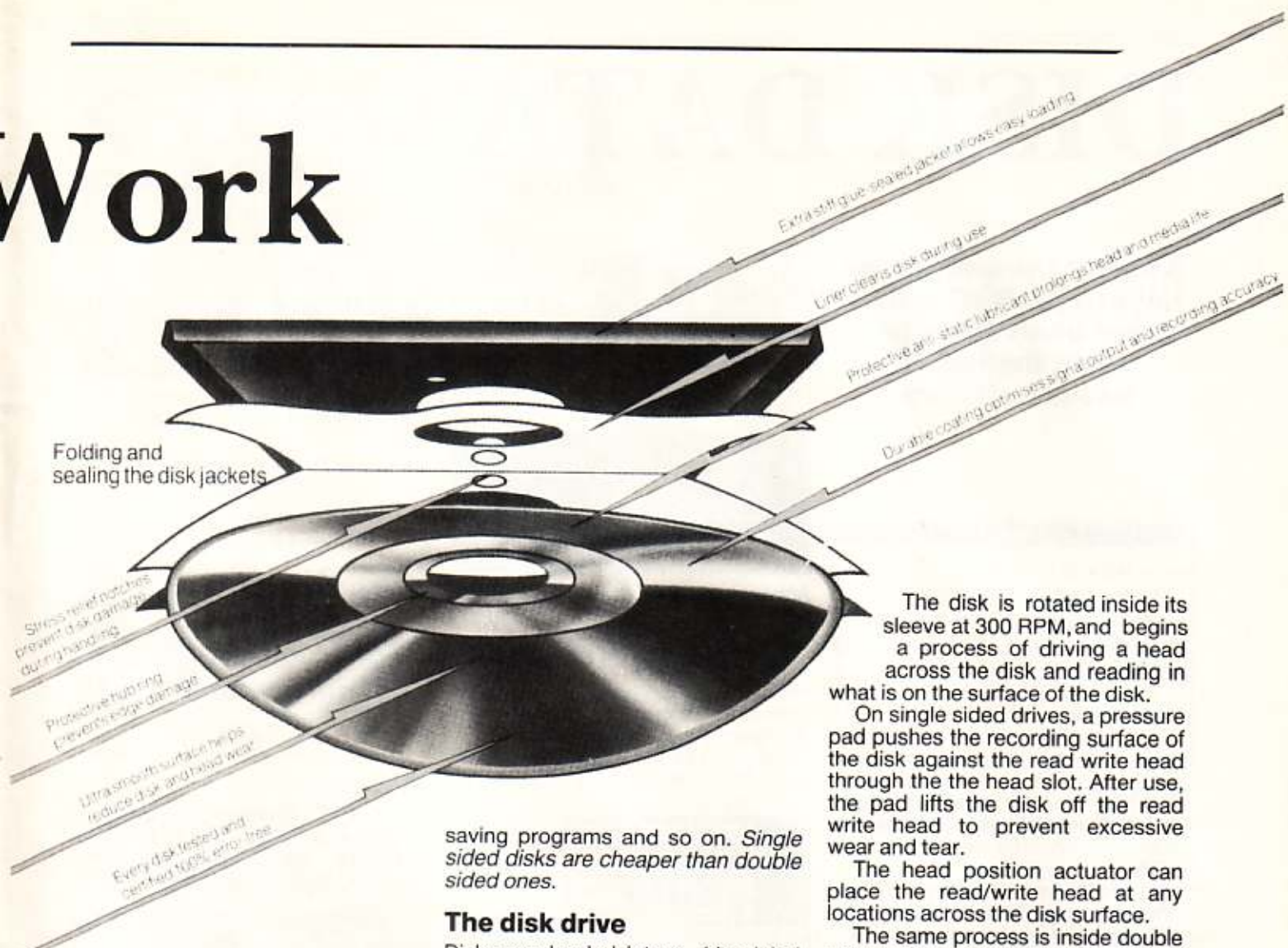
That's not all folks

After a floppy disk is made, it is rigorously tested. This isn't just advertising blurb trying to boost floppy sales; *each disk is rigorously tested* searching for a variety of possible faults, uneven coatings, tight sleeves, disk surface smoothness(!) and many other tests are undertaken to ensure that disks do not wear out – many disks these days carry a lifetime guarantee or at least a no-quibble replacement offer.

Double sided disks are exactly the same as single sided disks – it's just that only one side is guaranteed by the manufacturers to work! This means that you can buy single sided



Work



The disk is rotated inside its sleeve at 300 RPM, and begins a process of driving a head across the disk and reading in what is on the surface of the disk.

On single sided drives, a pressure pad pushes the recording surface of the disk against the read/write head through the head slot. After use, the pad lifts the disk off the read/write head to prevent excessive wear and tear.

The head position actuator can place the read/write head at any locations across the disk surface.


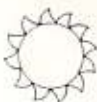










The same process is inside double sided disk drives which have two read/write heads, one for each side of the disk.

saving programs and so on. *Single sided disks are cheaper than double sided ones.*

The disk drive

Disks are loaded into a drive label side up, the flap is pulled down and then the magic begins.

Disks for your double sided disk drive and be reasonably assured of

 <p>Never touch the flexible disk surface. It is susceptible to damage and even fingerprints can cause data to be lost.</p>	 <p>Store flexible disks away from direct sunlight or heaters. Temperature extremes, especially heat, can damage disks. Also avoid sudden temperature changes (more than 20°C in an hour).</p>	 <p>Insert carefully by grasping the upper edge and placing it into the disk drive. Never force the disk into the drive. Call your authorised service engineer to check the machine if something is wrong.</p>	 <p>Keep disks in their envelope and store disks not for immediate use in their specially designed containers standing them upright.</p>
 <p>Do not bend or fold flexible disks. This will render disks inoperable.</p>	 <p>Put identification labels in a suitable position. Never use them in layers.</p>	 <p>Do not place heavy objects on a disk.</p>	 <p>Do not clip, staple or rubber-band flexible disks together.</p>
 <p>Do not use solutions like alcohol, thinners and Freon to clean the disk. The special liner cleans the disk surface continuously. No other cleaning is necessary. Dirty or damaged disks should be discarded as placing them in a disk drive can contaminate the read/write head.</p>	 <p>Do not write on a flexible disk with anything but a soft, felt-tip pen. Ideally, write labels separately, then place them on the disk jacket. Erasers can cause damage just as easily as ballpoint pens or pencils. Make a new label if you need to change reference information.</p>	 <p>Magnets and magnetised materials should never come close to a flexible disk. Be particularly careful with magnetised paperweights, motors and power cables. Data can be lost from a disk exposed to a magnetic field.</p>	 <p>Keep the area of operation as clean as possible. Occasional static electricity on disks can attract contaminants. Plastic carpet protectors increase the risk of such static and should not be used.</p>

DISK DATA

Some of the backroom boys in the BBC world have been filling us in on their disk based software

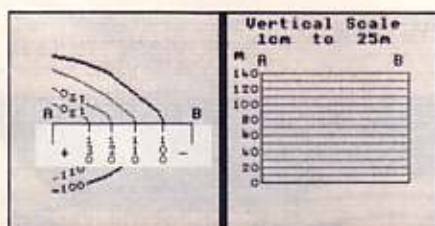
Horse Number 2

Enter the following Data :-

1. Horses Name : DREVFUS.....
 2. Jockey Rating : 7.
 3. Weight (Stones) : 11.
 (Pounds) : ..
 4. Latest Finishing Position : 2
 5. Number of 1st's 2nd's & 3rd's : 2.
 6. Number of Races taken from : 4.

Result for DREVFUS : 23.8365647

Okay (Y/N) : YES



Making a Cross Section Example 1
 3: On a piece of graph paper draw a horizontal line of the same length as the cross section and a vertical scale as above.
 Press Space Bar

Maps for 8 to 14

Maps and Landscapes 2 from Chalksoft is excellent educational software for the 8 to 14 age group learning about geography. The manual and copiable worksheets are designed for classroom use.
 Chalksoft Ltd PO Box 49 Spalding Lincs PE11 1NZ ☎ 0775 69518



PENTOMS - PUZZLE

- 1 > Pentoms Instructions.
- 2 > Kidtoms :(Pentoms Primer).
- 3 > Pentoms :(4 grids).

Please type the number beside the program you want.

Whenever you want to return to this menu, please press the BREAK key.

Press ESCAPE to abort the title page.

Pentup puzzle

Pentominoes from Tortoissoft has had a chequered career but is now marketed by its author Simon Anthony.
 Tortoissoft 60 Lynton Avenue West Ealing London W13 0EB

J. EVERYBODY - V.V. CARL Jun 88
 Three month period to end Aug 88
 SALES

C/Amount	E			
AMPLX 001	123.44	16	10	107.24
MENBY 002	431.94	36	04	375.60
GUMBY 003	513.43	67	04	448.20
JONES 004	41.40	67	40	36.00
CLARK 005	61.18	23	98	33.20
HOPES 006	569.21	24	43	428.06
LEWIS 007	2650.98	343	78	2305.20
BORAX 008	519.86	72	00	448.66
ALTED 009	1800.86	234	89	1565.88
MORAY 010	69.23	8	90	59.33
GARRAGE 011	74.06	23	00	55.58
STAINV 012	1789.33	23	00	1552.33
COUNCIL 014	39.80	2	80	32.00
XVDIST 015	346.45	45	19	301.26
S/Ttl	E 9205.10	1200	67	8004.42

Page 1 M month selector E info

Top tips

Tipster from Chestnut Software comes on an appropriately green turf disk!

