

MONTHLY

AN AMIGOS/FOCUS PUBLICATION

ISSUE 40 (MAY 1988)

FOR ALL SINCLAIR USERS

**MINI STUDIO:
TYPE-IN SOUND SAMPLER**

**FIRST STEPS IN
MACHINE CODE**

**THE DISCIPLE:
REVIEWED**

**PRINTER
GRAPHICS**

MONSTER HITS:

Silent Service | Goumlnet
BAX Simulator

COMPETITIONS!

Icon Graphix
Cosmic Shock Absorber

IMPOSSIBLE BALL

IMPOSSIBLE BALL -
A BOUNCING
3D TEST?

AMSTRAD 8.95 COMMODORE 8.95 SPECTRUM 7.95



It was a brilliant scientific accident but Number Five, designed to be a strategic artificially intelligent weapons system, the most sophisticated robot on the planet, has woken up — and here comes to the conclusion that he's alive! Now the scientist who put him together wants to take him apart

SHORT CIRCUIT

again to find out what went wrong. The president of Nova Robotics wants to capture him before the weapons he's carrying kill millions of civilians. And the security chief wants to blow him up so that he can get home in time for dinner. YOU are Number Five... YOU are alive and YOU have got to stay that way!

Ocean Software Limited
Ocean House, 4 Canal Street, Fleet Street, EC2A 4NE
Telephone 044 831 6631 Telex 899110 Ocean G



Short Circuit is a trademark of Ocean Software, Inc. and TCI Properties.
© 1989 Ocean Software, Inc. and TCI Properties. All Rights Reserved.



FEATURES

HEX STERS IN MACHINE CODE Advice for beginners from Alan Davis. 18

EXPERT SYSTEMS Let your computer decide. 19

PRINTED GRAPHICS How to get the best out of your printer. 22

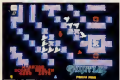
GREAT DISPLAYS A roundup of all the best graphics hardware and software. 24

DISCIPLI Could this be the ultimate interface? 26

MSB STUDIO A sound compiler for the Spectrum. 28

THE VALUE OF LABELS Peter Cook, author of *Assembly and You*, reveals some of his programming secrets. 33

SYSTEMS AND CHANNELS Test these ways that you can use your ZX Printer with the Spectrum 128. 34



Gauntlet (32)

PROGRAMMING PROJECTS

PROJECT X Ray Lister shows down a programming challenge. 35

SPROWORD The last part of our wordy prog. 37

THE BETTER LITTLE Identifying *Isosword*. 38

COMPETIONS

ICOM GRAPHIC Win Autodesk's graphics utility — versions for both 88 and 1288 machines. 39

COSSHO SHOOS ARCOBES Markachi's latest maze game. 37

REGULARS

NEWS The return of Monty Mele. 4

SHORT CUTS More readers' routines. 42

64 COLUMN News of a pure Superdisk emulator. 47

ABOVE THE FORD US news. 50

CROQUISSE Readers' letters. 51

RANDOM MEMOIR Programming advice from Glynis Birk. 52

DISCOVERY COLUMN Software from Discovery centers. 54

CRONIKING Problem Page. 57

SOFTWARE REVIEWS

ICOM GRAPHICS 128 A graphics utility for the 128 from Autodesk. 39

SPROWORD Great Service (34), Impassant! (34), BMS Smasher (35), Silhouette Screen (34) *Short Reviews begin* 34

MINDPLEX The Silicon Dreams Trilogy from Electronic. 36



Football Fortunes (12)

NEWS

Grange Hill After Hours

The Grange Hill cast get the classic television treatment in Angus Fairs Software's release based on the ITV television book. Grange Hill After Hours, the plot in the animated teen drama game centres on late "Monday" character Lorraine to relieve the frustrated staff, most from the school. With the character Paul, Lorraine has to break into the school after hours, get his little room and to avoid the wrath of his staff.

Not surprisingly they run into many of the other characters such as Sylvia the generous and friendly Doctor. He is filled with love sometimes when things get busy he'll be pleased with to carry on. Levels of degree of interaction that enables you to be a character and this makes it an exciting game. The television Grange Hill on The Complete Games (1994)



Monty Returns Again

Quentin Monty Male is back for his fourth outing in 4th Wednesday Monty Peter Monty, author of the first Monty platform magazine.

His latest mission is to persuade you to read in the platform the best Monty game. In the work of Monty you must guide him across Europe while in the process picking up sufficient funds to buy the Great Island of Monty where he can enjoy a peaceful retirement. But will it be the best Monty game? Will Peter Monty wake up one day soon and find Monty in the streets doing a hobby loving and might it be even the platform game again. The answer of course lies in your own Monty from that point for the linked best game.



Spectrum Games Top Ten

- | | | | |
|----|----|----------------------|-------------|
| 1 | 3 | Quicker | US Gold |
| 2 | 1 | Football of the Year | Quicks |
| 3 | 4 | Tribal Pursuit | Demarc |
| 4 | 7 | Space Invader | Elite |
| 5 | 2 | Super Doctor | Imagine |
| 6 | 6 | The Great Escape | Quicks |
| 7 | 8 | Computer Arts 3 | Elite Jetty |
| 8 | 5 | Paper Boy | Elite |
| 9 | 9 | Headlock | Activision |
| 10 | 10 | Breakfire | US Gold |

(Chart supplied by Bill Smith)

Editor: Bryan Smith
Assistant Editor: CEB Jones
Assistant Editor: Guy Blair
Advertising Manager: John Maloney

Design: Angus Design
A.S. Advertising and Printing
No. 1 Golden Square, London W1S 2AG 01-427-6226

Printed by: Clive Webb, Milton Parkville

Advertisement Copy Director: Mike Bennett

Distributed by: Argus Press Sales and Distribution Ltd, 42-48 Red Street, London E2 7EF

© Copyright in this issue is retained by the Society of Authors. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the copyright owner.

This publication and its contents, including all articles, images, photos, drawings and other material, are the property of Argus Computer Publications (UK) Ltd. All rights reserved by the Society of Authors and other Intellectual Property rights and by those of administration arrangements are specifically reserved to Argus Computer Publications (UK) Ltd. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the copyright owner.

Argus Computer Publications Limited © 1993

Enduro Racer

Adventure game and its release: the combination of Sega's coin-op motorcycle simulation Enduro Racer for the game is a

quaffing trail bike simulation. Don't progressively rougher tracks. Enduro Racer will give you



Super Sunday

In the wake of the Super Bowl, Texas are releasing an American Football simulation called Super Sunday. Getting 60-65 Super Sunday will feature 30 of the teams who have appeared in the Super Bowl since 1960. Details of the 1967 Oakland Raiders franchise and the New York Giants will also be included.

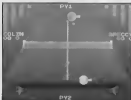
Super Sunday will combine entertainment with player performance. Based on real team statistics. When you've finished which team you'll have to play, and the players on screen move during the play. There is an in-play option or you can take on the computer. There is also an "easy" option where the computer plays two sides against the other based purely on the team's "real" stats.

Lift for Quicksilver



Exciting Action is a game where former lift operators will have a definite advantage. In this coin-op-powered on your small guide Agents One from the top of a skyscraper to the bottom where the gravity can't hold. 1200 coin sensitivity takes down on the ground on line is able to "shoot" jumps, like track and buckle from the top to safety! The game retails at \$9.95.

Indoor Games



Simulates real Adventure Golfers four more indoor games featuring 10 pin

loading, start, air hockey, and ping pong which will cost for \$9.95

Adventures On Paper

Incentive Software, producers of the Graphic Adventure Creator have released a new Designer tool for Adventurers. Each pack consists of over 100 double sided 4 1/2 pages and

is aimed at those who want to organize their adventure writing more efficiently. The GAC Designer Pack costs \$24.95 plus \$4.00 p.p. Incentive Software, (714) 770-0000

ICONS AND OLD MASTERS

COMPETITION

Here's your chance to win one of twenty-five copies of Icon Graphix



Twenty-five copies of *Icon Graphix* will be awarded to the artist whose work is judged to be the best. The competition is open to all artists who are currently employed in the UK. The competition is open to all artists who are currently employed in the UK. The competition is open to all artists who are currently employed in the UK.

All you have to do to win is correctly identify the three famous pictures on this page and name the artist who painted them.

If you can spot all three pictures and their painters and see among the last twenty-five entries pulled out of the hat, you will win the exciting mystery masterpiece on your Spectrum with *Icon Graphix*.

The competition is open to all UK resident, full-time employees of Acorn Specialist Publications, Chase Web and Astrogem.

Soborn Ltd. The editor's decision is final and no correspondence.

Send your entries to *Icon Graphix Competition*, ZX Computing, No 1 Golden

Square, London W1R 3AL.

Please remember to put your address on your entry envelope.

The closing date for entries is April 1987.

Icon Graphix Competition

- A) Title - () Artist - ()
B) Title - () Artist - ()
C) Title - () Artist - ()

Please tick your machine - 48K 128K

Name

Address

Send your entries to *Icon Graphix Competition*, ZX Computing, No 1 Golden Square, London W1R 3AL.



A 3D POW escape game that captures the imagination

The Great Escape Ocean £7.98

Locked in a prison camp your daily routine consists of roll calls, exercise periods and escape attempts.

If you're inductive for more than a few seconds you'll automatically follow the daily routine of sleeping, walking and eating if you want to try something else than a press of the fire button requires control. Now you can explore the camp, search and find potentially useful objects and scout out possible escape routes, but you must return to the camp routine if you want to stay out of solitary.

Slipping in and out of this "auto-prisoner" mode is essential if you are to escape as it ensures you get to roll calls, eat and sleep but allows you some time to explore the camp without rolling the clock.

Get caught in the wrong place at the wrong time and you'll be sent to solitary to cool your heels. This also destroys your morale and can also increase the number of guards that are assigned. A constant escort of three guards makes escaping almost impossible.

Your morale is extremely important and illustrated by a flag at the side of the screen.

As escape attempts fail and objects are confiscated your morale drops, the lower it gets the less likely you are able to break out of "auto-prisoner" and escape until finally at zero all player control is lost and the game is over.

After recent Ocean classics such as Miami Vice and Knight Rider I was surprised at how good this game was. It takes a little while to get into it as you just observe the normal prison routine.

Then you can find the opportunities to duck through a door, search the room contents, steal anything interesting and be out before the guards notice you're missing.

You can only carry around



two objects at the time and so you have to find a safe place to stash them before the big escape.

A good hiding place and a possible escape route is the tunnel that you can find by moving the boiler in your room. Unfortunately, the tunnel network is complex and difficult to navigate in the dark, but if you find it, it's worth it.

The game graphics are excellent as the camp is created from 255 scrolling 3D objects full of prisoners, guards, dogs, barbed wire buildings (most of them off limits) and searchlights.

All right the camp is enclosed in darkness that allows you to explore freely if you can avoid the patrols and the searchlights.

The odds however are against you and you'll probably spend the rest of the night in solitary. A brilliant idea, superbly implemented, Ocean's best ever Spectrum game!



The answer to the Spectrum-users prayer the **DISC**PLE

AVAILABLE NOW!

the all purpose interface to take
your Spectrum to the limit

DISK INTERFACE

- Access up to 1.0 Mb of storage on 2 drives
- Loads full disk program in 0.5 seconds
- Simple Disc System
- Compatible with any standard drive - single or double
- Simply interchangeable, 48/90 inch, 5, 8", 10", 11"

SNAPSHOT BUTTON

- Saves any program instantly - at any point you wish and as often as you like

POWER INTERFACE

- Compatible with the whole range of "Personal Power" units
- Easy-to-use power switching - 100% EFFICIENCY
- Permits use of 240V AC power packs

CURL JOYSTICK PORTS

- Simple and convenient installation

NETWORKING

- Connecting with the official Disciple to give up to 30 users access to any 1 computer and 100 drives for up to 64 computers
- 4800 baud 1 Channel



Use the Disciple with any Spectrum - and the results are **Disc**iplined!

the **DISC**PLE

£89.95 (incl. VAT)

The ultimate disk, printer, joystick, mouse, modem, disk, etc. Add 1 to the Spectrum

Designed by Mike Gosden, Technology for the People of the Future & Talking Spectrum of Weymouth, UK

FABULOUS PACKAGE PRICES

Disciple/Drive from £155

Disciple/Drive/Printer from £225

For more information call us

on 01-293 0191 or visit our

London showroom for a

demonstration

To Rockport Products, 81 Church Road, London W9 4DP

Pre-arranged POST FREE

to £89.95 each including VAT & P&P

Quantity DISCIPLE Interfaces

Please CASH/POSTAL ORDER for:

OR tick by ACCESS or VISA (tick as appropriate)

CARD NO.

EXPIRY DATE

SIGNATURE

NAME (Block Capitals)

ADDRESS

Postcode

Overseas orders and bank cheques welcome. Telex 448243 Tel: 90-2900

HANATOC



Specimens in October, Amstrad in November, Commodore 64 in December

All £100

DURELL

software getting harder

Sales Dept.,
Durell Lodge, Castle Green,
Farnham, Surrey TA11 4AB
England

DEEP STRIKE



1.999.000.000

Spectrum & Amstrad
available November 20th

Commodore 64
available January 20th

Other computers to Commodore
on 021 258 9388

Sales Dept.,
Castle Lodge, Castle Green,
Teunton, Somerset TA11 4NJ
England www.psygnosis.co.uk

DURELL

software getting better....





FOOTBALL FORTUNES

With a board game of a remarkable quality, Football Fortunes is an ideal choice for your PC.

**Brian Clough's
Football Fortunes
CD-ROM
£14.95**

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

Football Fortunes is a board game adapted for the home computer. It is a fast-paced, exciting game that is easy to learn and fun to play. It is a great way to spend an evening with friends or family.

COMPUTER GAMES



PROJECT X

Ray Eider sets the next programming challenge in this interactive series.

Welcome to the second in this occasional series intended to present a problem, discuss means of solving it, a solution and then leave it to you to produce the coding to do the job.

First may I say that I was pleasantly surprised at the good response to the last Project X. Entries are still coming and I will present the solution which I think is most efficient, original and interesting in the next issue, meanwhile let us to the project which is ...

The Shuffle

Over the last few years I have seen used and devised several ways of simulating the shuffling of a pack of cards. As a side product the solution to this may also be a means of producing a non-repeating sequence of random numbers which may widen the scope of the applications of this exercise.

1. Identify Problems.

We need to be able to select a series of numbers 1-52, which represent the 52 cards in a pack. It is a random set mixed up with no repetition of previously selected numbers.

2. Breakdown of process.

This is fairly straightforward in theory

- a) Set up a representation of a pack of cards.
- b) Swap the cards positions in the pack (simulate a shuffle)

3. Possible Approaches.

To produce the implementation uses several ways. The random number generator will obviously be called on in order to effect the shuffle. We need the most complex seen is likely to be the means of ensuring that selections are not repeated.

In the main basic computer card games (and in one I'll have been interested) games I have

seen this happen) a card is simply selected purely at random. The player loses when they lose the game when he/she draws two nine of diamonds or the ace!

One approach would be to select a card at random and then check it against a list of the cards previously chosen to make sure that it hasn't been picked before. This means that all previously selected cards have to be kept recorded to enable the check to be made.

A decision has to be made of the way the card will be selected. It could be a straightforward random number between 1 and 52 (INT (RND*52)+1) and then use a table to look up the actual card and value. It may equal the fact of 52*52 is the TWO of 52*52 will go to 52 perhaps being the KING of 52*52*52.

If this method is used then each selected number needs to be stored somehow, possibilities are to store it a number or as a string. It kept in numerical form then perhaps the simplest way would be to 794 on array variable to 52 = 52*52 = 2704. This also initialise each of the 52 elements to 0 and at each number is chosen then change that element to 1 at a flag that it has already been picked.

The check done would be something like IF (Number) WITH 52 TO check again's line.

This is a pretty OK for games where only a few cards are required each hand (such as pontoon) but if a lot of cards are used then the selection procedure takes to an unacceptable level of messy inefficiency and checks need to be made. You also have the problem of keeping another index of the numbers for reference.

An Alternative?

Rather than fill the empty array a card can be set to the number first then shift each number around. This also has the advantage of keeping a recording of each number.

Something like DIM S(52) FOR INT TO 52:FOR Q=1 TO 52

UP THE ARRAY AND THEN SELECT TWO RND NUMBERS BETWEEN 1 & 52 AND SWAP THE CONTENTS OF THE ARRAY. E.G. LET

```
A=INT(RND*52)+1:LET  
B=INT(RND*52)+1:LET  
S(A)=S(B):S(B)=  
S(A):S(A)=B
```

For this is a loop for as many times as it takes to get a good mix and away we go!

Of course we still need a table so that we can work out which number represents which card, but the actual shuffle takes a fraction of the time that the first method would take. Reasonably I wouldn't see either method, but I thought I worth looking at the first two was because the second method has a wider application than to our card shuffle in that it is a very effective way of producing a numerical non-repeating sequence of random numbers.

If you think of a pack of cards then there are two elements, the number or value and the suit — Hearts, Spades, Diamonds and Clubs — the non-numerical cards are Ace, Jack, Queen and King. It is almost as if the designer of cards had as computer keys in mind NONE of these names begin with the same letter!

This means we could set up a string to represent the cards using ASCII A-J, Q, and K to represent each of these words. A typical set up might be LET C="A1A2A3A4A5A6A7A8A9A10A11A12A13A14A15A16A17A18A19A20A21A22A23A24A25A26A27A28A29A30A31A32A33A34A35A36A37A38A39A40A41A42A43A44A45A46A47A48A49A50A51A52"

Note that I used 7 to represent 10 because this then means that each card is represented by only two characters. Shuffling becomes a matter of picking an element of 52 at random removing it from the string and adding it to the front or the end of the string for as many times as it takes to get a good mix.

If you set up the variables T, Q, A and B to a value of 10 then finding the numerical value of the card is a matter of getting the VAL of the first of the two elements that represent a card. So now what? I hope this has started a few ideas buzzing around and, just like the last time, we would like to see the results of your ideas, especially if you know of or create a completely unique method of solving the problem.

What we would like is a short program which shuffles a pack of cards and checks two cards face up. We do not want a full game but we will take into consideration the solution to the shuffle problem and the quality of the graphic presentation of the cards.

The program(s) which impress us most will be featured and their authors suitably rewarded, in a future issue.

FIRST STEPS INTO MACHINE CODE

Alan Davis makes the transition from BASIC into machine code.

Most of people seems to find no great difficulty in becoming reasonable proficient at BASIC programming, after a little practice and experience. But sooner or later they realise the limitations of BASIC, seek to go further — and find themselves up against the notorious "machine code barrier". It's a barrier that walls, many fall to (though often not for want of trying), and I gather that the question "How do I pass from BASIC to machine code programming?" is all one of the most common requests in the ZX Spectrum mailing. So here's my own attempt to provide a very simple approach with a minimum of jargon — by which I mean VERY SIMPLE indeed! I'm going to sacrifice discussion of a number of issues which would be theoretically desirable, because the most important thing is to make that initial transition and actually write machine code programs, no matter how trivial. Even if you've tried before and failed, I hope I can persuade you to have another go — because I'm convinced that most people give up simply because they've tried to swallow too much, too soon.

The cursed hex

If I have tried to come to grips with machine code before, you've probably been harassed by hexadecimal, "boxed" and "hex" complexities, and stuffed by "the stack" — but though you'll eventually need to get to grips with such things if you're to become really proficient, you can in fact get started without them. There are certain disadvantages in doing so — but none desperate, for, and disparate, features are called for. So let's begin. What, exactly, is "machine code"?

Though you may not be aware of it, you're using machine code every time you run a BASIC program on your Spectrum. You may think that the computer understands that the commands such as PRINT A:

PRINT B: — but it doesn't! When we program in BASIC we're being spoon-fed by a complex and lengthy program which is present in the Spectrum when we switch on, part of whose task it is to translate our BASIC

instructions into a form that the heart of the computer — the Z80 microprocessor — can cope with. What the Z80 understands is the stuff we call machine code: a sequence of numbers which, cause the Z80 to perform certain actions. If we bypass the Spectrum's "translating" system and load the Z80 directly with the correct numbers, we can exercise total control over the machine. Heresy stuff!

First steps

So let's write a machine code program. First of all we need to reserve an area of memory so that any jiggling about in BASIC that we do won't interfere with it and we can do this using `CALL` `MEMPR`. This means that we can safely re-allocate any address from 60000 upwards. Next, enter these two direct commands, one after the other:

```
POKE 60000,0
POKE 60001,0
```

What you're just doing is to store two numbers (which together make up a machine code program) in the Spectrum's memory at addresses 60000 and 60001. What we do now is to direct the Z80's attention to this sequence of numbers, and you can do this by entering `BANKSWITCH` `USR` `60000`. Try it yourself, now. You might think that nothing has happened — but you'd be wrong, quite a lot has happened! What? First, the Z80 looked at the contents of address 60000, and found the number 0. This is the machine code "NOP" `00000000` instruction which tells the Z80 to do nothing at all. So for a tiny fraction of a second it did indeed do nothing. Then it looked at the contents of the next address (60001) and found the number 201. This is the "RST20H" instruction, which in this case tells the Z80 to leave the machine code program at this point and effectively "return to BASIC". And so the obedient Z80, without further ado, did just that — which is why at the end

you were presented with the ZX report at the bottom of the screen. The machine code program has been completed, and normal service has been resumed.

Now the good news is that in understanding that, you've understood a machine code program. Of course you'll very rightly guess that it's not a very useful program — but that doesn't matter. The important thing to appreciate is that what to us looks like a string of numbers is effectively a sequence of commands to the Z80. Hence, of course, the 0 and neither are you. The number 0 doesn't immediately imply a problem; it's not a Z80, and neither are you. The number 0 doesn't immediately imply a "number 200 NOTHING", and the number 201 certainly doesn't make us think "STUPID" — if we had to learn to read and write machine code like this, life would be very hard indeed — but fortunately we don't have to.

What we need, really, is some sort of intermediate language to act as a buffer between ourselves and the Z80, and fortunately such a language does exist. It's called *Assembler* language, which is a kind of drastically abbreviated plain English where each machine code instruction (that is, a number or series of numbers) is represented by a mnemonic (a set of letters) which is much easier for us to understand. To take an example, if we translate our little machine code program above into assembler language, it looks like this:

```
NOP ;no Operation
RST ;return to BASIC
```

This is much easier to get to grips with. In fact you can understand it, I can understand it, but the bad news is that the Z80 can't! It won't recognise more than less than MEMPR's solution, but in an *Assembler* program — and this, I'm afraid is the point of which you'll need to know the deal of your cheque-book. There are plenty of assembler programs on the market and I'll try to return next time familiar with all of them. Most having suffered long ago on Home Drivers, assembler is also heard of in assembler programs available these days, more suited to beginners.

LISTING 1

Do nothing, then return to BASIC.

```

00000 10      ORG 00000
00000 20      NOP
00001 30      RET

```

LISTING 3

Load contents of one address, add one, and put result into another address.

```

00000 10      ORG 00000
00000 20      LD A, (ADDR1)
00003 30      INC A
00004 40      LD (ADDR2), A
00007 50      RET
00000 60      ADDR1  DEFB 2
00003 70      ADDR2  DEFB 0

```

LISTING 2

Copy contents of one address into another.

```

00000 10      ORG 00000
00000 20      LD A, (ADDR1)
00003 30      LD (ADDR2), A
00006 40      RET
00007 50      ADDR1  DEFB 2
00000 60      ADDR2  DEFB 0

```

LISTING 4

Add the contents of two addresses, and store result in a third address.

```

00000 10      ORG 00000
00000 20      LD A, (ADDR1)
00003 30      LD B, A
00004 40      LD A, (ADDR1)
00007 50      ADD A, B
00000 60      LD (ADDR3), A
00011 70      RET
00012 80      ADDR1  DEFB 2
00013 90      ADDR2  DEFB 3
00014 100     ADDR3  DEFB 0

```

Assembler

A good assembler program does nothing, but takes over the control of your Spectrum to allow you to enter and edit a program in assembly language, and will then "assemble" or translate your program into machine code. But it should also do much more. For example, it should allow you to use line numbers which, though not essential, help you to keep track of what you're doing. It will allow you to attach labels to certain addresses — which is yet another name besides naming your program more closely resembles English. And it will also allow you to use certain other instructions (assembler directives and assembler "statements") which are not translated into machine code, but which instruct the assembler program to take specific actions — and these generally make life easier for you. There's no shut out here, I'm afraid. You'll need to spend a bit of time reading and absorbing the relevant sections of the manual for your assembler in order to discover just what editing facilities are available to you. However, as example will make things clear — so let's take a look at listing 1, which is what I got after typing in (do nothing and return to BASIC) program

First, I should point out that in order to get all the listings given here I instructed the assembler to print addresses three or in the left hand column of the listing in decimal, rather than the more usual hexadecimal — and incidentally, the facility to have this option is something you may like to check when choosing your assembler. It can make things quite a bit simpler. I also instructed it not to bother listing the machine code that it produces; the Z80 needs those numbers, but we don't at this stage. Oh, then to the program proper. You'll see that line 10 contains an assembler directive which all assemblers will recognize, namely **ORG 00000**. The **ORG** directive is important because it tells the assembler where to put the machine code it generates — and in our case this is to be from address 00000 onwards.

Line 20 contains the first actual assembly language instruction (**NOP**) and line 30 completes the program with the **RET** instruction. The left hand column of the listing is produced automatically when the program is assembled (as when the assembler is instructed to translate the statements to actual machine code) and it shows the addresses of which each machine code instruction has been stored — in this case our machine code program

uses only two addresses, 00000 and 00001.

Registers

Now this is all very well, but you should remember by doing by this line: "Can we have a program which actually DOES something, please?" So let's move on and see what else the Z80 can do. You probably know already that the Z80 has a number of **REGISTERS**, and you may also know that these have some form of "R" register. "R" register "C" used to (or both of these registers can at any time hold any one number between 0 and 255 (ie. 0-255) — just like any address in the Spectrum's memory. And there are dozens of machine code instructions which allow the Z80 to copy a number from one register into another, and also to compare/relate with the value word in the Spectrum's memory. The assembly language instruction **LD B,A** (or simply **LD B,A**) means "copy the contents of the A register into the B register". Similarly **LD C,A** means "copy the contents of the A register into the C register". There are also instructions for putting numbers directly into registers — so that **LD A,20** for example means "put the number 20 into the A register".

As you can imagine, we could write a program like

program causing the 280 to shuffle away nervously to itself, recapping numbers back and forth between its registers without affecting the outside world at all — but I don't think that's what you mean by a program that "does something", so let's move locally on. How (do we pass information between the 280 and the Spectrum's memory)?

The simplest way of doing it is by means of an instruction such as LD A, ADDRESS. This would cause the 280 to look at the specified address in the Spectrum's memory and copy its contents into the A register. You can achieve the reverse process with the instruction LD ADDRESS, A — which would make the 280 copy the contents of the A register into the specified address in the Spectrum's memory (there's a limit to what you can do in this line, however, and the A register is the only single register which can communicate directly with specified addresses in this way — LD is address left allowed, for instance).

I think you'll see that we could use these instructions to copy a byte from one memory location to another, thus:

```
LD A, #0000
LD #0001, A
BT
```

This would load the number in address 0000 into the A register, copy it into address 0001, and then return to BASIC. (To do the same thing in BASIC, you'd use the instruction POKE #0001, PEEK #0000.) This would be fine if you knew in advance which memory locations to specify while writing your program, but in fact rarely often that way, you don't — at least not until you've assembled your code. It's time for the truly assembler program to come to the rescue again, so let's have a look at Listing 2.

Labels

There are two new concepts involved in Listing 2: both concerned with the ability of an assembler program to give you an easy ride. The first is the concept of labels. When I typed this assembly language program in, I decided to put my two addresses ADDR1 and ADDR2 respectively — and to lines 20 and 30 contain the instructions needed for copying the contents of ADDR1 into ADDR2. The assembler is perfectly happy about this provided I tell it what I mean by these labels somewhere — and I've done this in lines 50 and 60 using an assembler directive "DEFB number". (Think of DEFB as meaning "Define Byte") When the assembler encounters the DEFB directive it reserves a single byte of the Spectrum's memory

and inserts whatever number has been assigned — so in this case it will store the number 2 in ADDR1 and the number 8 in ADDR2. As you can see from Listing 2, the assembler has allocated ADDR1 the value 0002, and ADDR2 the value 0008, neatly leaving out reserved bytes away at the end of the machine code program block. Incidentally, you may have noticed that the addresses in the left hand column increase initially in steps of three. This is because the machine code instruction corresponding to LD A, address consists of three numbers — one to specify the original instruction LD A, address and two more to specify the address concerned, but don't worry about it for the present — that's the assembler's job, not yours.

You can try this for yourself by typing the program into your assembler assembler. It's fun returning to BASIC before running the machine code, check for yourself that PEEK #0002 and PEEK #0008 give the values 2 and 8 respectively. Now enter **SANDROMIZ USE #0000**, and do your PEKs again. This time BOTH addresses should contain the number 2, providing that your machine code program has indeed copied the contents of the first address into the second. **POKE** another number into 0002, try **SANDROMIZ USE #0000** again, and then have another PEEK.

Again, it's important that you do try these things for yourself, as opposed to merely reading about them, if only to become familiar with the use of your assembler program. By entering the program in Listing 2 so that it will copy the contents of ADDR1 into TRO other addresses (you'll need to add another instruction LD #ADDRESSA, and use DEFB to reserve another byte labeled ADDR3).

When you've mastered the foregoing you'll want to switch your newly grown machine code wings a little further instead of merely copying numbers from one place to another, but see if we don't stop a few changes. Actually it's not difficult to do so, because simple instructions exist for increasing or decreasing the contents of registers by 1 — for example, INC A (increase contents of A register by 1) and DEC A (decrease contents of A register by 1). Listing 3 shows how INC A can be used in a program to transfer the number held in one memory address to another, increasing it by one en route. Try it for yourself! Filling the appropriate addresses both before and after the machine code has been run to check that it all is well, and then try writing your own amendments using DEC A — or by using

INC A twice in succession to increase a number by 2.