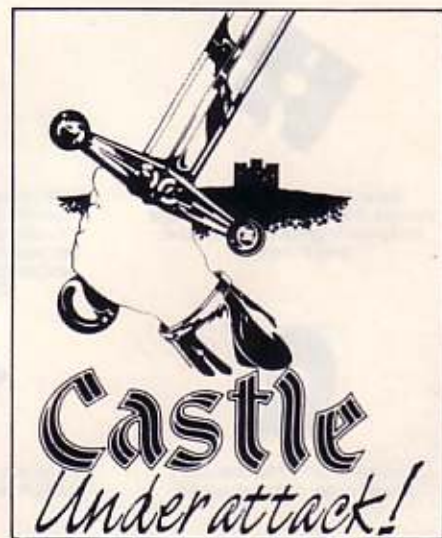
You have to type in your own data from the newspapers but bringing a more scientific approach to your betting must be a good move, surely?
 Chestnut Software 37 Chestnut Drive Bexleyheath Kent DA7 4EW

Know thyself

Life-Line from William Stuart Systems is a self-analysis counsellor for help with relationships, self-image and career. Designed for use "anywhere that people matter".
 William Stuart Systems Quarley Down House Cholderton Nr Salisbury Wiltshire SP4 0DZ
 ☎ 098 064 235

Care ware

Cash, VAT, Building Society and Investment Care from Squirrel Software are similar programs designed to help you keep track of the cash.
 Squirrel Software 4 Bindloss Avenue Eccles Manchester M30 0DV
 ☎ 061 789 4120



Castle Keep

Castle Attack from Fernleaf is a magnificent strategy game full of exciting educational content. It's also very professionally presented and easy to play.
 Fernleaf Educational Software Fernleaf House 31 Old Road West Gravesend Kent DA11 0LH
 ☎ 0474 359037

REALISE YOUR FULL POTENTIAL

LIFE - LINE

SELF ANALYSIS
 CAREER COUNSELLOR

COLLECTOR'S ITEMS



Abbas

The latest in the sequence of highly collectable animations from computer artist Abbas.

NAME THAT THEME

Name that tune and win a prize

Music plays an ever more important part in computer entertainment so it was inevitable that Disk User would get its own theme tune sooner or later. Ian Waugh, well-known musician and author on all subjects computer-musical, was commissioned to write a ditty which would set you humming. And that's exactly what he's done.

The only thing the tune doesn't have is a name. So any suggestions are welcome. And indeed the best suggestion will win a set of 10 Disk User data disks and the name will be adopted for future issues.



DISK DATA

Bonecruncher

This is Superior Software's best game this month, a Repton-meets-Ravenskull type adventure with some repetitive music, but saved by some nifty graphics.

The idea is original: You are a dragon manufacturing soap (from – of all things Bones) for customers who enjoy bathing in the sea surrounding your castle.

Bones litter your castle, and you have to brave all the creepy crawlies and monsters to retrieve them, boil them down into bars of soap (!) and then dump them into the sea for your customers.

The game has a few original ideas, eg. the ball shaped monsters who can be moved by tipping soap in the sea! and your friend and business associate called Fozzy who is so thick that he can only walk along one side of a wall without falling over!

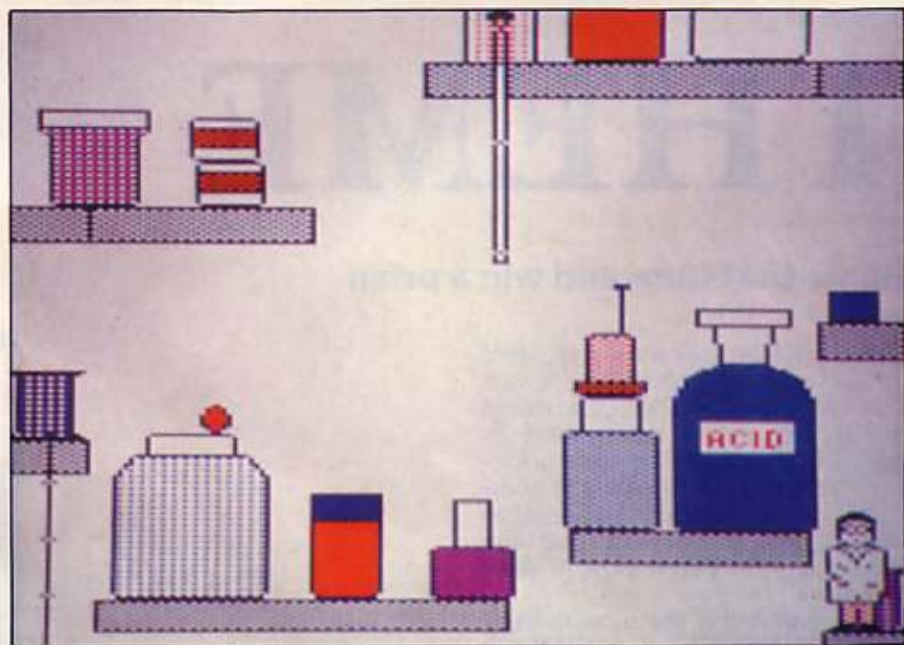
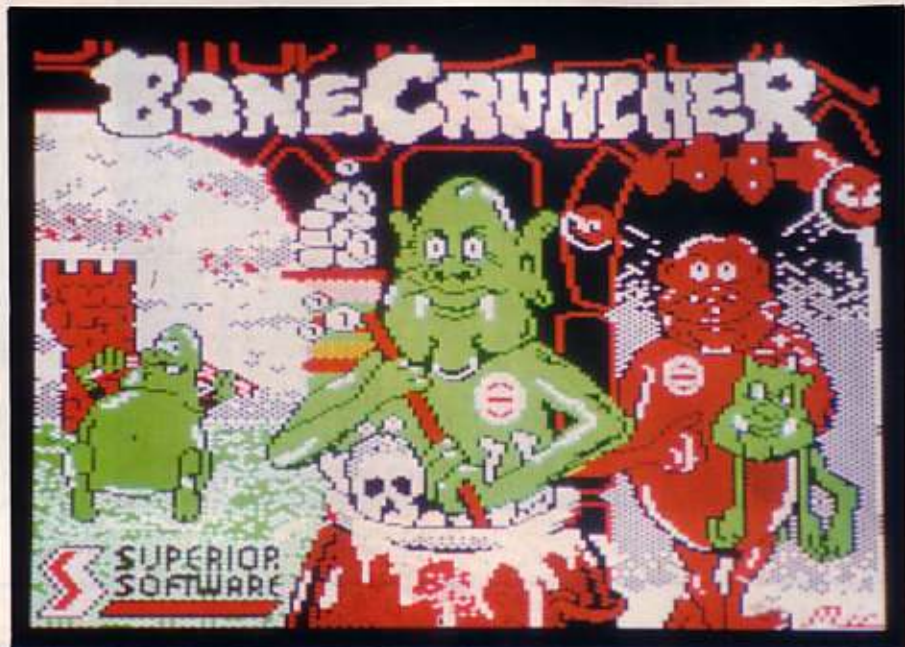
This game is very very addictive, I didn't like it to begin with as it seemed to be just another Repton clone, but it is fast turning into a "think it out" game with some witty ideas, loads of features and plenty of action.

This is a good game – not much fun if you like fast action shoot em ups, but then Bone Cruncher isn't like that – buy it.

Spellbinder

It was the dark ages, it was darker than anyone had ever expected, in fact it was so dark that people kept disappearing all over the place. Dark looking magicians fought dark monsters of the night (which was pretty dark) instead of day – a dark twilight fell over the land (which was pretty dark to begin with, but now it's *really* dark).

In these dark times, there lived in the (reasonably light-ish) land of Lorraine ten shadowy mage lords. One such mage lord (who was darker than the rest) had decided to do some very black things; It is, as usual, up to you to enter his dark



DISK DATA



castle and find all the potions to make a really potent spell and banish him from the world thus (hopefully) making the world a better place to live in where bunny rabbits can hop around nibbling at fresh, dewy blades of grass, where silly actors don't become presidents and streams, filled with bright golden fish, flow past peaceful meadows, where people can all be nice to each other and live happily ever after.

There's no doubt about Spellbinder being a fun game to play, it's just that the graphics aren't up to much – strongly resembling the 3D graphics of *Knight Lore* and *Alien-8*, in that all of the screens are only in two colours!

If you have a TV screen instead of a monitor, then you will have some difficulty seeing what the attacking characters are.

Spellbinder is not a bad game – but perhaps it would be best to wait until it appears on a compilation.

Elixir

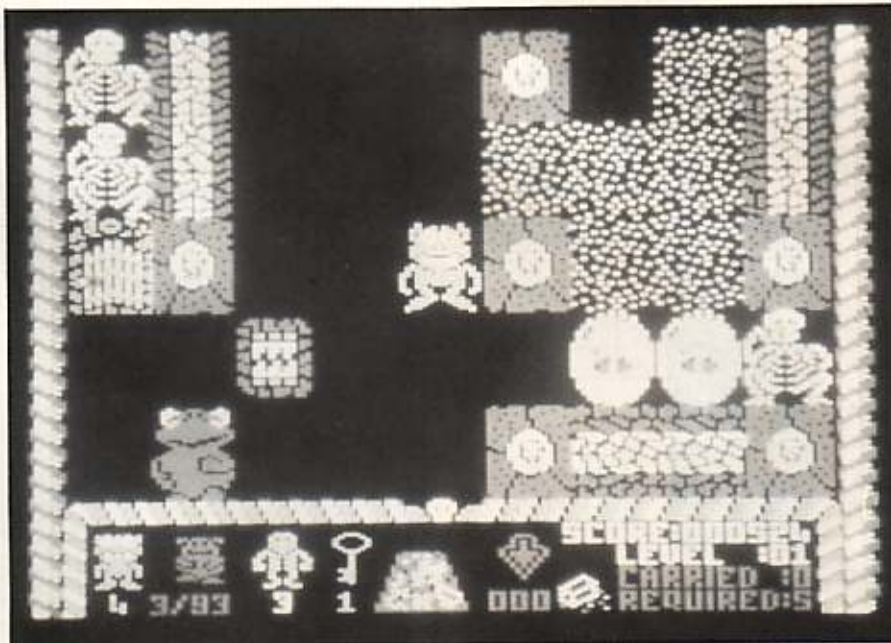
Do you ever wonder why medicine tastes so bad? I suppose that if anyone knows, then Cyril – the hero of this Superior software game – should.