Adding

Our last example — listing 4 — shows how simple addition of small numbers can be achieved using the instruction ADD A, B (ADD contents of B register to contents of A register, leaving the result in A). I try to follow this routine through, you'll see that the contents of the memory address ADDR1 are copied directly into the A register, and then into the B register using LD B, A. (We need to load the indirect route because there's no LD B, ADDR1 instruction.) Then A reads in the contents of ADDR2, and the addition is performed by ADD A, B. The result is now in the A register, so we finished off by copying it into B (using MOV B, ADDR3 before returning to BASIC). Again, you can check it all for yourself by filling ADDR1 #0002, ADDR2 #0004, and ADDR3 #0016 before and after running the machine code. You can also try different numbers of course, by copying them into ADDR1 and ADDR2 from BASIC. If you want to try subtractions, then by all means do so as the instructions you need is SUB B (subtract contents of B register from contents of A register, leaving the result in A) — but for the present you'd do well to ensure that you use a number which will produce a positive result.

This is all for as I intend to go in this article, and you may be understandably disappointed that my progress has been decidedly limited. Well yes, it has — and very deliberately so. I know that you want to learn about all the other registers to be able to use programming loops, to handle numbers in excess of 255, and lots more besides. That's a job for the future, though, and the important thing for now is to get your feet on the last rung of the machine code ladder. It's time for critical transition towards thinking in terms of registers and addresses. Use the space in this article to write simple routines for yourself — no matter how useless they may be except as practice examples — and then congratulate yourself when you get them working. Familiarity is the key.

Once you've got this for then you're on your way. But what next? Well, you could do much worse than to buy "Machine Code for Beginners" published by Osborne & Jones (1982). This will pretty fill in much background material that I've omitted here, and indeed will take you further on. So, you could well still need months when I'm sending further into machine code ...



SIGMA



Commodore 64 available November 2008
Spectrum & Amstrad available February

DURELL

Sales Dept.,
Cannon Lodge, Cannon Green,
Totton, Southampton TA1 4AD
England. www.durell.com

0 17 25 08

Trade enquiries to: enquiries@durell.com

tel: 017 256 8088

Software getting harder...

EXPERT SYSTEMS

Transforming your Spectrum or QL into an oracle of advice — with David Nowatnik.

Over the months, in part 1 of this series, I described the essential elements that go to make a computer expert system. As a reminder, an expert system provides the means to probe into a computer, not only the knowledge but the experience of a human expert, which a non-expert can interrogate to obtain benefits such as advice. Table 1 lists the important parts of an expert system, and in this article, due in the next few months, I'll be examining some of these elements, providing demonstration programs for QL and Spectrum computers.

Like any other programming problem, there is no single solution to the production of an expert system. There are many different libraries and more than a few are being developed, so all I can give you is the real-life example in a picture of what makes an expert system. This month we shall examine two ways in which you can get a computer to come to a decision.

So, going to a human expert for advice, and the exact course of events is something like this:

First of all the human expert will ask you to define the problem, to determine exactly what you wish to know from him. Next he will ask you questions. These questions are selected by the expert to provide him with information which he can match against his own knowledge base. It is experience which guides the direction of his questions. In an ideal world, he will end up with sufficient information to provide a perfect match with something in his knowledge base to give you a simple, unambiguous answer.

The real world isn't quite like that; your expert's knowledge base may not be perfect for the problem you have, you may not be able to provide all the information required. You are more likely to end up with one or more opinions from your expert, which compare with different levels of certainty.

Lack of certainty (uncertainty) is something an operator tries to limit by the series of how, let's consider ourselves and our computer expert to be perfect.

The majority of computer

Table 1. Main Elements of an Expert System

1. Knowledge Base
2. Inference Engine
3. User Interface
4. Use of Statistics and Probabilities
5. Ability to learn new rules/knowledge
6. Provide an explanation of its decision

expert system) produced to date are designed to tackle a fairly specific problem, i.e. with the computer expert, you have already defined the problem you wish to have solved by selecting a particular expert system. When you RUN any expert system, you should get a message to tell you what class of operation that program is capable of dealing with, and you will immediately start the question and answer dialogue.

How does the expert know which questions to ask, and what does he do with the information he receives? There are two important questions in designing an expert system in the rest of this article, I'll be considering two solutions.

Consider this rather trivial problem: You want your expert to tell you the name of a particular mode of transport. The expert will ask you a series of questions relating to features of the type of vehicle you have in mind, and after a number of questions and answers, he is able to name the type of transport.

Yes, it is trivial, but I have deliberately chosen an example in which we should all be experts, so that you can understand the logic of the interesting questions, and the interpretation of answers.

In the example program I provide, our computer expert is based on 25 modes of transport (see program lines 550 to 950 in Fig. 1 or Fig. 4). One strategy for finding the right answer is to ask a series of questions, each answer in turn able to eliminate a group of vehicles, until only one is left. This strategy is called serial decision making.

Pathways

For this example, the human expert can be split up into two

selected a set of questions to which the user provides yes/no answers. There is a set path through a series of questions from the start of the interrogation to any one particular answer. Fig. 2 shows all the paths through the question and answer in the expert system listing given in Fig. 4.

The first question is placed the same, does the vehicle transport people on land? A 'yes' answer immediately eliminates water craft and aircraft; the expert then asks if it travels on rails. If the answer is 'yes', the problem is solved, and the answer 'train' is given; if the reply is 'no', then the expert knows the vehicle travels on road, and two more questions are required to determine which of the four road vehicles known to our expert is the one in question.

For any expert system using the serial approach to decision making, the human expert designing the system has to be aware of all possibilities. The questions have to be unambiguous, so that the user can provide the correct answer every time, any else a random response always ends in a wrong answer from our expert.

But, before we examine the listings of this approach too deeply, why not try it out for yourself? The listing in Fig. 4 is suitable for both the QL and Spectrum; the program was written on the QL, hence the command GPRINT, which will appear to SH on the Spectrum, but listed on both computers.

QL SuperBASIC perforations will be handled by all the (5040) and (5060s), but some carelessness had to be made for the Spectrum less structured BASIC. Spectrum users should use '4' instead of '8' for joining strings (lines 5203 and 5310) and

```

30 REMark Single Expert System
35 REMark Using several decisionmaking
35 REMark
40 REMark David Borwick
50 REMark University, SDET
60 REMark
100 REMark Installation & Title
110 LET n=2000
120 REMARK n = 2000 is
130 LET n=2000 n
140 REMARK n
150 PRINT " I will ask you 2 series of "
160 PRINT " questions. Answer with a "
170 PRINT " yes or no if you can "
180 PRINT " each one while I practice "
190 PRINT " you with the software. "
200 PRINT n
210 REMARK
220 REMARK
230 REMARK Break Loop
240 PRINT n
250 REMARK n
260 REMARK n
270 IF n=0 THEN GO TO 3000
280 GO END STATE
290 REMARK n IF n=0 THEN GO TO 3000
300 IF n=0 THEN GO TO 3000
310 PRINT "Can it travel faster than a speeding "
320 IF n=0 THEN GO TO 3000
330 PRINT "Can it travel faster than a speeding "
340 IF n=0 THEN GO TO 3000
350 REMARK
360 REMARK
370 REMARK
380 REMARK
390 REMARK
400 REMARK
410 REMARK
420 REMARK
430 REMARK
440 REMARK
450 REMARK
460 REMARK
470 REMARK
480 REMARK
490 REMARK
500 REMARK
510 REMARK
520 REMARK
530 REMARK
540 REMARK
550 REMARK
560 REMARK
570 REMARK
580 REMARK
590 REMARK
600 REMARK
610 REMARK
620 REMARK
630 REMARK
640 REMARK
650 REMARK
660 REMARK
670 REMARK
680 REMARK
690 REMARK
700 REMARK
710 REMARK
720 REMARK
730 REMARK
740 REMARK
750 REMARK
760 REMARK
770 REMARK
780 REMARK
790 REMARK
800 REMARK
810 REMARK
820 REMARK
830 REMARK
840 REMARK
850 REMARK
860 REMARK
870 REMARK
880 REMARK
890 REMARK
900 REMARK
910 REMARK
920 REMARK
930 REMARK
940 REMARK
950 REMARK
960 REMARK
970 REMARK
980 REMARK
990 REMARK

```

Figure 4. Decision tree for the transport serial decision making example



for 60 users, the commands in lines 280 and 290 are optional.

The DATA statement in line 600 provides the title and purpose of our expert system. Once this is printed (line 100) and general instructions appear on the screen (line 150 to 190), the main question and answer loop is entered. The questions, answers, and knowledge base all appear in DATA lines, and REVERSE is used extensively to retrieve the appropriate piece of information.

On the first circuit of the main loop, the variable n is set to 2000, which is the DATA line for the first question. From this DATA line, four numeric options are READ to b, and c. b is the (line number of the question (character string) to be asked, 'n' is the appropriate DATA line in the knowledge base if the

Figure 2. Alternative knowledge base for Logic Generation Program

```

800 DATA "FATAL AND SPHERICAL BONES"
900 BONES
910 BONES BONESBONE
920 BONES "Is this bone in the hand?"
930 BONES "Is this bone in the ear?"
940 BONES "Is this bone in the upper arm?"
950 BONES "There are two bones in the lower arm in
10 the one direction the body with palm forward"
960 BONES "Is this bone in the finger?"
970 BONES "Is this bone in the end of the fingers?"
980 BONES "Is this bone in the upper foot?"
990 BONES "Is this bone in the mid foot?"
1000 BONES "Is this bone in the heel?"
1010 BONES "Is this bone in the lower leg?"
1020 BONES "There are two bones in the lower leg in
11 the anterior view"
1030 BONES "Is this bone in the shoulder?"
1040 BONES "Is this bone attached to the shoulder?"
1050 BONES BONESBONE
1060 BONES BONESBONE
1070 BONES BONESBONE
1080 BONES BONESBONE
1090 BONES BONESBONE
1100 BONES BONESBONE
1110 BONES BONESBONE
1120 BONES BONESBONE
1130 BONES BONESBONE
1140 BONES BONESBONE
1150 BONES BONESBONE
1160 BONES BONESBONE
1170 BONES BONESBONE
1180 BONES BONESBONE
1190 BONES BONESBONE
1200 BONES BONESBONE
1210 BONES BONESBONE
1220 BONES BONESBONE
1230 BONES BONESBONE
1240 BONES BONESBONE
1250 BONES BONESBONE
1260 BONES BONESBONE
1270 BONES BONESBONE
1280 BONES BONESBONE
1290 BONES BONESBONE
1300 BONES BONESBONE
1310 BONES BONESBONE
1320 BONES BONESBONE
1330 BONES BONESBONE
1340 BONES BONESBONE
1350 BONES BONESBONE
1360 BONES BONESBONE
1370 BONES BONESBONE
1380 BONES BONESBONE
1390 BONES BONESBONE
1400 BONES BONESBONE
1410 BONES BONESBONE
1420 BONES BONESBONE
1430 BONES BONESBONE
1440 BONES BONESBONE
1450 BONES BONESBONE
1460 BONES BONESBONE
1470 BONES BONESBONE
1480 BONES BONESBONE
1490 BONES BONESBONE
1500 BONES BONESBONE
1510 BONES BONESBONE
1520 BONES BONESBONE
1530 BONES BONESBONE
1540 BONES BONESBONE
1550 BONES BONESBONE
1560 BONES BONESBONE
1570 BONES BONESBONE
1580 BONES BONESBONE
1590 BONES BONESBONE
1600 BONES BONESBONE
1610 BONES BONESBONE
1620 BONES BONESBONE
1630 BONES BONESBONE
1640 BONES BONESBONE
1650 BONES BONESBONE
1660 BONES BONESBONE
1670 BONES BONESBONE
1680 BONES BONESBONE
1690 BONES BONESBONE
1700 BONES BONESBONE
1710 BONES BONESBONE
1720 BONES BONESBONE
1730 BONES BONESBONE
1740 BONES BONESBONE
1750 BONES BONESBONE
1760 BONES BONESBONE
1770 BONES BONESBONE
1780 BONES BONESBONE
1790 BONES BONESBONE
1800 BONES BONESBONE
1810 BONES BONESBONE
1820 BONES BONESBONE
1830 BONES BONESBONE
1840 BONES BONESBONE
1850 BONES BONESBONE
1860 BONES BONESBONE
1870 BONES BONESBONE
1880 BONES BONESBONE
1890 BONES BONESBONE
1900 BONES BONESBONE
1910 BONES BONESBONE
1920 BONES BONESBONE
1930 BONES BONESBONE
1940 BONES BONESBONE
1950 BONES BONESBONE
1960 BONES BONESBONE
1970 BONES BONESBONE
1980 BONES BONESBONE
1990 BONES BONESBONE
2000 BONES BONESBONE

```

answer (a, yes, and) g) is the DATA line number should the response be no. If the variable `%READ%` the relative zero, this indicates that an answer had to be found, and the loop is terminated. The sub-routine at 5040 ensures that the question is really printed on the screen while lines 5050 to 5060 print the answer, and ask you if you want to run through another set of questions.

Ambiguity

Like most experts, while I define my questions to you (I'll proceed in this computer program) the computer (inexpensive) you may find them confusing. This expert has no opinion for the user to say "I don't understand. Can you please rephrase the question." (also, the code for most expert systems) from the first question "Does it travel on land?" may provide uncertainty. A heuristics (heuristics can travel on land, but I, as the transport expert, decided that heuristics are used principally for sea transport, therefore would expect a "no" answer to the "land" question if the user was thinking of the features of a heuristics).

But like this system or not, it is the way many human experts agree that only the program I have given you (provided an expert system shell) to which other knowledge bases can be attached. Fig. 3 contains an example. This is an alternative knowledge base which allows you, using the same program, to determine the name of one of the body's joints or appendages bones. Lines 800 to 830 replace the transport

knowledge base in Fig. 1. Here, I assume, there are less experts among our readers, and you should, as non-experts, get some benefit on terminology on human bodies.

Once you feel you understand the logic of this expert system shell, there is no reason why you cannot build your own expert application. The first step is to draw up a decision tree, or flowchart in Fig. 3. Then translate that to the code required for the knowledge base.

The main criticism among us would say that this approach to making decisions is for the hobbyist. It is a little task to simulate a whole range of possibilities on the outcome of just one question, which might be misunderstood. It might be more sensible to ask a full set of questions, then judge on the information of one line. This is the parallel approach to decision making, and this approach is the basis for decisions made by the program listed in Fig. 4.

There are several passages in the program in Fig. 4 which are identical to that in Fig. 3, to start with the small decision-making program in your computer, and delete or rewrite lines 230 to 830, and the knowledge base 2000 to 2420. There is a right modification required to the questions lines 800 to 1090 and to line 9020. The addresses in notes in lines 300 and 360 are, again, optional on the CL, and the string joining a separator in these lines should be replaced for a `%` if you are using a Spectrum.

DATA line 800 now contains

not only the title of our expert system but two numbers, the number of questions and answers in our system. These two numbers set up a delay timer (240) in line 240. This delay contains zeros and ones, corresponding to expected no and yes responses to each question for all 12 modes of transport in our knowledge base.

All 12 questions are asked in response to the user reply a variable (A) is filled with zeros and/or ones (lines 360 and 362), to get a string of the same length as the strings in the knowledge base, one of all questions are answered. This reply string is then compared with the strings in the knowledge base.

Perfect Match

The first check (lines 330 to 340) is to determine whether a perfect match can be found, ie, a string of yes and no matches exactly the expert's opinion of the expected string of yes and no's in each of the 12 outcomes. If a match is found, then the expert's answer is given.

If no match is found, the computer first tests you, this goes through a routine (lines 420 to 800) to determine which set of expected yes/no replied in its knowledge base most closely matches the reply given by the user. It does this through each possible outcome, checking the expected response with the one given and assigns a score depending on the match. So, it is copied to the notes. If a "best answer" was given, when one is appropriate for that particular item and



question (line 400) and if it answered for a 'yes' reply when 'no' was expected in this way the item with the maximum 'yes' is identified (line 480).

Logic might suggest that you should also award a point for 'no' yes which match and subtract one from the score when 'no' is given to a question in which 'yes' is expected from a certain outcome. Well, the scoring system used does work admirably well, without these 'logical' additions. In fact, as these can for 'less yes' answers than 'no' in every thing in the knowledge base, matching 'yes' answers are more valid than matching 'no's'.

Another argument against this scoring system is that, for any outcome, certain questions are more important than others in differentiating items, so the amount added for 'yes/no's' should be weighted to reflect

this. That argument would lead us into the realm of probability which is something I shall save for later in the series.

One thing you'll note about parallel decision making is that a lot more questions will be asked, sometimes quite irrelevant when, in hindsight, you know the result. In the parallel decision program, you'll always be asked 10 questions, whereas, the serial decision program which takes the same problem data between two and five questions before giving its result.

No doubt, a better program expert that myself could have selected a better and smaller set of questions to enable the '10' items to be differentiated in the parallel program. But as a general rule, parallel systems will always ask more questions than serial systems, but even if the data matrix isn't perfect, the parallel system is more likely to

give you the right answer, one incorrect response with the serial system, and the expert's reply is certainly not to be the one you want.

These are expert system decision-making processes in their simplest form. Many expert systems allow more than 'yes/no' answers, decision making can be minutes of serial and parallel, depending on the problem. And these examples assume no uncertainty in your reply (or in the knowledge base).

These expert systems are also often limited in their knowledge, there are many more than 10 different modes of knowledge. The expert will have to learn a lot more about 'knowing' before a reply can begin to describe that as an expert. And how expert systems learn is the subject of the next part in this series.

Figure 4: Knowledge Listing for Parallel Decision Examination

10 OTHERS Simple Expert System	1100 OTHERS "Have I been loved at all?"
20 OTHERS Using parallel decision-making	1200 OTHERS "Have I been loved at all?"
30 OTHERS	1300 OTHERS "Is it what I want?"
40 OTHERS David Matthews	1400 OTHERS "Do I know how to be a doctor?"
50 OTHERS James Jones	1500 OTHERS "Do I know how to be a doctor?"
60 OTHERS	1600 OTHERS "Do I know how to be a doctor?"
70 OTHERS "Do I know how to be a doctor?"	1700 OTHERS "Do I know how to be a doctor?"
80 OTHERS "Do I know how to be a doctor?"	1800 OTHERS "Do I know how to be a doctor?"
90 OTHERS "Do I know how to be a doctor?"	1900 OTHERS "Do I know how to be a doctor?"
100 OTHERS "Do I know how to be a doctor?"	2000 OTHERS "Do I know how to be a doctor?"
110 OTHERS "Do I know how to be a doctor?"	2100 OTHERS "Do I know how to be a doctor?"
120 OTHERS "Do I know how to be a doctor?"	2200 OTHERS "Do I know how to be a doctor?"
130 OTHERS "Do I know how to be a doctor?"	2300 OTHERS "Do I know how to be a doctor?"
140 OTHERS "Do I know how to be a doctor?"	2400 OTHERS "Do I know how to be a doctor?"
150 OTHERS "Do I know how to be a doctor?"	2500 OTHERS "Do I know how to be a doctor?"
160 OTHERS "Do I know how to be a doctor?"	2600 OTHERS "Do I know how to be a doctor?"
170 OTHERS "Do I know how to be a doctor?"	2700 OTHERS "Do I know how to be a doctor?"
180 OTHERS "Do I know how to be a doctor?"	2800 OTHERS "Do I know how to be a doctor?"
190 OTHERS "Do I know how to be a doctor?"	2900 OTHERS "Do I know how to be a doctor?"
200 OTHERS "Do I know how to be a doctor?"	3000 OTHERS "Do I know how to be a doctor?"
210 OTHERS "Do I know how to be a doctor?"	3100 OTHERS "Do I know how to be a doctor?"
220 OTHERS "Do I know how to be a doctor?"	3200 OTHERS "Do I know how to be a doctor?"
230 OTHERS "Do I know how to be a doctor?"	3300 OTHERS "Do I know how to be a doctor?"
240 OTHERS "Do I know how to be a doctor?"	3400 OTHERS "Do I know how to be a doctor?"
250 OTHERS "Do I know how to be a doctor?"	3500 OTHERS "Do I know how to be a doctor?"
260 OTHERS "Do I know how to be a doctor?"	3600 OTHERS "Do I know how to be a doctor?"
270 OTHERS "Do I know how to be a doctor?"	3700 OTHERS "Do I know how to be a doctor?"
280 OTHERS "Do I know how to be a doctor?"	3800 OTHERS "Do I know how to be a doctor?"
290 OTHERS "Do I know how to be a doctor?"	3900 OTHERS "Do I know how to be a doctor?"
300 OTHERS "Do I know how to be a doctor?"	4000 OTHERS "Do I know how to be a doctor?"
310 OTHERS "Do I know how to be a doctor?"	4100 OTHERS "Do I know how to be a doctor?"
320 OTHERS "Do I know how to be a doctor?"	4200 OTHERS "Do I know how to be a doctor?"
330 OTHERS "Do I know how to be a doctor?"	4300 OTHERS "Do I know how to be a doctor?"
340 OTHERS "Do I know how to be a doctor?"	4400 OTHERS "Do I know how to be a doctor?"
350 OTHERS "Do I know how to be a doctor?"	4500 OTHERS "Do I know how to be a doctor?"
360 OTHERS "Do I know how to be a doctor?"	4600 OTHERS "Do I know how to be a doctor?"
370 OTHERS "Do I know how to be a doctor?"	4700 OTHERS "Do I know how to be a doctor?"
380 OTHERS "Do I know how to be a doctor?"	4800 OTHERS "Do I know how to be a doctor?"
390 OTHERS "Do I know how to be a doctor?"	4900 OTHERS "Do I know how to be a doctor?"
400 OTHERS "Do I know how to be a doctor?"	5000 OTHERS "Do I know how to be a doctor?"

Alan Davis explores the possibilities for printing graphics.

It's a recent issue of ZIG I write a short piece about the Remson C interface and Amstrad (and ZX) print drivers. I'd like to do it in this article to explore some of the graphics possibilities of the screen, and since the Amstrad is Epson-compatible the routines should work perfectly well with any Epson-type printer which is capable of small (24pin) line feeds.

Regular readers may be surprised to discover that the origin of what follows has nothing to do with adventure games of old (Zaxxon, Shock!), but the fact is that I also use the Spectrum for serious scientific work — and one of my requirements for this is to be able to get hard copy of graphs of fairly complex mathematical functions. Now if the graph you want to plot is fairly simple, there's no problem. You can just write a program to draw it on screen, and then COPY the result. But the screen resolution of 288x192 pixels is too coarse for the applications I have in mind, and so some way of getting higher resolution had to be found.

The approach I adopted was fairly obvious. Instead of drawing the "Y" axis of the graph vertically on the screen, I turned it through 90 degrees and drew it horizontally so that the "Y" axis becomes vertical (with a increasing downwards). The graph is then plotted eight points at a time (ie eight successive X values) to line 21 on the screen, the whole screen being scrolled upwards by one line between the plotting of one batch of eight points and the next. In this way you can get any resolution you like along the X-axis, if necessary plotting and scrolling many scientific or program data before the whole process is finished. Naturally, though, you'll never see the entire graph on screen at once, because the early parts will have scrolled off the top of the screen by the time the thing has finished.

What we need then, is a method of scrolling just a single line of graphics data (line 21 in this case) from the screen by the printer, so that we can dump it out continuously in

Exploring PRINTER GRAPHICS

between point plotting and screen scrolling, but sadly there's no single printer command for doing this. The Remson interface will give you a full screen copy that not just one line, and so a bit of programming is necessary to achieve this.

Bit images

The key to the problem lies in the "Bit Image Mode" printer commands. There's a lot of choice of these on the Amstrad machines, and in the end I settled for "double density" mode. This gives a printed dot

spacing which is half that obtained with "single density" mode — and therefore a printed image of higher quality and crispness. The relevant command is ESC L, = n1 + n2 translated into English this means that we need to send to the printer the codes CHR 27 (ESC), followed by two numbers: code n1 and n2. These last two codes tell the printer how many bytes of graphic data are to follow and together they form a two-byte number where n1 specifies the low byte, and n2 the high, so if you want to print say 256 bytes of data in