The idea is simple – You have swallowed some potent liquid by accident, and you are reduced to the size of a test tube.

Unlike *The incredible shrinking man*, you can get bigger, but you have to swallow 40 pills in order to neutralize the effects of the potion.

The game is colourful (with some effective dithering effects to obtain shades), but there is no sideways scrolling (as you would expect) – nor are there any reasonable sound effects to redeem this silly game. The story line is decidedly duff, and whilst the programming is clearly of a high standard (ultra fast plotting everywhere), the idea behind the game is extremely silly for a full price game.

Again, this would be a nice compilation game – but it doesn't stand up on its own.



TRANSFER

Taking your disk files one step further

Disk User programs can be so useful that you'll often want to transfer them to their own disks and use them separately, without title page or menu. To do this successfully you'll need to learn a little about BBC BASIC and the DFS (Disk Filing System). In Disk User we don't believe in referring you to the manuals to here's an explanation of how such a transfer can be achieved.

Let's take the disk function library as an example this month. The relevant file on Disk User is:

B.LIBRARY

Make sure you have a blank data disk ready to receive the file. Insert Disk User and type *COPY 0 0 B.LIBRARY and press the RETURN key. Follow the keypress prompts on the screen until the > prompt returns.

On completion place your new disk in the drive. Type *BUILD !BOOT

upon which the disk drive will whirr and a number will appear on screen. Type

LOAD "B.LIBRARY"

and press the RETURN key. Wait for a second number to appear and then press the ESCAPE key. Let the drive finish any activity and then type

*OPT 4,3

and press the RETURN key. Once again wait for any drive activity to cease. Now you can press the SHIFT and BREAK keys together, releasing the BREAK key first, then the SHIFT key. This will boot your new disk into action, automatically executing the BASIC command LOAD "B.LIBRARY". The program B.LIBRARY is now loaded into memory ready for use. You can test this by typing LIST. For now, press

the CTRL and BREAK keys together.

You can now type *TITLE LIBRARY and press the RETURN key. Now type *CAT

and press the RETURN key and the screen will display the title LIBRARY with the files !BOOT and B.LIBRARY lined up below. You may now use this as your program utility disk, saving other utility files (such as the F&R utility) in the space available (space is something we don't leave on Disk User itself).

The process described above can be repeated for any of the files which are detailed at the end of Disk Instructions each issue. A useful new disk you could create would contain the software database from issue two, another could be a complete utilities disk with this month's toolkit routines, our disk recover utilities from issue two and disk examiner from issue one, all handy and on a single disk ready to use. You can even modify our simple menu program to provide an easy way to access the programs. List the menu and you will find DATA statements at the end. We'll explain how to modify these for your own use next issue.

OGOTRON...SPECIAL OFFER...XOR SOLUTION ...LOGOTRON...SPECIAL OFFER...XOR

Disk User readers who are fans of the Logotron game XOR will appreciate this special offer:

- the XOR designer at a special discount price and
- a XOR solution disk EXCLUSIVELY available to Disk User readers

Designer and solutions

disk: price £11.00

Solutions disk only: price £3.00

You will need the XOR designer to run the solutions so, if you don't have the designer, you will need to order the designer and solutions disk combination. If you already have the designer then all you have to do is send off for the solutions disk.

Enclose a cheque made payable to Logotron Ltd for the required amount and send your order to Disk User Offer, Logotron, Dales Brewery, Gwydir Street, Cambridge, CB1 2LJ. ☎ 0223 323656 for credit card orders.



XOR
DESIGNER

The XOR Designer is the ideal way to create your own XOR solutions. It's a powerful tool that lets you design your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them. The XOR Designer is a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.

The XOR Designer provides you with a powerful tool that lets you create your own XOR solutions. You can create your own XOR solutions and save them to disk. You can also load existing XOR solutions and edit them.



XOR

DESIGNER

BBC B, B+ MASTER, SERIES 31/2, DISC 114.95

LOGOTRON

DESIGNER

SPOCK DISK MENU

Spock-like logic is behind this easy to use disk menu system.

When I initially bought a disk drive, one of the first things I noticed was that I had several disks with many multi-part programs on them. I decided I needed some way of distinguishing between the programs and the header files, therefore I put all the header files into a different directory. I discovered that other menus would only read the first 31 files on my 62 file Watford disks. Therefore I set about writing my own menu.

I wrote four versions of this Menu. Version 4 I showed to R.Casula, an Electron user, and together we perfected it, and converted it to read both sides of the disk simultaneously.

I used this for a long time, but I discovered that there were several files that would not load. Therefore I rewrote the loading routine, and devised *Spock logic* to distinguish between BASIC, machine code, text files and sideways RAM. This was implemented in my new menu: *Spock's Menu v1.0*, and so far I have not found a single file which it will not load. (Please treat this as a challenge - Ed.)

These were the requirements I had, and I have fulfilled them in version 1.1:

- It works on all disk filing systems
- It reads up to 128 files at once (the maximum displayed is 80)
- The files to be loaded go in only one directory
- Any file can be loaded by it
- It will download all files which occupy disk workspace
- It is very user friendly

Using the menu is very, very easy. Once it has been loaded it will read all the files in the "M" directory off the disk, which will take a few seconds. (This can be changed by altering the letter "M" in lines 140 and 150 to the required directory.) Then the files will be displayed on the screen. The file highlighted is the one to be loaded and this can be changed by using the cursor keys. <RETURN> loads the file, <SPACE> catalogues a new disk, and <TAB> toggles the sound.

How the program works

The only parts I shall go through in detail are the assembler and file reading routines since the rest of the program is fairly straightforward and written in BASIC.

All access to filenames is done by calling OSGBP(B&FFD1). On entry A% holds the command for which the values are as follows:

- A=1 Put bytes using offset
- A=2 Put bytes
- A=3 Get bytes using offset
- A=4 Get bytes
- A=5 Read disk information
- A=6 Read current drive/directory
- A=7 Read library drive/directory
- A=8 Read file from current directory

On entry X% and Y% point to a command block which is constructed as follows:

- XY Channel (when file handling)
- XY+1 Pointer to data (when file handling) or result data XY+5
- Number of bytes to transfer (when file handling) XY+9
- Byte offset

(optional when file handling)

The commands we are interested in are OSGBP(B 5 (OSGBP(B with A=5) to read the disk title and OSGBP(B 8 to read the filenames. The result data from OSGBP(B 5 is set out as follows (numbers indicate offset from pointer to result data):

&00 &0C (length of title) +&01 Disc title with zeroes making up to 12 characters +&0D Shift-break option +&0E Drive number

Line 150: The program reads this data into memory at S% and then counts down from +&0C until a non zero is reached, and then inserts a carriage return at the last zero (ie the end of the title), and reads the title for the appropriate drive from +&01.

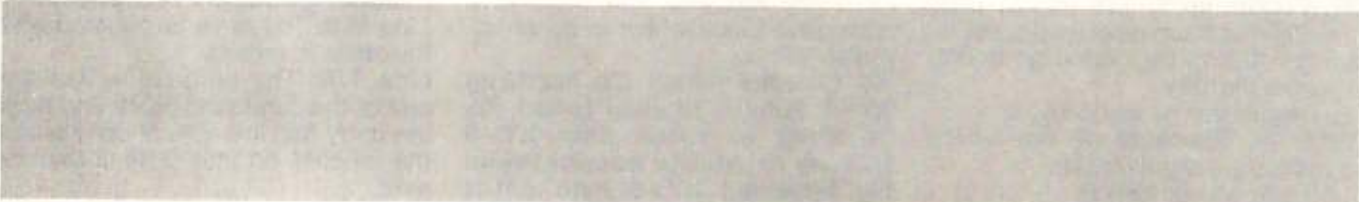
OSGBP(B 8 is similar to OSGBP(B 5 except that the offset is used to read the file names in order. The offset starts off at 0 and is incremented by seven each time OSGBP(B is called. The other difference is that a value is returned. From the result, all registers must be removed except for the carry flag. This is done by ANDing the result with &1000000. If the carry flag is set, then the last filename has been read. The result data is set out as follows:

&00 &07 (the length of each filename) +&01 Filename +spaces to make length up to 7 characters.

Line 160: The drive is selected and the offset is zeroed.

Line 170: The program writes the rest of the command block and puts the carry flag into E%, if carry is set the process on that drive is terminated.

2002



Line 190: A carriage return is inserted at the end of the filename, the filename is put into the array file\$ and the drive number is put into T%.

The reading of the length, load, and execution address of a file is done by calling OSFILE(&FFDD). As with OSGBPB A% is used to signify the command, which with OSFILE are:

- A=0 Save section of memory as a file
- A=1 Write catalogue information for a file
- A=2 Write load address for a file
- A=3 Write execution address for a file
- A=4 Lock/Unlock file
- A=5 Read catalogue information for a file
- A=6 Delete a file
- A=7 Load a file

As with OSGBPB X% and Y% point to a command block, with OSFILE 5 the block is as follows:

&00 Points to filename

On return from OSFILE 5 the file type is given, this can be one of three things:

0 Nothing found

1 File found

2 Directory found

Also the command block is rearranged as follows:

&00 Unchanged +&02 Four byte load address +&06 Four byte execution address +&0A Four byte length

Line 220: The computer sets up the command block, calls OSFILE and puts the result in R%. Then len,exec & load are read from the altered command block, and FNinfo then returns the value of R% MOD 2, so that if the directory is found then the value returned is 0, for file not found.

Processing information

That's all the information reading routines done, but now there's processing the information.

All the processing is done in the PROCEDURE called *spock*, and hence is very logical!