Listing 1

```

Page 1 errors: 00
                                10: 40H
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00000000 0000 0000 0000 0000 0000 0000 0000 0000
40 00000000 0000 0000 0000 0000 0000 0000 0000 0000
60 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80
000000 70 000 0000 0000 0000 0000 0000 0000
0000 80 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0000 90 0000 0000 0000 0000 0000 0000 0000 0000
0000 100 000 000 000 000 000 000 000 000 000 000 000
0000 110 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0000 120 0000 0000 0000 0000 0000 0000 0000 0000
0000 130 0000 0000 0000 0000 0000 0000 0000 0000
0000 140 0000 0000 0000 0000 0000 0000 0000 0000
0000 150 0000 0000 0000 0000 0000 0000 0000 0000
0000 160 0000 0000 0000 0000 0000 0000 0000 0000
0000 170 0000 0000 0000 0000 0000 0000 0000 0000
0000 180 0000 0000 0000 0000 0000 0000 0000 0000
0000 190 0000 0000 0000 0000 0000 0000 0000 0000
0000 200 0000 0000 0000 0000 0000 0000 0000 0000
0000 210 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0000 220 0000 0000 0000 0000 0000 0000 0000 0000
0000 230 0000 0000 0000 0000 0000 0000 0000 0000
0000 240 0000 0000 0000 0000 0000 0000 0000 0000
0000 250 0000 0000 0000 0000 0000 0000 0000 0000
0000 260 0000 0000 0000 0000 0000 0000 0000 0000
0000 270 0000 0000 0000 0000 0000 0000 0000 0000
0000 280 0000 0000 0000 0000 0000 0000 0000 0000
0000 290 0000 0000 0000 0000 0000 0000 0000 0000
0000 300 0000 0000 0000 0000 0000 0000 0000 0000
0000 310 0000 0000 0000 0000 0000 0000 0000 0000
0000 320 0000 0000 0000 0000 0000 0000 0000 0000
0000 330 0000 0000 0000 0000 0000 0000 0000 0000
0000 340 0000 0000 0000 0000 0000 0000 0000 0000
0000 350 0000 0000 0000 0000 0000 0000 0000 0000
0000 360 0000 0000 0000 0000 0000 0000 0000 0000
0000 370 0000 0000 0000 0000 0000 0000 0000 0000
0000 380 0000 0000 0000 0000 0000 0000 0000 0000
0000 390 0000 0000 0000 0000 0000 0000 0000 0000
0000 400 0000 0000 0000 0000 0000 0000 0000 0000
0000 410 0000 0000 0000 0000 0000 0000 0000 0000
0000 420 0000 0000 0000 0000 0000 0000 0000 0000
0000 430 0000 0000 0000 0000 0000 0000 0000 0000
0000 440 0000 0000 0000 0000 0000 0000 0000 0000
0000 450 0000 0000 0000 0000 0000 0000 0000 0000
0000 460 0000 0000 0000 0000 0000 0000 0000 0000
0000 470 0000 0000 0000 0000 0000 0000 0000 0000
0000 480 0000 0000 0000 0000 0000 0000 0000 0000
0000 490 0000 0000 0000 0000 0000 0000 0000 0000
0000 500 0000 0000 0000 0000 0000 0000 0000 0000
0000 510 0000 0000 0000 0000 0000 0000 0000 0000
0000 520 0000 0000 0000 0000 0000 0000 0000 0000
0000 530 0000 0000 0000 0000 0000 0000 0000 0000
0000 540 0000 0000 0000 0000 0000 0000 0000 0000
0000 550 0000 0000 0000 0000 0000 0000 0000 0000
0000 560 0000 0000 0000 0000 0000 0000 0000 0000
0000 570 0000 0000 0000 0000 0000 0000 0000 0000
0000 580 0000 0000 0000 0000 0000 0000 0000 0000
0000 590 0000 0000 0000 0000 0000 0000 0000 0000
0000 600 0000 0000 0000 0000 0000 0000 0000 0000
0000 610 0000 0000 0000 0000 0000 0000 0000 0000
0000 620 0000 0000 0000 0000 0000 0000 0000 0000
0000 630 0000 0000 0000 0000 0000 0000 0000 0000
0000 640 0000 0000 0000 0000 0000 0000 0000 0000
0000 650 0000 0000 0000 0000 0000 0000 0000 0000
0000 660 0000 0000 0000 0000 0000 0000 0000 0000
0000 670 0000 0000 0000 0000 0000 0000 0000 0000
0000 680 0000 0000 0000 0000 0000 0000 0000 0000
0000 690 0000 0000 0000 0000 0000 0000 0000 0000
0000 700 0000 0000 0000 0000 0000 0000 0000 0000
0000 710 0000 0000 0000 0000 0000 0000 0000 0000
0000 720 0000 0000 0000 0000 0000 0000 0000 0000
0000 730 0000 0000 0000 0000 0000 0000 0000 0000
0000 740 0000 0000 0000 0000 0000 0000 0000 0000
0000 750 0000 0000 0000 0000 0000 0000 0000 0000
0000 760 0000 0000 0000 0000 0000 0000 0000 0000
0000 770 0000 0000 0000 0000 0000 0000 0000 0000
0000 780 0000 0000 0000 0000 0000 0000 0000 0000
0000 790 0000 0000 0000 0000 0000 0000 0000 0000
0000 800 0000 0000 0000 0000 0000 0000 0000 0000
0000 810 0000 0000 0000 0000 0000 0000 0000 0000
0000 820 0000 0000 0000 0000 0000 0000 0000 0000
0000 830 0000 0000 0000 0000 0000 0000 0000 0000
0000 840 0000 0000 0000 0000 0000 0000 0000 0000
0000 850 0000 0000 0000 0000 0000 0000 0000 0000
0000 860 0000 0000 0000 0000 0000 0000 0000 0000
0000 870 0000 0000 0000 0000 0000 0000 0000 0000
0000 880 0000 0000 0000 0000 0000 0000 0000 0000
0000 890 0000 0000 0000 0000 0000 0000 0000 0000
0000 900 0000 0000 0000 0000 0000 0000 0000 0000
0000 910 0000 0000 0000 0000 0000 0000 0000 0000
0000 920 0000 0000 0000 0000 0000 0000 0000 0000
0000 930 0000 0000 0000 0000 0000 0000 0000 0000
0000 940 0000 0000 0000 0000 0000 0000 0000 0000
0000 950 0000 0000 0000 0000 0000 0000 0000 0000
0000 960 0000 0000 0000 0000 0000 0000 0000 0000
0000 970 0000 0000 0000 0000 0000 0000 0000 0000
0000 980 0000 0000 0000 0000 0000 0000 0000 0000
0000 990 0000 0000 0000 0000 0000 0000 0000 0000
0000 1000 0000 0000 0000 0000 0000 0000 0000 0000
0000 1010 0000 0000 0000 0000 0000 0000 0000 0000
0000 1020 0000 0000 0000 0000 0000 0000 0000 0000
0000 1030 0000 0000 0000 0000 0000 0000 0000 0000
0000 1040 0000 0000 0000 0000 0000 0000 0000 0000
0000 1050 0000 0000 0000 0000 0000 0000 0000 0000
0000 1060 0000 0000 0000 0000 0000 0000 0000 0000
0000 1070 0000 0000 0000 0000 0000 0000 0000 0000
0000 1080 0000 0000 0000 0000 0000 0000 0000 0000
0000 1090 0000 0000 0000 0000 0000 0000 0000 0000
0000 1100 0000 0000 0000 0000 0000 0000 0000 0000
0000 1110 0000 0000 0000 0000 0000 0000 0000 0000
0000 1120 0000 0000 0000 0000 0000 0000 0000 0000
0000 1130 0000 0000 0000 0000 0000 0000 0000 0000
0000 1140 0000 0000 0000 0000 0000 0000 0000 0000
0000 1150 0000 0000 0000 0000 0000 0000 0000 0000
0000 1160 0000 0000 0000 0000 0000 0000 0000 0000
0000 1170 0000 0000 0000 0000 0000 0000 0000 0000
0000 1180 0000 0000 0000 0000 0000 0000 0000 0000
0000 1190 0000 0000 0000 0000 0000 0000 0000 0000
0000 1200 0000 0000 0000 0000 0000 0000 0000 0000
0000 1210 0000 0000 0000 0000 0000 0000 0000 0000
0000 1220 0000 0000 0000 0000 0000 0000 0000 0000
0000 1230 0000 0000 0000 0000 0000 0000 0000 0000
0000 1240 0000 0000 0000 0000 0000 0000 0000 0000
0000 1250 0000 0000 0000 0000 0000 0000 0000 0000
0000 1260 0000 0000 0000 0000 0000 0000 0000 0000
0000 1270 0000 0000 0000 0000 0000 0000 0000 0000
0000 1280 0000 0000 0000 0000 0000 0000 0000 0000
0000 1290 0000 0000 0000 0000 0000 0000 0000 0000
0000 1300 0000 0000 0000 0000 0000 0000 0000 0000
0000 1310 0000 0000 0000 0000 0000 0000 0000 0000
0000 1320 0000 0000 0000 0000 0000 0000 0000 0000
0000 1330 0000 0000 0000 0000 0000 0000 0000 0000
0000 1340 0000 0000 0000 0000 0000 0000 0000 0000
0000 1350 0000 0000 0000 0000 0000 0000 0000 0000
0000 1360 0000 0000 0000 0000 0000 0000 0000 0000
0000 1370 0000 0000 0000 0000 0000 0000 0000 0000
0000 1380 0000 0000 0000 0000 0000 0000 0000 0000
0000 1390 0000 0000 0000 0000 0000 0000 0000 0000
0000 1400 0000 0000 0000 0000 0000 0000 0000 0000
0000 1410 0000 0000 0000 0000 0000 0000 0000 0000
0000 1420 0000 0000 0000 0000 0000 0000 0000 0000
0000 1430 0000 0000 0000 0000 0000 0000 0000 0000
0000 1440 0000 0000 0000 0000 0000 0000 0000 0000
0000 1450 0000 0000 0000 0000 0000 0000 0000 0000
0000 1460 0000 0000 0000 0000 0000 0000 0000 0000
0000 1470 0000 0000 0000 0000 0000 0000 0000 0000
0000 1480 0000 0000 0000 0000 0000 0000 0000 0000
0000 1490 0000 0000 0000 0000 0000 0000 0000 0000
0000 1500 0000 0000 0000 0000 0000 0000 0000 0000
0000 1510 0000 0000 0000 0000 0000 0000 0000 0000
0000 1520 0000 0000 0000 0000 0000 0000 0000 0000
0000 1530 0000 0000 0000 0000 0000 0000 0000 0000
0000 1540 0000 0000 0000 0000 0000 0000 0000 0000
0000 1550 0000 0000 0000 0000 0000 0000 0000 0000
0000 1560 0000 0000 0000 0000 0000 0000 0000 0000
0000 1570 0000 0000 0000 0000 0000 0000 0000 0000
0000 1580 0000 0000 0000 0000 0000 0000 0000 0000
0000 1590 0000 0000 0000 0000 0000 0000 0000 0000
0000 1600 0000 0000 0000 0000 0000 0000 0000 0000
0000 1610 0000 0000 0000 0000 0000 0000 0000 0000
0000 1620 0000 0000 0000 0000 0000 0000 0000 0000
0000 1630 0000 0000 0000 0000 0000 0000 0000 0000
0000 1640 0000 0000 0000 0000 0000 0000 0000 0000
0000 1650 0000 0000 0000 0000 0000 0000 0000 0000
0000 1660 0000 0000 0000 0000 0000 0000 0000 0000
0000 1670 0000 0000 0000 0000 0000 0000 0000 0000
0000 1680 0000 0000 0000 0000 0000 0000 0000 0000
0000 1690 0000 0000 0000 0000 0000 0000 0000 0000
0000 1700 0000 0000 0000 0000 0000 0000 0000 0000
0000 1710 0000 0000 0000 0000 0000 0000 0000 0000
0000 1720 0000 0000 0000 0000 0000 0000 0000 0000
0000 1730 0000 0000 0000 0000 0000 0000 0000 0000
0000 1740 0000 0000 0000 0000 0000 0000 0000 0000
0000 1750 0000 0000 0000 0000 0000 0000 0000 0000
0000 1760 0000 0000 0000 0000 0000 0000 0000 0000
0000 1770 0000 0000 0000 0000 0000 0000 0000 0000
0000 1780 0000 0000 0000 0000 0000 0000 0000 0000
0000 1790 0000 0000 0000 0000 0000 0000 0000 0000
0000 1800 0000 0000 0000 0000 0000 0000 0000 0000
0000 1810 0000 0000 0000 0000 0000 0000 0000 0000
0000 1820 0000 0000 0000 0000 0000 0000 0000 0000
0000 1830 0000 0000 0000 0000 0000 0000 0000 0000
0000 1840 0000 0000 0000 0000 0000 0000 0000 0000
0000 1850 0000 0000 0000 0000 0000 0000 0000 0000
0000 1860 0000 0000 0000 0000 0000 0000 0000 0000
0000 1870 0000 0000 0000 0000 0000 0000 0000 0000
0000 1880 0000 0000 0000 0000 0000 0000 0000 0000
0000 1890 0000 0000 0000 0000 0000 0000 0000 0000
0000 1900 0000 0000 0000 0000 0000 0000 0000 0000
0000 1910 0000 0000 0000 0000 0000 0000 0000 0000
0000 1920 0000 0000 0000 0000 0000 0000 0000 0000
0000 1930 0000 0000 0000 0000 0000 0000 0000 0000
0000 1940 0000 0000 0000 0000 0000 0000 0000 0000
0000 1950 0000 0000 0000 0000 0000 0000 0000 0000
0000 1960 0000 0000 0000 0000 0000 0000 0000 0000
0000 1970 0000 0000 0000 0000 0000 0000 0000 0000
0000 1980 0000 0000 0000 0000 0000 0000 0000 0000
0000 1990 0000 0000 0000 0000 0000 0000 0000 0000
0000 2000 0000 0000 0000 0000 0000 0000 0000 0000
0000 2010 0000 0000 0000 0000 0000 0000 0000 0000
0000 2020 0000 0000 0000 0000 0000 0000 0000 0000
0000 2030 0000 0000 0000 0000 0000 0000 0000 0000
0000 2040 0000 0000 0000 0000 0000 0000 0000 0000
0000 2050 0000 0000 0000 0000 0000 0000 0000 0000
0000 2060 0000 0000 0000 0000 0000 0000 0000 0000
0000 2070 0000 0000 0000 0000 0000 0000 0000 0000
0000 2080 0000 0000 0000 0000 0000 0000 0000 0000
0000 2090 0000 0000 0000 0000 0000 0000 0000 0000
0000 2100 0000 0000 0000 0000 0000 0000 0000 0000
0000 2110 0000 0000 0000 0000 0000 0000 0000 0000
0000 2120 0000 0000 0000 0000 0000 0000 0000 0000
0000 2130 0000 0000 0000 0000 0000 0000 0000 0000
0000 2140 0000 0000 0000 0000 0000 0000 0000 0000
0000 2150 0000 0000 0000 0000 0000 0000 0000 0000
0000 2160 0000 0000 0000 0000 0000 0000 0000 0000
0000 2170 0000 0000 0000 0000 0000 0000 0000 0000
0000 2180 0000 0000 0000 0000 0000 0000 0000 0000
0000 2190 0000 0000 0000 0000 0000 0000 0000 0000
0000 2200 0000 0000 0000 0000 0000 0000 0000 0000
0000 2210 0000 0000 0000 0000 0000 0000 0000 0000
0000 2220 0000 0000 0000 0000 0000 0000 0000 0000
0000 2230 0000 0000 0000 0000 0000 0000 0000 0000
0000 2240 0000 0000 0000 0000 0000 0000 0000 0000
0000 2250 0000 0000 0000 0000 0000 0000 0000 0000
0000 2260 0000 0000 0000 0000 0000 0000 0000 0000
0000 2270 0000 0000 0000 0000 0000 0000 0000 0000
0000 2280 0000 0000 0000 0000 0000 0000 0000 0000
0000 2290 0000 0000 0000 000
```


Listing 3

Page 3 error: 88

```

20 GO
25 .BASIC for printing a vertical strip of screen &
40 screen wrap, starting at the pixel value stored
50 in left in address 20000
60
80000 70 000 00000 (start of 800 rows
80004 80 10 0,0
80008 90 0000 00000 (address printer
80012 100 000 0
80016 110 10 0,0
80020 120 10 0,0 (use line 3 and 4 registers
80024 130 10 2,400000
80028 140 LOOP1 10 2,4 (starting position is 0 register
80032 150 LOOP2 10 000 00 (use registers
80036 160 0000 00000 (position to POINT to,0
80040 170 0000 00000 (start of 800 rows 2 register
80044 180 000 00
80048 190 000 00 (use registers
80052 200 00 0 (clear registers
80056 210 00 1,0 (use screen pixel only
80060 220 0000 0 (start of 800 rows 2 of 0
80064 230 000 0 (Position 0 register to use bit
80068 240 10 0,0 (use on next pixel on the right
80072 250 000 0,0 (1,000000)
80076 260 000 0,0
80080 270 00 0 (start on zero 0 pixels past
80084 280 00 00,0000 (if not 800 pixels then use
80088 290
80092 300 10 register use holds 2 byte ready to print
80096 310 10 0,0
80100 320 000 10 (no print in case
80104 330 10 0,0
80108 340 000 00 (if ... then use case
80112 350 10 2,0 (Clear 0 register
80116 360 000 0 (use on the screen by use pixel
80120 370 000 0,0
80124 380 00 0 (if ... then on top of the screen)
80128 390 00 00,0000 (if not then get screen area
80132 400 00 0000 (Back to BASIC)
80136 410 000 000000
80140 420 000 000 0,0 (Use line 7 of 0 register
80144 430 000 000
80148 440 00000 0000 0 (in case of left edge of strip
80152 450 0000 000 00000 (1,000)
80156 460 00 0,0 000 00000 (1,000)

```

Page 3 error: 88

Text: read 88 Page 208

Just point it in from BASIC when the program runs. Just a couple of further points about the listing: (a) You need to instruct your interface to turn off the tokens before proceeding, and line 40

does this for the Simpson 'T' interface. (b) Line 40 is needed only if your printer is normally set to give automatic line feeds with a carriage return. You may wonder why I've left

so much of the printer handling to BASIC, and there are two reasons for this. First, given the machine code routine to transfer the data is the actual physical process of printing which takes the bulk of the time, so that there's very little to be gained by writing the remaining bits in machine code. Second, with the subroutine as it stands you can fiddle about with it easily. You might, for example, decide to go for only two passes of the print-head rather than three in order to speed up the printing - and if you do you'll need to change the counter in line 200 (FOR 1 = 1 TO 1), and to change the line feed in line 230 to 22000.

Stripper

So much, then for high resolution graph plotting. Is there any other use for this idea of printing strips of screen in double density mode? There certainly is. How would you like a variable density screen display routine so that you can compensate for the wear on your printer ribbon? You would? Then read on.

The Simpson 'T' interface allows you to get two kinds of screen copy - an enlarged one which is fine for read purposes, and one of 'normal' size which is OK but could be better. This latter is a single density screen dump which looks rather grainy when your ribbon is worn with a little badly distorted aspect ratio to boot. If we take command of the printer ourselves, and scan the screen in strips (reading each strip much as we did in the graph plotting example), we can improve matters considerably.

There's some advantage to be gained by spinning the screen in vertical strips rather than horizontal ones in order to get a more appropriate aspect ratio, and at least it's only necessary to handle the transfer of data from screen to printer in machine code. Listing 4 gives the assembler program we used. Though similar to Listing 3, it differs in some important ways: namely (a) it transfers data from a vertical strip eight pixels wide and 600 pixels high, and (b) we need to fill it with strip to transfer by coating a outside a co-ordinate into 8424 (ADDRESS). Again, the code level is so short that its ready-to-go in form within a BASIC program, and I've incorporated it in this way into Listing 4.

Using 4 is a self-contained

Figure 3. Variable density screen copy



GREAT GRAPHICS

Here it is, the ZX guide to the best in graphics equipment for the Spectrum.

Art Studio £14.95

Unless you're going to stick to BASIC (not a good idea) the first thing you'll need when you start experimenting with graphics is a good bit of software, and, with one possible exception (Rainbird's Art Studio stands head and shoulders above everything else. Despite the somewhat Lenox anti-price device Art Studio is a fast and flexible load-driven package that will help you squeeze the best out of the Spectrum's graphics capabilities. It's also available in two versions for both 48K and 128K machines (although the 128 version costs £24.95).

The Artist II £14.95

Soltek's original Artist program was good, but it felt hefty to use, but the enhanced Artist II has adopted the same pull-down menu system as Art Studio and there's now not a lot to choose between the two programs when it comes to creating screen graphics. However The Artist allows you to take text files from Soltek's Writer wordprocessor and add graphics to them in order to create a possible imitation of a magazine page — desktop publishing on the Spectrum!

Cheeky Moch II £14.95

Once you've plotted your software you then have to decide how you're going to draw your pictures on the screen. All graphics packages allow you to use the keyboard for positioning a drawing cursor but this can get a bit tedious unless you've got the fingers of a concert pianist so a joystick can come in handy. Peoples choice of joystick is fairly personal, but we've found this one to be good for both delicate drawing movements and all out alien zigging, and it has the additional advantage of having an adaptor that allows you to

use it with both Komputer Interceptor and the +2's built in joystick socket. Of course if you don't have a Plus Two you'll need:

Komputer Joystick Interface £6.95

Almost as old as the Spectrum itself is the Komputer Joystick Interface, one of the de facto standards for the Spectrum. This latest model from Komputer is small, neat and very inexpensive.

AMX Mouse £69.95

Everyone's got a joystick (even if they're!) but the most ambitious among you might want to splash out on a mouse. Most 'real'

business computers have used it for 10 years, and there are now a couple available for the Spectrum (with another on the way from Edgo quite soon). The AMX Mouse from Advanced Memory Systems, appeared shortly before Rainbird released Art Studio and, being incredibly versatile, Rainbird made Art Studio compatible with the mouse (though you don't have to buy Art Studio at all since the mouse comes complete with its own bundled graphics software, AMX Art). The mouse interface plugs into the Spectrum just like a joystick interface into which the mouse is plugged. A good value-for-money feature is that the mouse interface also includes a printer interface which could save you some £30 if you haven't already got one.

Trojan Light Pen £14.95

Not as flexible as a mouse but less expensive, light pens have the virtue of allowing you to draw directly onto the surface of your monitor or TV screen. This can be useful if you're trying to draw curves, jagged lines etc. which can't easily be drawn using keyboard or joystick control. In the pen allows you to draw freehand. Trojan's pen



comes with its own graphics software, though this isn't as sophisticated as it could be.

Kempston Mouse £49.95

The price of this is about to come down to around £49.95 according to Kempston, making it excellent value. At the moment the mouse comes bundled with a copy of the Art Studio, making it almost as powerful a graphics tool as you're likely to get on the Spectrum (though when the price comes down Kempston are intending to replace Art Studio with their own graphics software).

The mouse itself is a sleek white unit with two buttons set into its 'hood'. It's very solidly constructed, and wouldn't look out of place beside a computer in times as powerful as a Spectrum. Like the AMX mouse, it also plugs into the Spectrum via its own interface which, not surprisingly, looks just like the rest of the extensive range of interfaces.

Kempston T² Interface

£34.95

When there's an interface to be found to be a Kempston, and the Kempston T² comes highly recommended by ZX regular

Alan Davis. All the software you need to run it with is held onboard, it's versatile enough to meet just about all your needs, and, most importantly, it's simple enough to use that you don't have to be a genius in order to wade through endless lists of control codes — unlike a lot of printer interfaces this one is actually 'user-friendly'!

Euroelectronics

ZXLprint III

£34.95

We had to give this one a mention since we've had one in our office for a couple of years (and never had a moment's trouble with it (this article was dumped onto our printer with it, it's compatible with both Centronics and IBM-style printers you just have to buy the relevant cables), and like the Kempston T² it has its own onboard software.

What about a printer?

Oh dear, this is where it could start to get a bit complicated. Once upon a time you had a choice between a big expensive 'real' printer or a number of small cheap 'dedicated' printers which were specifically intended for use with the Spectrum (Sindrain's own ZX

prints the Alphacom 32 and Sinclair SP900). Unfortunately none of these are still in production, though you might be able to find one or two still floating around in shops.

So you no longer have any choice — if you want to print out any of your graphics, mate-patched, you're going to have to look out somewhere in the region of £200 for a full-size printer and interface.

A printer is a fairly tedious piece of hardware and, in general, the price reflects the quality of the machine, although the Spectrum owners are likely to need to typewrite quality print offered by the top-range resolutions (going 2400 upwards). If you're only interested in printing screen dumps you probably don't need near letter quality printing, but it's not going to cost you much more and it's bound to come in useful one day — especially if you ever want to sell your printer on second hand, so bear that in mind! However, there are a few printers that offer high quality at relatively low prices, our particular favourites being

Aminrad BMP2000 £159

About as cheap as you're likely to get, and offering surprisingly good quality (an IBM-style product they've kept it simple and cheap, so it's looks a bit like a fully computered to some of the better numbers around) but it will do screen dumps and near letter quality text printing — and for under £200 that's good going.

Compliment LB £199

Part of Sargent's Compliment series, printer package this is an excellent resolution that's as good as some much more highly priced printers. Obviously geared towards word-processing rather than graphics, but still capable of good quality screen dumps.

Citizen 1200 £235

Striving to get a little price, but this is about as good as most home computer owners will need. There is a very good letter quality mode for printing text, and it's provided with international character sets and self-diagnostic tests, and takes single sheets of paper or perforated printed paper. Some of these features are luxuries, rather than necessities, but if you can afford it you're not likely to have to buy another printer very soon.

GREAT GRAPHICS



Hot on the heels of The Advanced Art Studio comes Audiogenic's graphics package for the 128 and Plus 2.

Icon Graphix 128
Audiogenic
£12.95

Audiogenic's Icon Graphix package was launched about a year ago, at the same time as the Spectrum 128 appeared. The first version to appear was for the 48k machines but, not surprisingly, Audiogenic also announced that an enhanced version for the 128 was on its way and, now that Amstrad's own version of the 128 is well and truly here, they've finally gotten around to releasing Icon Graphix 128.

New features

Unlike the enhanced version of Art Studio, which uses the 128's additional memory mainly for fast data manipulation and which has relatively few new facilities for actually creating graphics, the new Icon Graphix package features a number of options which were absent from the original. Mind you, the original Icon Graphix lacked certain features that could be found in both Art Studio and the Amstrad (such as text handling and cut and paste commands) so the upgraded version has been brought into line with what we've all come to expect as standard from graphics packages these days.

The layout of the screen display remains unchanged, with the central drawing area surrounded by a border made up of the various F11 patterns and along the bottom of the screen, both symbols for the graphics commands. This menu now has an added feature though — a PAGE menu, which when selected scrolls up pages 2 and 3 of the menu containing the new commands.

The F11, F12, and shape commands for drawing boxes, lines, circles and arcs were all standard on the 48k version but the new commands include some for adding text to your pictures (and there is an impressive variety of fonts included on disc) two of the tops, although only one font of

ICON GRAPHIX 128



it this can be held in memory!

The other new commands are mainly for 'cut' 'n' paste, allowing you to define small sections of the screen which can be 'lifted up', moved around, rotated, re-sized and otherwise altered. As I mentioned these can be found on ordinary 48k packages, so while their inclusion now is a good idea it's not really a giant leap forward.

Trace

One new feature which is quite original is the TRACE command. Once you've defined a shape or a cutting, as the manual calls it, you can use this command on whatever is drawn within it. What it does is to draw

the paper and ink, with the cutting and to then transfer any solid areas with either a black or white outline, depending on the nature of the drawing. This might sound a little pointless but it actually creates a kind of 'poisoned' effect that makes even simple drawings look more interesting, and it can be used repeatedly to build up some unusual shading effects which might otherwise take longer to produce if they had to be drawn conventionally.

The TRACE command is the highlight of the package, and, going with the Icon menu (which I found simpler) to see how an Art Studio or The Artist is one of the ones where it scores over the competition. However



There are other features which could be imposed upon given the 128k potential.

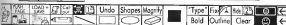
Although the program allows you to control the cursor from keyboard, joystick or (Kempston) mouse, the movement of the cursor is fairly slow and it can take a couple of seconds to move the cursor from one side of the screen to another (good if you're doing detailed work using the MAGNIFY option, but irritating if you're trying to quickly move from one icon to another).

The fill and shape drawing commands are also slow and it seems defining a window with the cut and paste commands that is more than about 1/3 of the size of the screen you can almost hear the poor machine straining to keep up as it tries to constantly update the window. Finally, the SUBSTORC option lets you do a bit of a mess. You can save or load a normal SCREEN, a document or a picture, though the manual doesn't really explain how a document differs from a SCREEN. The loading option is a bit tricky about choosing names and you don't just load the first SCREEN that you come to on tape (is using LOAD "" SCREEN) as you've got to know the name of the screen to start off with. This is fine if you've saved a picture down from another site the package, but if you've got some pictures saved onto tape that are taken from converters, then you've got to make sure you know their names before you can get at them. With Graphics 128 it's an improvement upon the original. I found it easier and more enjoyable to work with than many other graphics packages I've had to review in the past, mainly because the icon menu is so self-explanatory and visible on screen all the time rather than just popping up and down before you've managed to figure out what you're doing.

As I mentioned, the program could be better than it is, but on the other hand it does cost just £12.95 compared to a fully forty £25 for the 128 version of Art Studio, so while it might not match Art Studio in sheer speed and number of options it still works out of quite good value.



Above and below: Sir Clive before and after being 'fixed'



DISCIPLE

John Wase plugs in a new multi-purpose interface.

**Disciple
Rockford Products
£73.70 (+ VAT)**

After several weeks I had been anticipating its arrival. At last a cryptic message came — 'to Disciple will shortly arrive at your letter' — it was on its way!

My delay in receiving this interface was due to a technical problem: the NEC ROM version of the Spectrum gave trouble with these first issues and the ROM in the Disciple had to be re-written. Advantage was therefore taken of this to incorporate several improvements into the system and it is this version, version 2, which I tested. Disciple comes complete with instruction manual, software and a cassette. As the picture shows, the interface itself is a long, flat plastic box on which type Spectrum sits a bit interface One-ah in shape, with an edge connector on top to connect the Spectrum and the customary through-port on the back. To the right of this are the joystick port and network connector with the printer interface on the end of the box. To the left are a

standard NES-type disc interface connector, a serial printer port and network connector and on the left hand end the in-built serial expansion options.

I approached the system with some hesitation being particularly wary of any familiarity with the Cpu system. So at first I didn't like the rather untidy plastic box, or the Spectrum 2 overhang (to be fair a lot), and I was glad I had left on the big test which I tested with the Discovers, for they just supported in the right places the overhang from the Disciple. A standard 48k Spectrum was less lucky and needed a cassette temporarily switched in front of it. The first two Disciples had enough hanging in the mail and the original switch was damaged in each case (which led me that more robust switches are now being used) and that the packaging is being modified. What none of this impaired my confidence, as I found it did produce good use so if I began to like it did appreciate its many virtues.

Hooked

The manual contains a lot of useful basic information, but lacks anything detailed on machine code (though it mentions that the hook codes are the same as those for micro-ohms). I am not that far into the book, but there will be a nice, step-by-step version of the instructions which will have provision for containing comprehensive information on machine code. The first operation is to connect everything up and load

the utilities program supplied on cassette. After observing various prompts, you are ready to format a disc. Two supplies: first the disc is not formatted, you have to format it on the label, secondly a small code program called 'systems' is also saved on the disc at the same time. A cold start involves switching on computer, inserting disc and typing ROM when the system program is needed. This can be a bit irritating but don't stress that the system can be updated to cope with alterations in other systems. There is provision for one further sub-program on any one disc.

Disciple comes with a truly comprehensive array of commands, **LOAD** (to program), for instance, works, as well as micro-ohm commands, so does **SAVE** (to program), **SAVE** (to program) save disc then also gives you the short directory or catalogue which numbers each file tells you what it is and the start address and length if it is code. This allows you to use the very short syntax **LOAD p**, when number *p* in the file is loaded from the 1st disc in use. All the commands one would expect are there, including wild card files (to disk systems), numbers, numbers *i* etc, enter **DATA** (to *i*). A well thought out release.

In addition to the disc interface itself, one or two other files (games) are incorporated. Clearly aimed at the games player is a special option, this takes a copy of what it is necessary and, in the case, gives it direct to disc as a 'special file'. This is the only type of file



which will not copy from one disc to another of the two disc system, thus piracy is minimized. The two floppy ports can be configured either as Sinclair or left hand only as Kensington.

A great deal of thought has been given to networking. Disciple has two network ports, each taking inexpensive standard 3.5mm jack plugs and standard cables. In this way, a shared desktop network can be set up with a master station and pupil stations. The master can transmit programs to the pupils and even call up their screens to check their work; they, in turn can call up programs from a central disc or print on a central printer.

Centronics

The centronics interface is fairly straightforward. The controls to the left of the program involve information about the printer. In an I/O it is not an Epson compatible. The printers are incorporated in the systems program which has to be used to load at the system, as instance if you are using one, make it for rough copy and dotry speed for the finished document. LPT and LPR are supported, together with COPY SCREENS which copies the screen. Any control codes that you want to send to

the printer are provided by CH0227, even if you wish to send CH0227 itself, but you can but effective. This can, however, be changed. It has to be for forward, for instance, for reason 2 will forward only four control codes at any one time.

Now for the messy bits. I took one of my own games and converted it onto disc. This consisted of a basic loader, a screen and a piece of machine code. 30000 bytes long (initially a full 48K Spectrum).

Loading this from tape took just about all five minutes, the same thing took a spectropipe seven seconds from the Disciple. I loaded the game again and took a snapshot, taking it back between eight and nine seconds. Reloading took five

thirty times get as short as this it is impossible to be very accurate, all you can say is that they are impressively fast. Saving is equally fast — I used the LPT 01 "format" screen for the picture and for the code, in each case the bulk of the screen or eight seconds was the time required to display the directory. Indeed, the only operation which took longer was formatting.

Timing of formatting, I found that this was one of the interesting comparisons to be made. Naturally has a directory length (20 files) which is

independent of the disc type, and which can be extended if necessary. Disciple allows only 40 entries in single density and 60 in double. The sector length is similarly interesting. Disciplex and Disciple 40 track has 40 byte sectors. Although these are bigger sectors and the lack of a compaction facility on Disciple could mean more wasteful use of the disc, this is made up for a larger formatted capacity, although do remember that you have to put the "system" program on the disc.

My overall impressions and comments? Well, for the boot news, it's going to cost quite a lot for a complete system, but you've got the cost of a drive on top of the CPU odd for Disciple. I tell it was a bit pricey, a bit firm, and I managed to get the disc controller in the wrong way round. Much of this was when, all things, I was trying to connect up and get it going having got used to it, well, it's great there are all sorts of additions, the snapshot, the comprehensive networking, the compatibility of two Sinclair or one Kensington interface ports, the ability to run interactive software and to have it transferred by means of the initial button, and above all, the incredible speed of transfer. You know, I often enjoy one aspect.

ARE YOU A BUDDING PROGRAMMER?

ZX is always looking for top quality games and utilities for publication. If you have a top notch game or a useful utility for the Spectrum or QL, why not send it to us for appraisal on a cassette or microdrive compatible with a listing if possible.

Then, if we like our new Spectrum Short Cuts to showcase your product, now it can be interactive, short routines with cash prizes for published utilities. For longer programs we pay competitive rates, and if you have an idea for an article or series for ZX — drop us a line or phone Bryan or Cliff on 01-437 0526 to talk it over.





Mini Studio

A Sound Sampler for any 48K or 128K Spectrum by Massimo Pilla.

To enter the program type in listing 1 and save it with SAVE "MINI STUDIO S". Then type in listing 2 and save this file under another name with SAVE "L1". Run listing 2 and when the saving process operates, position your HEAD tape (with H1 on it) just after listing 1 and start recording. Press a key below the tape and load listing 1. Once it has loaded and run itself you are ready to go through the following instructions.

What is Sound Sampling?

Sampling is a way of digitally recording a small amount of sound, ready for use in music. A computer's memory is an ideal medium for this type of recording, since a sound is in memory it can be subjected to any number of bizarre operations.