To be able to load any file, you have to know the characteristics of each file. The hardest is BASIC. BASIC files have an execution address of &801F, &8023 or &802B, depending on the version of BASIC the were saved from, therefore it should always be very easy to find a BASIC file since it should have one of those execution addresses. Unfortunately this does not always work, since many files which are

downloaded have incorrect execution addresses. Therefore the only way to discover if a file is BASIC is to look at the first few bytes of it. This is done using OPENUP (or OPENIN on BASIC I). The first byte in a BASIC file is always &0D. So any programs which begins with &0D are BASIC. Unfortunately some machine code programs have a blank BASIC program at the beginning. Therefore we must check to make sure that &FF, the terminating byte of a BASIC program, is not the next byte.

The next type of file I thought about was sideways ROM images; these usually load and execute between &8000 and &BFFF. Therefore it is easy enough to distinguish a sideways ROM, but unfortunately I did not have enough memory to write a sideways RAM loader.

Now, any file which loads below &8000, having failed the above two rules, must be machine code. After that, any other file is probably text. If that isn't logical then what is?

Machine code

There are three machine code routines in the program. The first one, found in PROCassemble, is an all purpose downloader, and I'm sure most people with a basic understanding of 6502 assembler will be able to work out what it does. Here is a short explanation:

The program is a downloader. The address at &72 is where the data to be downloaded begins. Stored at &74 is the address of where the data is to be downloaded to, and at &70 is the amount of data to be downloaded.

The second routine, in PROCmcode, loads in the required file, sets up and calls the downloader, then jumps to the execution address.

The last section, in PROCbasic, does virtually the same as the second two, except instead of jumping to the execution address, it stores the correct page value (The high byte of load) in &18, which is where the variable PAGE is stored, and inserts RUN into the keyboard buffer.

Additions

As the program stands it will load all BASIC, machine code, and text files that I have come across. You may like to add other routines, such as picture loading, sideways RAM loading, a routine to print out all the files on the disk, or even a routine to load in programs in other languages. Happy programming!

Variables and procedures

A\$ Equivalent of file\$ in FNinfo. **QS** title\$ in red text. **T%** An array containing the drive of each file. **at** A label used in PROCassemble. **block** The command block used in FNinfo. **downl** A label used in PROCassemble. **drive%** The loop variable to read both drives, in PROCcat. **end%** Number of the file to be loaded. **exec** The execution address of the file to be loaded. **finish** A label used in PROCassemble. **file\$** An array to store the filenames in. **file%** Original end%, used in PROCselect. **file\$** The file to be loaded. **file** The channel allocated to file\$ in PROCspock. **i%** Loop variable used in PROClist. **k%** ASCII code of keypress in PROCkeys. **In** A label used in PROCassemble. **loop** A label used in PROCassemble. **loop1** A label used in PROCassemble. **len** The length of the file to be loaded. **load** The load address of the file to be loaded. **ld** A label used in PROCmcode and PROCbasic. **name** The block where the filename is stored in FNinfo. **of%** A variable used in PROCselect to prevent flicker. **pass** The option loop variable in PROCassemble, mcode and basic. **q\$** The equivalent of QS in PROCcen. **sound%** The status of OSWORD 210 sound on/off. **to** A label used in PROCassemble. **title\$** An array holding the title for both sides of the disk. **title\$** The title produced by FNtitle. **y%** A variable used in PROCcen. **z%** The status of file\$ as read by FNinfo. **PROCassemble** Assembles the downloading code. **PROCinit** Initialises all arrays. **PROCcat** Reads the catalogue off the disk. **PROCoscli** Equivalent of OSCLI in BASIC 1. **PROCpage** Prints the title on the screen and makes it look pretty. **PROCcen** Used to centralise the title on the screen. **PROClist** Lists the files on the screen. **PROCkeys** Checks for <TAB> and <SPACE>. **PROCselect** Checks for the cursor keys. **PROCspock** Analyses the file and calls the correct loading routine. **PROCbasic** Loads a BASIC file. **PROCmcode** Loads a machine code file. **All resident integer variables are not listed since most of them are used very frequently and it would be impossible to document them satisfactorily.**

OVERLAY TECHNIQUES

Overlays get over memory restrictions and allow you to build libraries of often used procedures

Overlay is designed so that overlays of other BASIC programs can be run from your main program. The overlay programs are loaded into the computer memory and then automatically removed when not needed.

Thus it is now possible to run programs that are the same size of the computer disk, not just the same size as the memory. Also, as you can see from the program, it hardly takes up any room (the main overlay program is 315 bytes long).

Program description

Line 10 This line is important in that it sets the size of the overlay you wish to have (as the variable L%).

Therefore alter this variable to to the maximum size of overlay you will be using. To find out the size of your files type in

*INFO * and hit the RETURN key

The size of your files will be given in hexadecimal. You can print a hexadecimal number in decimal by typing:

```
PRINT &hexnumber
```

Note the use of the & character in front of the number.

If the overlay is too big an error trap in the program will report this (see line 90).

Line 20 This line records the TOP of the original program and T% so that at the end of the run the overlays can be wiped out to leave only the program. This is done in the procedure PROCDELETEEND.

Also LOMEM is increased by the amount of room allocated to the maximum size of overlay. This space is only for the program and not for the variables. Also, because the program needs to alter LOMEM, no variable should be declared before

this change is made – unless the variable is in the range of A% to Z% (the integer variables). This is because the variable names would be stored just above LOMEM, an action which would corrupt memory in the space we reserved for the overlays.

Line 30 This reserves space for the *command which loads in the overlays at the last position of the program

Line 40 This is how you call up the overlays. The overlay is a filename so you type in PROC("File name of overlay").

If the filename is not present on the disk the computer will produce an error. This procedure can be anywhere in the program. However an overlay should *not* call up another overlay as this would crash the program.

Line 41 This procedure deletes the last overlay to leave the original program behind. This should be the last line of the program.

Line 60 This is the procedure that deletes the last overlay to leave the original program. It places &0D and &FF characters in the starting value of the TOP variable location (T%).

Line 70 This is the procedure to load and run the overlay. The name of the file to be loaded is taken from the procedure parameter

Line 80 The file to be loaded in is opened up and the size of it is taken by using the command EXT#. The length is stored in N%. If this length is greater than the length allocated (L%), then the computer reports that it is too big. On the press of a key it returns from the procedure.

Line 100 to Line 110 This loads the overlay onto the end of the main program. The program is loaded in by the command *LOAD so it is very quick.

The next line will be that of the overlay. So when producing the overlays make certain that the last command is an ENDPROC. This will then return the program from the procedure as if it was any other procedure. Note there is no difference between this overlay procedure and any other procedure, in which variables that are used or created – eg DIMs – are remembered when the program returns or even uses another overlay.

Overlay programs that are to be used by the main program are easy to write as if they were procedures but there are two important exceptions. The first is that the DEFPROC is not included. This is because the computer still thinks that it is using the procedure PROCAD("FS"). The second is that overlays must not call up other overlays, else the program will completely crash.

So at the end of the overlay leave an ENDPROC and this will return the program from the overlay when it's run.



The main program on disk is called PROGEM and it is full of highlighted REM statements (in Mode 7) to show you what is going on.

I have provided some examples of overlays on the disk which are short programs. However they could be much larger if L% on line 10 is altered. The overlay examples on the disk are file names 1,2,3,4 and END. The program PROG runs these overlays as a demo. Thus from this very simple program you are now not limited by the space of the computer. I hope you enjoy creating your own useful overlay procedures. If you come up with any you might think beneficial to other readers why not send them in. If we publish them on Disk User we'll send you some free software by way of thanks.

SUBSCRIPTIONS SENSATIONS! AT COVER PRICE ONLY!

Any of the magazines from the list below can be delivered to your door
with : NO handling charge NO postage costs

A&B Computing	£18.00
Aeromodeller	£22.20
Citizens Band	£16.20
Clocks	£27.00
Commodore Disk User	£15.00
Disk User	£17.70
Electronics Digest	£11.00
Electronics Today International	£16.80
Ham Radio Today	£15.60
Military Modelling	£15.60
Model Boats	£15.60
Model Engineer	£26.40
Photography	£15.00
Photoplay	£12.00
Popular Crafts	£16.20
Radio Control Model Cars	£16.20
Radio Control Boat Modeller	£7.50
RCM&E	£14.40
Radio Control Scale Aircraft Quarterly	£9.00
Radio Modeller	£14.40
Sea Classic International	£11.70
Scale Models International	£15.00
Video Today	£15.00
Which Video?	£15.00
Woodworker	£15.60
Your Commodore	£14.40
Model Railways	£13.20
Practical Wargamer	£7.80

The Above Prices Refer to U.K. Subscriptions Only

This offer is also open to existing U.K. subscribers who
may wish to extend their current subscriptions.

Please commence my subscription(s) To with the Issue
I enclose my cheque/money order for £ made payable to Argus Specialist Publications Ltd.
or debit £ from my Access/Barclaycard No.
valid from to Signature

Name

Address



Send this form with your remittance to:
INFONET LTD. (N.R.S. 88) 5 River Park Estate,
Berkhamsted, Herts. HP4 1HL.