This program allows sound (up to about six seconds long) to be recorded, reversed, speeded up, slowed down and resequenced. played in a specific order with other sounds, the basis of the program is the machine code that records and plays back the sound. This works by recording the cassette part at high speed, and feeding the sounds that are sent into memory in the form of data bits. By reversing the process and reading the bits from memory and out of the cassette part and buzzer, the sound is reproduced. Changing the speed that the sound is played back of will also change the pitch of the sound.

There are four modes of operation:

SOUNDS — this is the mode where the sounds are recorded and cut up to use in the other modes.

PLAY — sounds set up in SOUNDS mode can be played back at eight pitches covering an octave.

BACK — up to eight bits, each

containing up to eight sounds, can be altered in this mode.

PROGS — bass created in SOUNDS mode are strung together in up to 64 steps here.

The SPACE key switches you between modes and the cursor in SOUNDS, BACK and PROGS are controlled using keys: 0-left, 2-down, 7-up and 8-right with 3 for edit.

Sounds Mode

In this mode the program is controlled with the following keys:

- 1 — Selects the number of beats per bar (up to eight)
- 2 — Selects the 'blocks' per beat (see note below)
- 3 — Input (record) a sound
- 4 — Output (playback)
- 5 — Load
- 6 — Save
- 8 — Key (musical key in PLAY mode)

When using 1, 2 and 3 they operate off the sound undetermined by the cursor, this is the meeting time which undetermines various functions and sounds and is controlled by the keys 6, 7, 8, to move 6 in the direction of the arrow.

The memory that sounds are compressed into is divided into 64 'blocks', each 256 bytes long. There are ten sound configurations (0-9) which are simple windows on this memory.

Each of these configurations has four values as shown on the top left of the screen. The top right displays the reader you are in. The meaning of these values is:

BA — this is the 'block' where the sound starts in memory

(0-15)

LA — the length of the sound

(0-120)

DA — the direction of the sound (forward/backwards)

DE — the delay used when playing back or recording the sound. This determines the speed and quality.

Another value — BE — is also shown but it just the number of beats the sound occupies. This is calculated by $(LA/DE)/BA$.

Recording

- 1 Set up the computer and tape recorder as shown in figure 1.
- 2 On most tape recorders pressing PAUSE+PLAYBACK should allow you to sample from the microphone.
- 3 Continue a sound so as to use up the rest of recording memory (64 cut up into 0-63).

by moving the cursor with keys 6-8 until it is beneath the value you wish to change, press 0 and then enter the value you require.

The delay is set to 01 for maximum quality. If the delay was higher the sound could then be replayed at higher pitches (ie. lower delay) but would be of poorer quality.

3 Press 1 for input and reply "Y" to the "use?" prompt.

4 Press a key to start sampling when you are ready. The sample is complete when the "PRESS A KEY" prompt disappears.

5 Press 0 to listen to your sample. It will sound best through headphones plugged into the ear socket at the back of the computer.

6 Now you must lighten up the sound, in short of any excess sound before or after the part you want. Start by increasing the start value (BA) till by 01, you will have to decrease the LA value before each increase of BA, as the (LA/DE) or $(BA+LA)$ is a maximum of 120. Listen to the sound each time until there is no superfluous sound before the portion you want.

Now decrease the length (DE) until the sample is exactly the portion of sound you require.

7 Your sample is now complete. You may reverse it by moving the cursor to the DE column and pressing 0. You may also change the delay (speed) if still within the various ways of using a sample in a moment.

8 Now decrease the length (DE) until the sample is exactly the portion of sound you require.

9 Your sample is now complete. You may reverse it by moving the cursor to the DE column and pressing 0. You may also change the delay (speed) if still within the various ways of using a sample in a moment.

10 Your sample is now complete. You may reverse it by moving the cursor to the DE column and pressing 0. You may also change the delay (speed) if still within the various ways of using a sample in a moment.

11 Your sample is now complete. You may reverse it by moving the cursor to the DE column and pressing 0. You may also change the delay (speed) if still within the various ways of using a sample in a moment.

Play Mode

This mode uses the following keys:

- 1 — Record
- 2 — Playback

In the Record sub-mode the speed is currently undetermined by the cursor. In SOUNDS it is turned into a musical key (not to

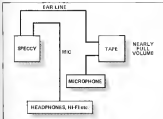


Figure 1 Sampling chain

musical instrument played on keys 1-8 in the key specified by the R key from the SOUND mode.

0 toggles recording of tunes SOUND Y exits the sub-mode. In the Playback sub-mode if a tune recorded in Record can be re-played by tapping on the rhythm on the Q key Y exits.

Ears Mode

This mode uses the following keys:

- Q - Run/initiatives (beats)
- R - Toggle Stop/Record
- O - Toggle beat counter ON/OFF

Hold down any key on the bottom row to stop the sequencer looping endlessly. This happens when repeat is ON.

When the beat counter is ON, any bar entered that does not contain the right number of beats (i.e. when the total BEATS of every sound) in the bar is not the same as SOUND) will be selected. This bar must then either be entered correctly or deleted by just pressing DELETE.

In this mode eight bars (list of sounds) can be edited and saved. These can be played back in order in this mode or in a more complicated sequence can be used.

Editing bars

- 1 Move the cursor under the bar to be edited and press R.
- 2 Type in your bar (eg. 11134112) the numbers are the numbers of the sound samples that you set up in SOUND mode.
- 3 Enter any other bars you want to enter.
- 4 Press Q and then enter "Y" to initiate.
- 5 Press Q and then "Y" for RUN to listen to your choice of musical genre.

The example bar in step 2 will play sounds 1,1,1,3,4,1,2 in that order. When counting the beats it counts sound 1 four times and 3 twice. You can walk into eight parts, 1 comes to know the counter OFF but it is there for a purpose.

Prog Mode

This mode is controlled by keys Q and R.

- Q - Run/initiatives (sequence)
- R - Toggle SOUND/EAR

Hold down any key on the bottom row to stop a sequence from playing.

Bars can be strung together in up to 84 steps. This is for longer more complicated sequences and for quicker editing.

Editing steps

- 1 Position the cursor under the step to be edited.
- 2 Press R to edit and enter the number of the bar you want.
- 3 Enter Q to signify the end of the sequence.
- 3 Enter any other steps you want to enter.
- 4 Press Q and then Y to initiate. Press Q and then "Y" to listen to the sequence.

Whenever an alteration is made in either SOUND or EAR mode remember to re-initialise them.

Using a Sample

Once a sample has been created several things can be done with it.

- The sample can be used as a musical instrument in PLAY mode.
- The sound can be put into a similar sequence on its own. If you have identified some speech as 3-003-016 PD-01 then you could set up sounds 5 and 2 as similar consists of 0-1-1-003-003-PD-01 3-003-004-PD-01 3-003-004-PD-01

In EAR mode enter bar 1 as 211220 initiate and run R for a N-N-N-N-N effect.

3. Echo can be simulated by "layering" the sound. Set up the sounds like this:

- N - 014 - 12N - 01 - 01
- 0 - 000 - 000 - PD - 01
- 1 - 002 - 004 - PD - 01
- 2 - 004 - 004 - PD - 01
- 3 - 004 - 004 - PD - 01
- 4 - 008 - 004 - PD - 01
- 5 - 000 - 004 - PD - 01
- 6 - 002 - 004 - PD - 01
- 7 - 004 - 004 - PD - 01

Enter bar 1 as 0034667 initiate and run R.

4. The sound may also be used as part of a more complicated sequence using other sounds (ie a drum beat).

Sequencing a Tune

Sequencing a tune poses a problem because when the sounds pitch is changed its length in time is also changed. This means that the rhythm goes out the window. As the program SOUND leave a timer counter method has to be used.

Set up sounds 0-7 like:

- N - 014 - 12N - 01 - 01
- 0 - 000 - 000 - PD - 01
- 1 - 008 - 004 - PD - 01
- 2 - 004 - 004 - PD - 14
- 3 - 004 - 004 - PD - 30
- 4 - 002 - 004 - PD - 30
- 5 - 000 - 004 - PD - 30
- 6 - 002 - 004 - PD - 42
- 7 - 004 - 004 - PD - 48

Sample the same sound (presumably a recorded sound) into each delay. Now change all delays back to 00.

These delay settings are in the key of Q (if a guard will represent the note). Use R, A, B and C. Add 4 to a delay for a bar and subtract 4 for a step.

Example Rhythm

If you managed to take all that in you are ready for your first drum rhythm using MIN-SOUND.

- 1 Enter SOUND mode.
- 2 Set up sounds 1-3 like this:

- 1-0000 PD-01
 - 2-003 003-PD-01
 - 3-000 003-PD-01
- 3 Input or load a bass drum sound into 1.
 - 4 Input or load a snare drum sound into 2.
 - 5 Input a blank (silence) into 3.
 - 6 Enter EAR mode.
 - 7 Set the beat counter off (press C).
 - 8 Enter the following bars:
 - 00000 00000 00000 00000
 - 00000 111111 111111 111111
 - 9 Initiaize
 - 10 Enter prog mode and enter these bar delay:
 - 1 2 3 4 5 5 3 3 2 3 6 0 7 3
 - 11 Initiaize and run EAR.

ABSORBER THE GREAT!

COMPETITION



There are thirty copies of Morloch's Cosmic Shock Absorber to be won.



Cosmic Shock Absorber is a would be Superhero with aspirations to join the Supermen and Batman of this world. Unfortunately he must save his apprenticeship by taking on foes such as rampaging robots armed with machine guns and legions of bouncing balls. Cosmic Shock Absorber may be lacking super-strength and super-abilities, but with your help he can make it into the Superhero class.

There are thirty copies of Morloch's awesome 3D shoot 'em up to be won, and all you have to do is name the superheroes who fought the following three adversaries.

of The Penguin
of Ice Luthor
of Imperial Ming

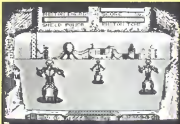
Just put the names of the various victorious superheroes against the appropriate letter on the entry coupon.

The competition is open to all ZX readers except employees of Argus Specialist Publications, Chase Web and Morloch. The editor's decision is final and no correspondence can be entered into.

Send your entries to Cosmic Shock Absorber Competition, ZX Computing Monthly, No. 1 Golden Square, London W1R 3AE.

Please remember to put your name on your entry envelope.

The closing date for entries is April 19th.



Cosmic Shock Absorber Competition

The vanquishing superheroes are

- a)
- b)
- c)

Name

Address

Send your entry to Cosmic Shock Absorber Competition, ZX Computing, No. 1 Golden Square, London W1R 3AE.

GAMES

KONAMI'S GOLF

Konami's Golf is played over a course of the BBC - the Konami Country Club. The entire green hole hides the foot foot - all we have here are nine holes and a computer but unlimited golf simulation.

All the GOLF features we expect to find in a golf game: you have options for one player and two players (single and multiplayer), a variety of clubs from a bag of 15, an animated golfer which hits the ball off the tee and a computer to adjust the length of shot.

The display design attempts to cram everything into a single screen. The default view of the course on which you choose the direction of the ball shot is just as well. Here we can see it is a relatively conventional looking game even if the main bottom eye view screen display

There are five buttons to take into account when setting up your shot. You can choose to look back at the target and the green, or the green. The display shifts to an over head view and options such as slope and speed of the green come into play. These are really set by a single slider bar and don't really add to the difficulty.

Konami's Golf is only worth shelling out money for if you are desperate to get a Golf simulation into your software collection but with any alternatives, it's a good idea to get out very quickly.

GOOD



you that it looked a bit dodgy but was still fun to play.

So, some is true of Lunar Jetman. The sequel to Jetman this year (Jetman's last add game) is a bit more interesting. Though it's still essentially a shoot 'em up, this time around the Jetman has a good bugby to control, at least as a number of objects to collect and others to avoid. But what the Jetman games have remained interesting over the years, the other two bits on the tape have done better.

Good is a variation on the shoot 'em up style. It's what you have to spray enemies with a number of spray guns, while from Am you can only bring some. Rather than in these games

really stand up city marks being about the same standard as the Demolition budget game.

So, while the two Jetman titles are a tribute to the quality of early 16-bit games, this competition of which class really stand out when compared to some of the other titles in the other two tapes arrived (I think) Computer-generated characters in this level is worth a closer look.

GOOD



LEAGUE CHALLENGE

At home strategy

This game lets you be the best of your league. It's a football manager type game so if the love of league tables and team selection problems holds no fascination, hold on tighter.

If you're stayed on you can be lulled into the routine of it's always the same - 20m this year. When League Challenge attempts to cool the big match atmosphere we are treated to some very odd graphics. Instead of having a simple line-up of the players on the pitch, we have a grid of 11x11 where each cell contains a player's name. The grid is a bit of a mess and it's a good thing the graphics decide your league and you

score.

You start off in the early days of Division 4, obviously and by buying players, training them and selling a strong team, you hopefully rise to play in the First Division.

Your team can be decidedly quirky games, often followed by a lot of wisdom after you'd bought and then you're to be expected to see League Challenge will certainly win no friends but if you can make the promotion, the time it takes to make your team to the top will justify the budget price.

GOOD



MASTERS

It's good to see that the competition class of old 16-bit games, both entitled Master Games and with the best looking playing to tell the tale (the title of the first console the first four games that available were produced) which were the first to be developed for the Spectrum (and it's memory

games, all but one of them had to under 16k, as they were produced in the days when the 16k Spectrum was still going strong).

Although very nice game, Jetman, a simple but highly addictive shoot 'em up in which you have to collect fuel pods and weapons on a number of planets, all of which are teeming with deadly aliens. At the time, Jetman was based and shrouded others, any other game for the new Spectrum machine, with smooth and well-written and good sound effects. I should Jetman to 16k, who didn't do it well. It had some old and their writing.



HENRY'S HOARD

Alternative Software
1988

Alternative Software is the latest company to start producing custom software, and so far that's a ticket for the Spectrum. How have these possible winners of pretty old arcade games, Henry's Hoard? Possibly the most often forgotten of games is a return to that old favourite, the pattern game that long it is case you've played one of these?

It is a fill screen game in which you have to avoid Henry in most spots with single legal off a board of forms of various confined around the various items. As is the way with these games there are lots of little bonus objects all around, moving sideways and steadily

moving to avoid off by resorting to the handish strategy of jumping and avoiding. As Henry's Hoard moves on it's fairly easy to see — the graphics are a little on the simple side but it's clear enough what's going on, and some of the obstacles require a bit of thought to get past.

Henry's Hoard isn't going to set the world alight but I've enjoyed Hoard a week spot for platform games. Henry's Hoard is a little more tedious than most, but it's a good idea to avoid Henry's Hoard.



MASTERS OF THE UNIVERSE

1988

When US Gold bought the license for this old TV series it was under the impression that it was a simple matter to take the basic format of the game, the main five games, the classic adventure and a host of other fun activities.

Master's Hoard is a game of the old TV series, it was under the impression that it was a simple matter to take the basic format of the game, the main five games, the classic adventure and a host of other fun activities.

Once you've had the game, you'll find it's a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

On screen, the main is represented by a little yellowed figure that is surrounded by a

black shadow that looks a bit like the middle of a shadow. However, you'll find it's a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

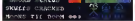
The way to the castle is a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

Inside the castle you are met by even more enemies that look like a bit of a disappointment. The game is a bit of a disappointment.

power. This is a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

powerful monsters in the game. This is a bit of a disappointment. The game is a bit of a disappointment.

Another great feature.