2 issues FREE

When you take out a subscription to any of these magazines

	EUROPE	MIDDLE EAST	FAR EAST	REST OF THE WORLD
A&B Computing	£27.30	£27.60	£31.10	£28.30
Aeromodeller	£27.00	£27.20	£29.00	£27.50
Citizens Band	£20.10	£20.30	£21.80	£20.50
Clocks	£32.40	£32.60	£34.70	£33.00
Commodore Disk User	£18.00	£18.20	£19.30	£18.40
Disk User	£21.60	£21.70	£23.20	£22.00
Electronics Digest	£12.90	£13.00	£13.70	£13.10
Electronics Today International	£21.00	£21.20	£22.80	£21.50
Ham Radio Today	£20.10	£20.30	£22.00	£20.60
Military Modelling	£22.40	£22.60	£25.20	£23.10
Model Boats	£20.00	£20.10	£21.80	£20.40
Model Engineer	£35.40	£35.70	£39.00	£36.30
Photography	£22.00	£22.30	£25.00	£22.80
Photoplay	£16.70	£16.90	£18.70	£17.20
Popular Crafts	£21.40	£21.60	£23.60	£22.00
Radio Control Model Cars	£21.10	£21.30	£23.10	£21.60
Radio Control Boat Modeller	£10.00	£10.20	£11.10	£10.30
RCM&E	£20.40	£20.60	£22.90	£21.00
Radio Control Scale Aircraft Quarterly	£11.10	£11.20	£12.00	£11.30
Radio Modeller	£20.00	£20.20	£22.40	£20.60
S&A Classic International	£10.70	£10.80	£12.00	£11.00
Scale Models International	£19.60	£19.80	£21.50	£20.10
Video Today	£20.20	£20.40	£22.30	£20.70
Which Video?	£19.40	£19.50	£21.20	£19.80
Woodworker	£22.80	£23.00	£25.70	£23.50
Your Commodore	£22.00	£22.30	£25.20	£22.80
Model Railways	£19.10	£19.30	£21.50	£19.70
Practical Wargamer	£7.10	£7.20	£8.00	£7.40

The Above Offer Applies to Overseas Subscriptions Only

This offer is also open to existing subscribers who may wish to extend their current subscriptions.

Please commence my subscription(s) To with the Issue
 I enclose my cheque/money order for £ made payable to Argus Specialist Publications Ltd.
 or debit £ from my Access/Barclaycard No.

valid from to Signature

Name

Address



Send this form with your remittance to:
**INFONET LTD. (S.O. 88) 5 River Park Estate,
 Berkhamsted, Herts. HP4 1HL.**



INTERCEPTORS

See DU Dec/JAN 88
P 31

Installing BASIC program utilities by intercepting your command call

Last month we published the first two of our *Interceptor* programs, the *screen compression* utility and the *BASIC program resequencer*. Unfortunately we didn't publish full instructions for using the resequencer utility. These now follow, along with instructions for using the third utility in the series, the *find and replace* command.

Resequencer

Note that these instructions apply to the utility to be found on Disk User issue 4. *B.RES* is a resequencer utility enabling the order of lines in BASIC programs to be rapidly altered. As with the other utilities the machine code object file is generated by running the assembler program. The resulting machine code will be saved automatically as *res*. A disk other than your copy of Disk User should be set up to receive this before running "B.RES".

The machine code is assembled from &7000 and so will operate at the same time as F&R, the find and replace utility on this issue's disk.

To install the utility from your utility disk type *res or *RUN res. Once installed, a call to the routine takes the form:

***RES <start line number>/<end line number>/<move to line number>**

Giving the command results in the lines between <start line number> and <end line number> inclusive being removed from their current position in the BASIC target program and inserted immediately after <move to line number>. The target program will be renumbered to suit the new order and any GOTO/RESTORE references changed as needed.

Default values are allowed and bad parameter values are rejected with appropriate error messages.

Examples:

*RES 100/200/750 would move the lines from 100 to 200 inclusive to immediately after line 750.

*RES /200/750 would move the lines from 0 to 200 to after line 750.

*RES 80040 would move the lines from 800 to the end of the program to after line 40.

*RES /500/ would move the lines from 0 to 500 to the end of the program.

*RES 200/500/400 would be rejected as the destination line is within the range of lines to be moved.

How it works

Lines 130 to 170 set up CLIV.

Lines 200 to 1190 carry out the parameter checks. This not only involves converting and checking the given line numbers, but also transforming these into more useful memory addresses by scanning the program line by line for the proper match to the line numbers. In every case the final addresses, if acceptable, are stored in zero page, in order, from the lowest address to the highest. By this means the same algorithm for moving the code may be used no matter what order the *RES parameters were given: ie *RES 100/200/900 is handled in the same way as *RES 1000/2000/20.

At this point the routine enters the address of the next major routine, reentry, into the PAGE 4 addresses corresponding to the BASIC system variable A%. It then uses OSBYTE 138 to insert characters into the keyboard buffer and exits. The characters form the commands REN, followed by CALL A%. So the utility is reentered after the program has been renumbered in steps of ten (1210-1300). This was done to make the calculations for new GOTO/RESTORE tokenised line numbers a simple matter of addition, subtraction or no change depending on whether or not the lines referred to were involved in the resequence.

The remainder of the utility consists of three main routines: lineset, 1660-1880, calculates the constants needed to be added/subtracted to tokenised line numbers as mentioned above.

movecode, 1900-2270, performs the actual movement of the lines. These are moved in 1 page blocks via PAGE 6 with the program lines from the top of this block to the final address involved in the resequence being moved down into the gap thus created. The contents of the buffer may then be inserted into the space made available by this downward movement. This procedure is repeated until all the lines to be moved have been processed.

resetgoto, 2500-2770, scans the target program line by line, renumbering the lines in steps of ten. Tokenised line numbers are detokenised into ordinary numbers. Addition/subtraction is performed if required and the result is retokenised and returned to the proper position in the target program.

Find and Replace

B.F&R, is the find and replace utility on this issue's disk. Any piece of text in a BASIC target program may be located and displayed or replaced by this routine. As it makes use of two areas of the BASIC ROM (the table of keywords and the tokenisation of text routine) this utility will only run on the version of BASIC on which it is created. *B.F&R* automatically sets up the code for your BASIC (1, 2 or 4) and saves it to your own utility disk as *F&R*. If you wish at some later date to use the utility on a machine with one of the other versions of BASIC the Disk User source code will have to be run again on a machine with the alternative BASIC installed. Once again remember to replace Disk User with your own disk before running the source program so that the machine code generated may be saved.

Since F&R is likely to be used more than once in a program debugging session function key 0 is set up with the command *F&R to save time and typing.

The search may be global, *F&R, or selective eg *F&R 100/200.

After giving the *F&R command with any parameters and pressing RETURN you are prompted for text to be searched for. This text may be of any length up to the maximum allowed in a line of BASIC and may contain keywords.

You are then prompted for a replacement string. Pressing RETURN immediately will simply carry out a search for all occurrences of the first string and display these lines. Otherwise any text entered will replace the first text wherever it is found and the altered lines will be displayed. To increase readability any multi-statement lines are displayed one statement to a line on screen.

Further information on the operation of this utility can be found in Disk User issue 4.

SERVICES

Software

9 Hall Road, Marylands Wood Estate, Hemel Hempstead, Herts HP2 7BH 0442 211882

CHECK WITH THE TABLES TO SEE IF OUR SOFTWARE IS AVAILABLE FOR YOUR COMPUTER. PLEASE INDICATE YOUR REQUIRED PRODUCT NAME, NUMBER AND TRACK FORMAT IN THE ORDER FORM

INFORMATION

Software is only available on the formats recorded in the tables. It is not available on cassette tape. All prices include VAT and postage and packing. Please make cheques/postal orders payable to Argus Specialist Publications Ltd and print your name and address clearly on the back of your cheque. Telephone orders are taken for Visa and Access. All software is sent first class. Prices include postage and packing for European destinations. Please include £1.00 postage and packing (airmail) for other overseas orders. Please allow 28 days for delivery.

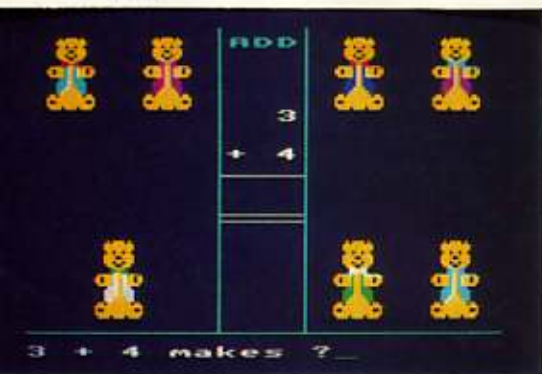
In Disk User Three:

Cholo Player's Guide - and disk editor **Auto Save** - 'Safety first' filing **Quick Break** - Personalised default messages **Tazman** - fast moving game **Dual catalogue creator Feeling the Squeeze** - reduce the size of your BASIC programs **Count with Teddy** - make maths fun for kids with these graphical games **Quick Copy** - fast memory copy for Model B **Calculator** - icon driven scientific calculator Utilities reviewed Archimedes and disks Collectors Items

Back issues of Disk User cost £2.95 and £0.50 postage and packing. Disk User back issues and subscriptions are sent in special packaging to protect your software.

Send your remittance to: INFONET LTD, 5 River Park Estate, Berkhamsted, Herts HP4 1HL.

Tel: 04427 76661-4 for VISA and ACCESS credit card orders.



Ordering

All the disk based products in our tables are available from Disk User Reader Services. Just fill in the order form with the appropriate product number and track format.

Disk User One Order product number DU01

In Disk User One:

Collectors Items - animation A of the collection **Barmy Butterflies** - a platform game with a built-in screen designer **Cube of Zoth** - science fiction adventure complete with a scenario which should leave you in no doubt as to the pleasure of beating Zoth at his own game! **Auto Cataloguer** - powerful utility which automatically catalogues your disk collection **Disk Examiner** - convenient and efficient way of checking the contents of your disks **Karate Kid** - arcade game Winchester Way Tape Transference

Order product number DU1. Price £2.95 and £0.50 postage and packing.



AVAILABILITY AND PRICE

Name	Product	40(small)	60(large)	Price					
double disk					Statistics	DB41	yes	yes	£10.00
Global View	DB01	yes	yes	£10.00	ADFS	DB42	yes	yes	£10.00
Electron +3	DE06	no	yes	£12.00	Compact 3 1/2	DB43	no	yes	£12.00
ADFS	DB16	no	yes	£10.00	double disk				
Compact 3 1/2	DB33	no	yes	£12.00	Games Compendium 1	DB45	no	yes	£10.00
Graphics Pack 1	DB03	yes	yes	£10.00	Electron +3	DE45	no	yes	£12.00
Electron +3	DE01	no	yes	£12.00	Home Office	DB46	yes	yes	£6.00
ADFS	DB17	yes	yes	£10.00	double disk				
Compact 3 1/2	DB37	no	yes	£12.00	Graphics Construction Set	DB49	yes	yes	£10.00
Ikon Utilities	DB19	yes	yes	£6.00	ADFS	DB50	no	yes	£10.00
Masician	DB06	yes	yes	£6.00	Compact 3 1/2	DB51	no	yes	£12.00
double disk					double disk				
Venturescapes	DB48	yes	yes	£10.00	Compact 100	DB52	no	yes	£15.00
double disk					Bibliography Upgrade	DB55	yes	yes	£5.00
Educational Compendium	DB47	yes	yes	£10.00	Master Only	DB56	no	yes	£6.00
Mode 7 Utilities	DB12	yes	yes	£6.00	Compact 3 1/2	DB57	no	yes	£6.00
double disk					A&B'87	DB58	yes	yes	£6.00
Adventurescape III	DB10	yes	yes	£15.00	ADFS	DB59	no	yes	£6.00
ADFS	DB28	yes	yes	£15.00	Compact 3 1/2	DB60	no	yes	£6.00
Compact 3 1/2	DB35	no	yes	£17.00	Getting Into Assembler	DB61	yes	yes	£6.00
Combat Zone	DB21	yes	yes	£6.00	ADFS	DB62	no	yes	£6.00
Compact 3 1/2	DB32	no	yes	£8.00	Compact 3 1/2	DB63	no	yes	£6.00
Procyon	EB1	EPROM	only	£11.50	Getting Into Wordwise Plus	DB64	yes	yes	£6.00
Easyword	DB22	yes	yes	£7.50	ADFS	DB65	no	yes	£6.00
Electron +3	DE22	no	yes	£9.50	Compact 3 1/2	DB66	no	yes	£6.00
Videobase	DB24	yes	yes	£6.00	Getting Into Sideways RAM	DB67	yes	yes	£6.00
Delivery	DB25	yes	yes	£7.50	ADFS	DB68	no	yes	£6.00
Easy Font	DB26	yes	yes	£10.00	Compact 3 1/2	DB69	no	yes	£6.00
ADFS	DB39	no	yes	£12.00	DB68 DB69				
Compact 3 1/2	DB44	no	yes		Getting Into Pascal	DB70	yes	yes	£6.00
ADFS Menu	DB27	yes	yes	£12.00	ADFS	DB71	no	na	£6.00
Compact 3 1/2	DB31	no	yes	£14.00	Compact 3 1/2	DB72	no		£6.00
Graphics pack 2	DB28	yes	yes	£10.00	Adventure Special	DB73	yes	yes	£20.00
ADFS	DB34	yes	yes	£10.00	ADFS	DB74	no	yes	£15.00
Compact 3 1/2	DB38	no	yes	£12.00	Compact 3 1/2	DB75	no	yes	£20.00
Electron +3	DE38	no	yes	£12.00	Graphics Special	DB76	yes	yes	£15.00
Colour Ikon	DB36	yes	yes	£6.00	ADFS	DB77	no	yes	£15.00
A&B Bibliography	DB40	yes	yes	£15.00	Compact 3 1/2	DB78	no	yes	£15.00
ADFS	DB51	no	yes	£15.00	Disk User One	DU1	yes	yes	£2.95
Compact 3 1/2	DB54	no	yes	£15.00					

FULLY FUNCTIONAL

Have you ever needed to use the same routines again and again? We have the answer...

The *Disk Functions Library* is a must for any beginner or advanced programmer writing routines in BASIC; it provides the most useful and common facilities you will need for using disk drives and disk files within your programs without the need to waste time flipping through pages of a manual.

The procedures are real time savers for beginners unable to find get to grips with the tangle of OS calls and such lurking inside the BBC Micro.

These functions will work on all BBC micros – most of which are compatible with both DFS and ADFS filing systems, others are *filing system specific*, meaning they will not work on both filing systems. In such cases, I have tried to provide routines for both, and you should select between them with the help of a third routine which detects which filing system you are using.

Using the functions

The function lines start at BASIC line 30000. You can renumber these of course, but with this way it is very easy to add them to the end your own programs, which aren't likely to be longer than 30000.

Enter and save the library as "B.Library". (if you use ADFS, you will have to create a directory called "B" using *CDIR B.

To merge a library to your own program, type the following 4 lines:
LOAD "B.Library" *SPOOL Library
LIST *SPOOL

A file called **Library** will be saved to disk. To append the function library to a program (eg a program called "my-prog"), type in:

LOAD "my-prog"
*EXEC Library

Listing your program will show the library appearing at the end, starting at line 30000!

These functions do not use global variables (with the exception of 2 routines); This means, that the variables in your program are not

changed, even if the same variable names are used. This is achieved using the LOCAL definition which, when, once declared, will save the values of all external variables, and create a new set of them for use inside our own functions.

When a function is terminated, the original values are retrieved.

Some routines call other routines. This means that if you need only some of the functions, make sure you add only the functions that are needed by the routines you require.

You may rewrite the routines so that they occupy less memory, by making variable names shorter, deleting extra spaces and compacting lines.

Some routines need *entry parameters*, these are values needed to be declared *before* you go into the routine in order to set up values and so on. Although they may look technical, they really are quite simple really.

fsp\$ is the sorting variable set aside for a *filename*. This must stick to filename guidelines as stated in your DFS/ADFS manual.

trk% represents a track number. On a DFS disk, this may be between 0-39 or 0-79 depending on whether you use 40 or 80 track drives. On an ADFS disk this may be between 0-159 on an 80 track disc drive.

sec% represents the *sector number*. This is between 0-9 on a DFS disk and 0-15 on an ADFS disk.

unit% reflects the number of sectors. This must be within 1-10 on a DFS system and 1-16 on an ADFS disc.

addr% is the memory address, used in the load/save sectors/tracks routines. This space is best allocated by using the **DIM** command (see the BASIC manual). There are two important things to note here: The number *must not* be surrounded by brackets, and there must be a space between the variable name and the number. The number is calculated by: 256 * sectors-1.

For example, if you wish to load 5 sectors from disk, you would need $5*(256)-1=1279$ bytes, so you will define:

```
DIM memory% 1279
```

To load five sectors you would use:

```
err% = FNload - sectors(3,2,5, memory%)
```

This line will be explained later, but for the moment have a look at the last two numbers – they are 5 (which is the number of sectors to load), and the memory variable **memory%**.

handle% When opening a file it is assigned a value by the filing system; eg **f% = OPENIN("a-file")**. In this case, the handle of "a-file" is **f%**.

load% Represents the *load address* for a file, ie the address at which it will be used if it is loaded with *LOAD or *RUN. This can be in the range 0-&FFFF. Some areas of memory are unsuitable for loading in a program so please be careful (see the BBC user guide regarding the PAGE command).

exec% is the *execution address* for a file, ie. if a file is loaded using *RUN, the computer will jump to the specified execution address (see the BASIC manual for the *RUN command).

from% is used only in the **FN read-filenames** function, I'll get onto this in a minute.



The library routines

FNwhich-drive - DFS+ADFS

This function returns the current drive number. This is set using the *DRIVE <drive no> command in the DFS, or *MOUNT <drive no> on the ADFS. The routine is used by other routines. (load & save sectors) so if you want to use a selected drive you can use the above * commands to set it.

FNwhich-fs - DFS+ADFS

As I mentioned earlier, this can be a very useful function as it makes it possible to identify the current filing system. This function returns a string which is an abbreviation of the filing system name for example:

CFS1200 is the Cassette filing system (CFS) at 1200 baud (fast tape loading), whilst **CFS300** is the same at 300 baud (slow speed).

Others include:

RFS ROM filing system. **DFS** Disk filing system. **NFS** Econet filing system. **TFS** Telesoft filing system. **IFS** IEEE filing system. **ADFS** ADFS. **none** No filing system found. **FNfile-type(fsp\$) - DFS+ADFS**

This function tests the *type* of a given filename. The result is 0 if no file was found, 1 if the file exists or 2 if it is a directory.

FNsectors - DFS only

This returns the number of sectors on a DFS disk; each sector represents 256 bytes (4 sectors=1K). This works on DFS only, so a check should be made with **FNwhich-fs** before it is called.

FNdfs-load-sectors(trk%,sec%,unit%,addr%) - DFS only
FNadfs-load-sectors(trk%,sec%,unit%,addr%) - ADFS only

These simply load *unit%* sectors, starting at track *<B1%>trk%* and sector *sec%*, to the memory location specified by *addr%*.

For example, to load from track 4 sectors 2 and 3 and print an error if one occurs, you may use something like this:

```
DIM memory% 511
```

```
err% = FNload-sectors(4,2,2, memory%)
```

```
IF err% THEN PRINT "error";err%.
```

```
FNdfs-save-sectors(trk%,sec%,unit%,addr%) - DFS only  
FNadfs-save-sectors(trk%,sec%,unit%,addr%) - ADFS only
```

These functions are generally the opposite of the above - they save sectors from memory to disk.

```
FNdfs-load-track(trk%,addr%) - DFS only  
Ndfs-save-track(trk%,addr%) - DFS only  
FNadfs-load-track(trk%,addr%) - ADFS only  
FNadfs-save-track(trk%,addr%) - ADFS only
```

To simplify loading and saving whole tracks, these commands may be used. They use the external functions **FNwhich-fs**, **FNdfs-load-sectors** (**FNadfs-load-sectors** for a DFS disk) and **FNdfs-save-sectors**. The minimum memory allocated for these calls is **2559 bytes for a DFS disk, and 4095 bytes for an ADFS disk.**

FNfiles - DFS only

Sometimes it is important to find the number of files saved to a DFS disk due to the limit of 31 files on a

catalogue. (the standard). The external function **FNload-sectors** is used, and if no error occurred, then the number of files are returned. If an error occurred, a value of &FF is returned.

```
FNdfs-free-space - DFS only  
FNadfs-free-space - ADFS only
```

On some types of programs, it is convenient to inform the user how much space is on the disk. (For example, for a database program, the user should be informed how many records can be stored on a disk). The value returned represents the number of free sectors.

```
PROCupdate(handle%) - DFS+ADFS 70
```

The BBC micro has a 256 byte memory buffer for each file open. When this buffer is full (or if you try to access a location on the disk not in the current buffer), it is updated. It is also updated after a **CLOSE#** command (see the User Guide).

This means that pressing **BREAK** without previously closing the open files, may lead to loss of data. To avoid this, **PROCupdate** will ensure that the latest copy of the memory buffer is saved. If *handle%* is assigned, a handle number of an open file will be updated, alternatively **PROCupdate(0)** will update *all* open files.

```
FNadfs-read-filenames - ADFS only
```

This function reads the filenames on the current directory and puts them into an *array*. Before calling this routine, you must declare in your program:

```
DIM fname$(50)
```

The files will be loaded into this array, from the first position. The value returned by the call is the number of files read - this will work on DFS disks but has many limitations, I have written the next function to do the job better.

```
FNdfs-read-filenames - DFS only
```

There are four differences between this call and the previous one. It is DFS only compatible, it is much faster, it will read *all* of the filenames from the disk from *all* directories, and the directory letter is assigned to the filename (ie "E.file").

Before calling this routine, you must make some space in order to put the filenames somewhere:

```
DIM fname$(30)
```

Files will be loaded into this array, from the first position. The call returns the number of files read, or -1 if an error occurred. Functions used with this function are **FNfiles** and **FNload-sectors**.

```
FNdisk-title - DFS+ADFS
```

This is a favourite of mine! It is possible that you will change your disk, and you will therefore try to

write to an open file that doesn't exist! If this is the case, it is possible to check that it is the same disk as before, when a read or write operation is required. Using this function, the disk title is returned as a string.

```
FNread-load(fsp$) - DFS+ADFS  
FNread-exec(fsp$) - DFS+ADFS  
FNread-size(fsp$) - DFS+ADFS
```

These three functions return the load address, the execution address and the size of a program. If the specified file does not exist on the disk, an error message will be displayed and the program will stop.

```
PROCwrite-load(fsp$,load%) - DFS+ADFS
```

```
PROCwrite-exec(fsp$,exec%) - DFS+ADFS
```

These two procedures write both the load and the execution address of a file respectively. If the file is locked, then these procedures will not work.

```
FNdfs-write-count - DFS only  
FNadfs-write-count - ADFS only
```

Every operation performed on a disk (eg saving a file) is recorded. This function reads the total times the disk has been written, and can be used in a similar manner to **FNdisk-title**. **FNfile-locked(fsp\$) - DFS+ADFS** If a file is locked a value of **TRUE** is returned, otherwise the result of the call is **FALSE**. If the specified file does not exist then a message will be displayed.

```
FNadfs-read-directory - ADFS only
```

This function reads the directories on the current directory into a global **DIM** declaration. Before calling this routine, you must declare in your program:

```
DIM dir$(50)
```

The directory names will be loaded into this array, from the first position. The value returned by the call represents the number of directories read.

A final word

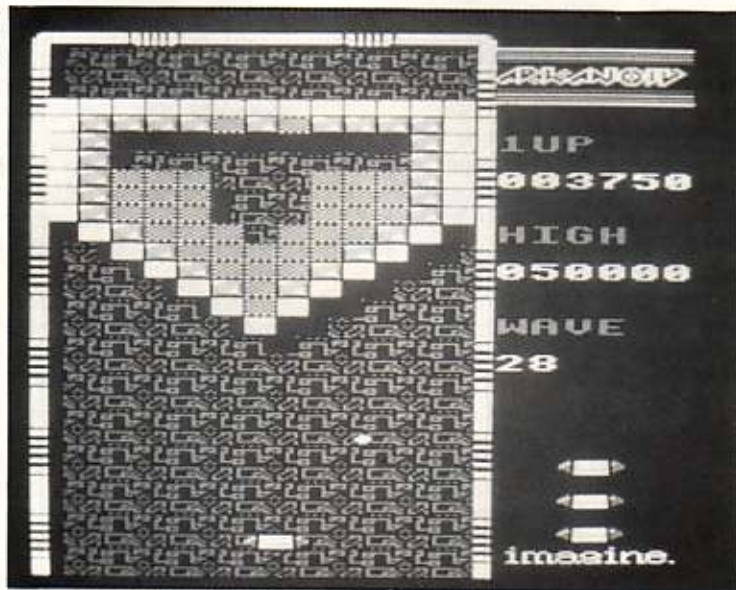
● *Advanced users note!* BBC B owners using the Solidisk ADFS 2.1 20/2 (and maybe later versions too) should be aware that there is a bug in the ROM. It does not run **OSFILE** correctly with **A=2** or **A=3**. (it should have written the load address only for **A=2**, and write the execution address only for **A=3**, but it writes the other address as &0000).

This means that users with the faulty ADFS ROM cannot use the procedures **PROCwrite-load** and **PROCwrite-exec**. This bug is not present on the Acorn ADFS 1.30 which executed these 2 procedures correctly.

I hope you will find most (if not all) of the functions and procedures useful, happy programming!

FLOPPY FUN!

Cheats for 1987's top disk games!



Once again this column sends out a plea – so far in *Floppy Fun* we've looked at classic games on disk, supplied a few useful pokes and tracked down a great *Repton 3* map printer, but what else would you like?

Until a clear consensus appears,

this column will provide you, every two months, what seems most appropriate from the material to hand, but I would value your thoughts.

Coming up next issue, for example, will be a complete solution to

Superior Software's *Palace of Magic*, together with some cheat programs to allow you to explore all of the game fully. Naturally, after the competition has closed!

FLOPPY FUN! FLOPPY FUN! FLOPPY FUN!

You're a brick!

Most requested cheats of the year were for Peter Johnson's excellent arcade conversion *Arkanoid* from Superior Software.

The most simple cheat gives you infinite lives, so Graham Veitch's short program does just that. Simply RUN it and then CHAIN the first *Arkanoid* program.



Go for the burn

Thrust from Superior (again!) was another hot favourite but the combination of tricky gravity and constant dexterity required got the better of many of you (me too! – Ed.)

Graham Veitch comes to the rescue again with a short program allowing you infinite fuel, infinite lives and the option to jump to any level – keep them coming in Graham.

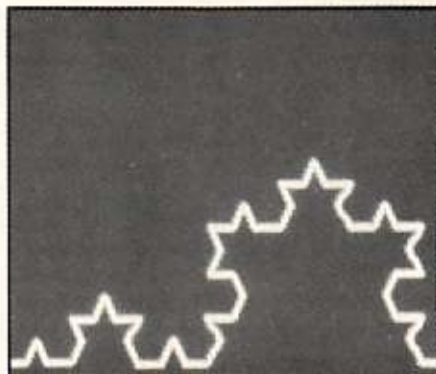
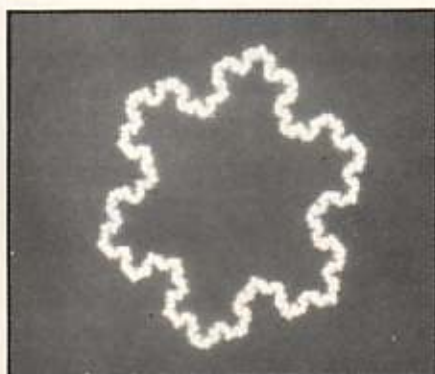
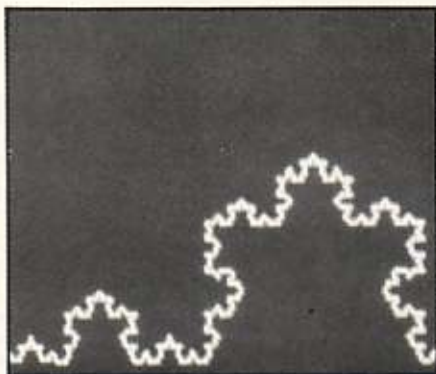


Success – at last!

Some of my favourite games playing moments of the year were spent with *Ravenskull*, a Superior Software classic written by Martin Edmondson and Nicholas Chamberlain.

Here on *Disk User* are the maps to the four levels – just cycle through the quartered maps and return to the game if you never completed it. It's still a real winner.





QUICKSTEP DEMO

Just one of the software packages available this issue from our SER-

VICES section. QUICKSTEP is an animation suite which provides a professional machine code finish to

your BASIC drawing routines.

Anyone interested in graphics can take advantage of our graphics special offer, which includes QUICKSTEP plus:

3D GRAPHICS
ICON DRAWING PACKAGE
LOGO DESIGNER
AND MUCH MORE!

Order product number DB76 for the DFS version or DB77 if you prefer an ADFS original. See SERVICES for full details.

Michael Spalter



TOOLKIT

Character Editor

This utility will either display the VDU 23 commands for all current characters or allow you to edit any of them on a large grid. To list the codes, type 'L' from the sub-menu and then enter the character you wish to start from.

The character editor is selected by typing 'E' from the sub-menu. All current characters will be displayed on a grid and a prompt will appear in a box on the bottom right of the screen asking "Character:" To this you enter the ASCII code of the character you wish to edit as it appears on the grid above. Characters with ASCII values less than 32

cannot be edited. Once the number has been typed in, that box will be filled with a list of each key's function. The character is edited by means of moving a cursor around a large grid sub-divided into 64 cells, and filling or clearing the relevant cell(s). The function of each key is as follows:

Z X : /

These control the movement of the cursor around the grid which are left, right, up and down respectively.

V

This will display the VDU23 code for the current character just above the grid. This display is non-interactive and needs to be pressed every time the code is required.

S

This will save characters 224-255 to disk, after you have entered a suitable filename (less than 8 letters long).

E

This exits back to the program sub-menu.

<RETURN>

This key will fill the cell over which the cursor lies at the time. If the cell is already filled, nothing will happen.

<SHIFT>

This key does the opposite of the above, ie clears a cell.

<ESCAPE>

This will return you to the start of the program, to edit another character.

disk

LINK

TO ADVERTISE IN THIS
NEW SECTION — RING

MARK LINACRE ON 01-437 0626

HIGHEST
QUALITY

Premium
Diskettes

LOWEST
PRICE

EACH DISK 100% CERTIFIED AT MANUFACTURER

- 5¼" DS/DD and SS/DD Diskettes (Sleeves Included)
Box of 10 — £4.99 10-99 — .38p each
- 5¼" High Density AT Diskettes (Sleeves Included)
Box of 10 — £7.40 10-99 — .68p each
- 3½" Diskettes DS/DD and SS/DD
Box of 10 — £9.90 10-99 — .88p each

CERTIFICATION: Every Diskette is tested and certified to be 100% error-free.

All Diskettes are certified to meet or exceed ANSI standards.

Includes: Sleeves, write protect tabs and reinforced hub rings.

★ SPECIAL OFFER ON 1000 DISKETTES OR MORE

★ PRICES INCLUDE POSTAGE AND V.A.T.

A.M.A. SHIPPING & TRADING CO. LTD.

Regd. Office: 198 Belmont Road, Erith, Kent DA8 1LQ.

Telephone: 03224 31771 — Telex: 8951182 GECOMS G

Quality Disks

Unbranded Lifetime Guarantee

Qty.	10	25	50	100	200
5.25" DS/DD 48tpi	5.70	11.00	19.00	34.00	61.00
5.25" DS/DD 96tpi	6.10	12.50	20.50	36.00	69.00
5.25" HD 1.6Mb	16.00	36.00	70.00	130.00	250.00
3.5" DS/DD 135tpi	11.50	26.60	51.90	98.00	189.00
3.5" SS/DD 135tpi	10.95	25.50	48.90	95.00	185.00

Please Call for 200+ Quantity Discounts.

5.25" 10 Disk Plastic Lib. Case £1.50 / 3.5" 10 Disk Plastic Lib. Case £1.30.

All Disks supplied with Labels. 5.25" also have sleeves and write protects. Price inc. VAT & P&P (U.K.)

Unbranded does not always mean poor quality, why not take advantage of our 14 day money back guarantee and try a trial pack of 10 disks.

Credit Cards Welcome

Telephone 0329 282083 (24 Hours)

Cheques/PO to: Athene Consultants, Dept. (DU),
33 Holly Grove, Fareham, Hants. PO16 7UP.

JOIN OUR NEW SECTION IN 'DISK USER' MAGAZINE

— DISK LINK —

Phone Mark Linacre

on

01 437 0626 for details
of how you can use this
space to market your
products/services.

Dear Sir,

I think this advertisement
breaks some rules

Advertisements are expected to conform to rules and standards laid down by the Advertising Standards Authority. Most do. The few that don't we'd like you to write in about.

And if you'd like a copy of these rules for press, poster and cinema advertisements, please send for our booklet. It's free.

The Advertising Standards Authority.

We're here to put it right. ✓

ASA Ltd., Dept. Y, Brook House, Torrington Place, London WC1E 7HN.

This space is donated in the interests of high standards of advertising

A&B

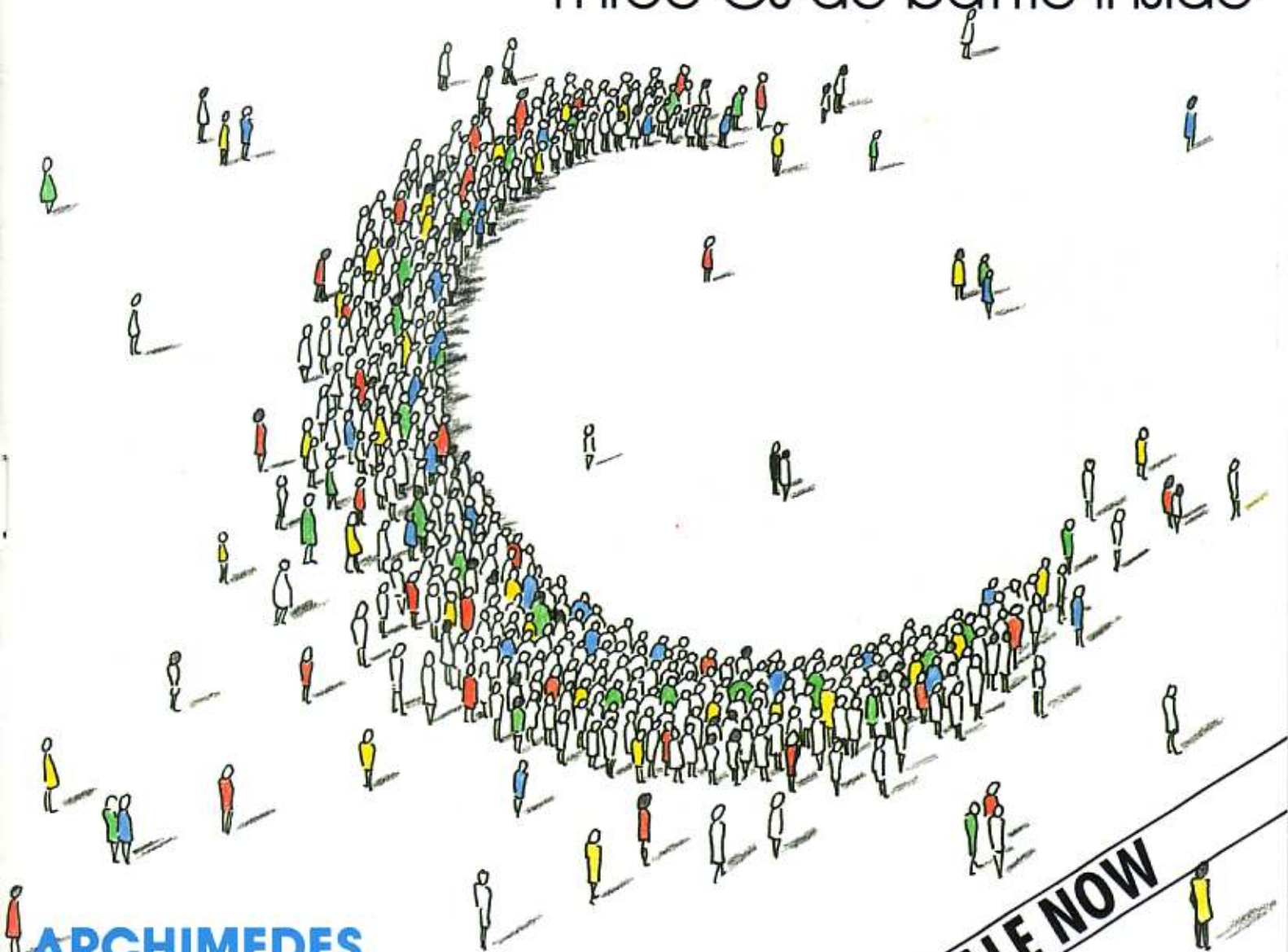
February 1988
£1.50

COMPUTING

FOR BBC MICRO, BBC MASTER & ELECTRON

THE C LANGUAGE

Three Cs do battle inside



ARCHIMEDES

ArcWriter reviewed

DELTABASE AND INTERBASE

ELECTRON SPECIAL

Teletext at last

ON SALE NOW



Bono takes the skeletons to a cauldron



Bono's partner Fozzy restrains a monster

BONE CRUNCHER

A PRIZE FOR EVERYONE WHO COMPLETES THE GAME

Buy Bonecruncher and see if you can complete all the game screens!

All successful players will receive a superb portable radio with headphones (illustrated on the right), a colourful Bono's Bathing Co. badge, and a signed certificate.

Closing Date: 30th April, 1988



• Full of Strategic Puzzles and Problems

• Over 20 Increasingly Difficult Caverns

• Password Feature helps you to Progress

• Prize Competition for Skilled Players

• Humorous, Witty, a Barrel of Laughs

Bono The Dragon and his Flourishing Business

You are Bono, a friendly dragon residing in a picturesque sea-castle. As a shrewd business-dragon, keen on "nice little earners", you make your living by selling soap to the giant monsters who bathe in the sea surrounding the castle.

With the aid of your business partner, named Fozzy, you produce the soap by collecting skeletons amongst the castle caverns and then boiling the bones in a cauldron.

However, the castle caverns are fraught with danger: there are monsters who will kill you, spiders who will eat you alive, and glocks who will block your path. It will pay you dividends to also avoid the perilous trapdoors and hidden volcanoes.

A soap-opera not to be missed!

COMMODORE 64/128 • COMMODORE AMIGA
BBC MICRO • ACORN ELECTRON

Cassettes: £9.95 (C64, BBC, Electron)
5 1/4" Discs: £11.95 (C64, BBC, Electron)
3 1/2" Discs: £14.95 (Amiga, BBC Master Compact)



Dept. BC8, Regent House, Skinner Lane,
Leeds LS7 1AX.

Telephone: (0532) 459453



24-HOUR TELEPHONE
ANSWERING SERVICE FOR ORDERS

OUR GUARANTEE

- All mail orders are despatched within 24 hours by first-class post.
- Postage and packing is free.
- Faulty cassettes and discs will be replaced immediately. (This does not affect your statutory rights).