DEFGOM

1988

In the year 2050 it was a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

The way to the castle is a bit of a disappointment. The game is a bit of a disappointment. The game is a bit of a disappointment.

One man could save the world in the game. The game is a bit of a disappointment. The game is a bit of a disappointment.

new firing system.

Being the world's best going to be an easy job. The game is a bit of a disappointment. The game is a bit of a disappointment.

And that's the way to the castle. The game is a bit of a disappointment. The game is a bit of a disappointment.

A menu of options can be used to set the game. The game is a bit of a disappointment. The game is a bit of a disappointment.



and can be improved by the bonus option if you take a goodish 100 or a full or fivefold colour.

However, most of these options are little more than standard features in other games and contribute to games that take about an hour to play and take an age to play.



GRIM



XCEL Mastertronic £19.95

XCEL was originally released as a full price game about a year ago and it sported every set the world's got. Except it's a fraction less in worth & look &



you'll still shoot six ups. The game has a few basic and straightforward in the form of some levels, but there don't really shift the emphasis of the game from much other than straightforward alien blasting.

The action takes place on a grid of planets which are controlled by Gemini computers and fit up to six to try from planet to planet, blasting everything in sight in order to return control to the human

production. You do over the landing surface of the planets, avoiding mines and other obstacles in order to blast the computer control centres and after each bombing run (there's a G-galaxy's data index of attacking aliens).

The action isn't as fast and furious as other shoot em ups and at times the game looks a little played out. There are about 100 planetary locations to blast your way through to at least there's a fair bit of variety in the game. Mind you I also find it a little boring that when first you get blasted you have to go right back to the start of a particular sequence rather than carrying on from where you got killed, which means that you can find your self going over the same sequence over and over again.

Still, the game is nicely presented and at £19.95 is a much better value than when it was first released.



THE HIVE Creative £19.95

The Queen of the Hive must be slain and it's your job to do it in this extremely smart but in the same game. The action is in the middle of a three dimensional hive that spins and spins through eight complex and arranged levels.

You pilot a versatile electro-mechanical Grasshopper ship that runs, jumps and crawls

It slides through the maze. Just running through the maze is hard enough without the need to jump to make sure you don't run around in circles and back to the start. The action is really fast to get you those instant systems of stop levels, beams, glass, diamonds, pillars and spikes, which do things that block your path.

Contact with any of these will drain your energy that can be replenished in a few seconds. You must use your energy wisely that can be found in the maze to re-

place your original supply if you have only one grain then damage is not required (using them all to reach the level's end is your strategy to quickly drained away).

The screen display shows your view down the tubes and linkways that show your current energy level, score and any objects. Drones, aliens and supports you may face collected.

Linking underneath the 3D display is yet another maze game that has been collected there by the underload key controls that speed drawings.

Grasshopper 3 maze games there.

Finally should you want to save your aliens from you only have to drag them to code numbers that can be added when a new game is loaded.



GRIM



AGENT X Mastertronic £19.95

This latched out title is a previous surprise — another Mastertronic Challenge that opponents met with many of the expensive games that were omitted out for Christmas.

Agent X loads in several parts, each one of a different being in a different stage in the game, and as the game progresses you quite a lot into just 1000 worth of points. The game starts with a bonus the 1000 of Agent X was off in last part of the 1000 part. Part 2: The graphics in this sequence are excellent — a large and nicely detailed 3D representation of a highway with colour reds and blues in both directions and going in the way.

The action is a bit chaotic, and getting through the action is more luck than judgement.

until you get the hang of it, but it's all in the end because that's the way.

The next sections of the game take you into the next profession's role where you have to take on the security men getting an extremely nice-looking flying M4 machine, and then use comparison with the level itself to be able to finish you with anything that comes to hand.

It's all rather silly really but the graphics are good and the judgement is clearly kept. It's longer well and for there, making the experience's best budget game to watch.



A software 'legend'
from U.S. Gold?

Crystal Castles
U.S. Gold
£5.99

CRYSTAL CASTLES



game is a little more complicated. Bentley has the ability to jump over monsters rather than simply moving left/right forward/backward and it quite often happens that he will move behind a section of the 'castle' structure. When this happens the program allows you to 'see through' the structure so that you can follow Bentley's movements although you are unable to see the actual path that he is on.

The drawback with using 3D structures is that the graphics are not much more complicated and in order to get everything onto the screen the moving figures and blocks which make up the structures all have to be quite small. This makes the game quite fiddly to control. Most of the paths around the screen are so narrow that it's not at all unusual for Bentley to shoot past one when you actually want him to turn into it so you have to slowly double back and fire his up precisely, before going down the path. By this time, of course, every

monster on the screen has formed in on the happy face and until him to bear down. As a result of this fiddliness the games excitement is rather considerably instead of trying to do an exact conversion the programmers of the Spectrum version would have done well to keep the castle structure a little less complicated (using the lines of Asteroids) and which had similar 3D structures but ones which were drawn larger and more clearly yet were still complex enough to present a challenge!

By trying to pack a bit too much onto the Spectrum's screen the programmers have made the game too complex to give you to build up some speed and get really involved in it.

GOOD



called Bentley first, so in went the logo.

The 3D Pac Man description was fairly accurate in that Bentley's task is to wander around a three-dimensional 'castle' which is made up of structures of towers and walkways which become increasingly complex as each successive screen, collecting little pills which are dotted along the main pathways (just as in Pac Man). As you'll expect there are various types of monsters that have to be eaten that have to be eaten, including things that look like centipedes, bees and ghosts, and give coloured items of treasure that can be collected or left where they are in order to block the path of the pursuing monster.

Of course, as each screen is in three dimensions rather than the original Pac Man's two, the



SHORTCUTS

Ray Elder presents
more programming
gems from IX readers.

M/C Break

Fast up this month is a program from Andrew Velosost of Miami who got so fed up with his machine-made programs getting stuck in infinite loops during development that he wrote this short routine which makes use of the Spectrum interrupt mode two to scan for the 8044 key being pressed.

Two minor points to note. First you will have to reinitialize the routine each time after 8044 is pressed with `8044CMM2:USR 23377`, and secondly machine code or BASIC loader programs must not occupy the addresses from 32348 to 32375.

A useful aid for machine code programmers.

M/C Break

```
1 REM M/C Break
10 DATA 255,243,245,42,127,219
,254,203,71,32,15,42,254,219,254
,203,71,32,7,241,237,44,251,195,
3,19,241,251,201,42,40,237,71,25
7,94,201
20 CLEAR 32347
30 RESTORE:FOR I=32340 TO 323
75:READ a:POKE I,a:NEXT I
```

```
Fast 1 REM Fast
10 LET HC=48000:LET CH=58000
20 FOR I=HC TO CH:256:READ a:
POKE I,a:NEXT I
30 DATA 17,8,299,237,83,54,92,
37,8,40,126,10,234,48,32,43,42,1
24,19,19,35,124,254,44,32,-14,29
1
40 RANDOMIZE CHR:POKE HC+1,PC
50 234780:POKE HC+2,PCD:234714:
LET L=USR HC
```

Fast

It doesn't have been a regular contributor from the start, but offering this month is a variation on a frequently used theme that of a reinitialized character set. What we used above this set was that it was changed subtly giving it more space affect.

Simply enter the program (and RUN it, the character set resides at address 58000).

Scroll Relocate

```
9999 REM Scroll Relocate
9908 INPUT "Start Address "I$
9910 FOR n=1 TO 7:READ a,b:LET
a:=a:LET b=b+I$
9920 POKE a,b-256*INT (b/256):P
OKE a+1,INT (b/256)
9925 NEXT n
9930 DATA 10,120,27,144,30,159,5
9,161,70,159,73,161,126,0
```

Scroll Relocate

One of the things I like about this column is the way readers often send in odd shape modifications of previous programs. If you're of the opinion that a problem he found with Left Eye's Scroll Relocate program in the DEC 84 issue.

The original program is located in the printer buffer, not good news for printer owners, so Mr. Dossert's program will relocate it to anywhere in memory that you may desire.

First (provided you have a M/C copy of the program on tape) CLEAR memory to the new start address—1. As an example we will imagine that we are going to move the routine to address 48000. Therefore CLEAR 48000.

New LOAD "SCROLL" will CODE new address (32) LOAD "SCROLL" CODE 48000. Type in the REDOATS program and RUN it.

Finally SAVE "SCROLL" CODE new address, 48 (32) SAVE "SCROLL" CODE 48000.

Don't forget to change the line in your program which defines the function to call the new address (30:DEF FN (a):USR (a+1)-USR (a+2)).

Some alterations to the way in which the program works can also be made. POKE start-123,201 will turn the program to BASIC if a key is pressed or when the scrolling is finished. POKE start+118:POKE start+123,251 prevents the program from being interrupted and returns to basic when finished. POKE start+118:POKE start+123,0 will let the program will only return to basic when a key is pressed.

STALLONE COBRA

Cobra
Ocean
£7.95

Sylvester Stallone
converted into fun
cartoon character ...
Strange but true as
Ocean unveils an off-
beat spin-off

This is a movie spin-off game that stands up in its own right, perhaps because it bears a little similarity to the original film.

Cobra, a vigilante cop movie with Stallone at the star, was a failure by Romba standards although it contained the by now familiar quota of mindless violence. Ocean have made no attempt to make a cartoon movie game of the film and in fact Cobra is a tongue-in-cheek send-up with a ludicrous plot, absurd characters and cartoon logic.

The cop, "Cobretti", is a musclebound chunky spile who is usually armed only with a detonating headbutt which if he connects send his enemies wringing off screen in the direction they come on. As for these legions of psychopaths,



they are made up of obese wretches, old ladies looting retail outlets, knife throwing assassins and the like. This is not to mention the turbo charged prams that run Cobretti and zap his energy.

If all this sounds a bit weak, well for the plot Clay Cobretti has to rescue top fashion model Ingrid Isakson from the clutches of the evil Night Soother. All right so far but in order to do this he's got to find weapons which are hidden inside hostages at certain points in the three playing areas: city, country and a factory. Quite why a knife, pistol, and laser sighted machine gun are concealed in this way is a mystery. Anyway shaking off the mice from the weapons,

Cobretti must issue heard and completely deduce the obstacles before progressing to the next section. The weapons

however have a limited life span and tend to disappear just when you need them most.

Cobra is a very fast, very well animated game that's hard to master if not simply because you are enticed by so many bizarre comic hit/miss runs.

You are given a minute or three time to start out with and additional time can be obtained at 10000 points and further up the high score table.

The game, which can be played with keyboard or joystick, is really enjoyable and arcade fans who are looking for something a little bit different will not be disappointed. Unfortunately the image created by the film is misleading. Stallone on the play codes with his laser sighted machine gun and the hard man philosophy "Clay is a class. It's the one", look a bit silly once you've got the Stallone figure stomping up the platform after the barfucker.

It may well have been the intention to send the whole thing up and that's fine but those who like their destructive blasting untainted by any form of fun had better stick to the film.



GREAT



It's time to test your
space pilot prowess of
the Academy.

Academy
CRL
£8.95

As an unfortunate incident or an Ogyrus when a rocket was revealed up his spaceship and left the planet when he tried to dock with his central reactor the GCS came set up the Academy is an way to train rookie pilots to produce an elite corps of pilots and prevent such disasters while providing a new challenge and sequel to GCS excellent low cost.

To graduate from the Academy a pilot must complete 30 missions and achieve an average rating of over 80%. The missions are grouped into five levels of four missions that are stored on a separate data tape. Most players won't need this tape as they'll still be struggling with the first four that are loaded in with the master program.

A main menu allows you to register as an Academy cadet, view and select the missions and the skimmer to complete them.

The first four missions range from the less than subtle 'If I Move... civilization run to the beach through a minefield for local mission coded 'Silly Silly', 'Mindnum' is a race against time to close down a critical reactor and a polar world, finally 'Red Dawn' is 'You Get



style mission in which you must destroy robot factories in all quadrants controlled by jump drives.

Whichever mission you choose (and you have to tackle them all sooner or later) you'll be attacked by robot droids of both familiar and new designs. Four types of laser firing humies lead the assault supported by mines, super mines, droids, hoppers, guardians, breakers and gaily named kormloans.

Therefore it's important that your skid-coop skimmer is armed and shielded for the mission. Three skimmers are ready for take off each offering a different range of equipment from the GCS. Wilson complete with jump unit to the weapons and shields tank known as the GCS units.

If neither of these or the balanced GCS Unicorn suit your needs or tastes then you can design your own as long as it remains within the weight and cost limits set down by the Academy.

Your basic skimmer consists of low, medium or high powered laser, main drive, shield and steering units which you can then arm with missiles, anti-

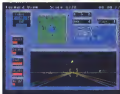
missiles and delay bombs then add whatever anti-missiles, jump units, compasses and breakers that you need (and can afford). Finally you can customise the skimmer view screen and give your landing design a tape for further missions.

Once the newly constructed GCS Bombie had wiped out the robot factories and mines that stood in its way I was ready for the next set of missions but still a long way to go before graduation. Ahead lay the dubious delights of such encounters as 'of the OC, Congo', 'Don't Panic', 'Hide and Seek', and 'Mission Impossible'.

Finally as a bonus the gamebook also includes a star map program so you can find your favourite stars and constellations. A game featuring 30 missions each as challenging as the original game doesn't need a bonus program.



ACADEMY





SILENT



SERVICE

Action above and below the waves in US Gold's sub simulation.

Silent Service
US Gold (Microprose)
\$9.95

Commanding a US submarine during the Second World War, your mission is to sink as many Japanese ships as possible while avoiding the attention of enemy destroyers.

Armed with 34 torpedoes and a 4" deck gun you must hunt down the enemy shipping in a variety of scenarios. These range from the training mission to hunting convoys in the South China Sea.

Before you walk the Submarine Commander can help America win the war against her must first learn how to control the ship. This means frantic use of the keyboard (despite the frequent mention to a pointer in the instructions) is, of 33 key controls must be mastered before you can tentatively Japanese shipyard) so practice is essential using the torpedo gun training mission. The object is simply to destroy the four old cargo ships that are anchored in position to make things easier!

Pressing Caps Shift and one of the number keys moves you between the battle stations on your submarine. That includes a navigation map, bridge and deck gun, periscope for aiming and firing torpedoes, instrument and gauge display, damage reports and a display log that reports your "kill".

As you move between these stations you must plot your course control, your engine speeds and submarine depth is well as planning your attack

Convoy Attack

How you attack the convoy depends on the conditions (day/night) and how well it's protected. An unprotected convoy is easy and you can steam up to it at full speed and blast anything that moves. Although you should attack its centre so you can use both sets of torpedo tubes, be ready for a chase though as the convoy will scatter.

A protected convoy is more of a challenge particularly if there's more than one destroyer. Now you must stalk the convoy at slower speeds while keeping a narrow profile to the ship to avoid radar detection.

able to plot your attack run, so as the destroyer is stuck helplessly on the wrong side of the convoy, giving you a chance to strike and get away. Your periscope also gives you speed, course and distance of target so you can make every shot count.

If there's more than one destroyer and you can't find a gap then you'll have to take one out. This isn't going to be easy as you'll only have one shot before the destroyer looks onto you with its deadly depth charges. Then it's DIVE, DIVE, DIVE, as you head for the safety of the depths. Colling the engine will start the

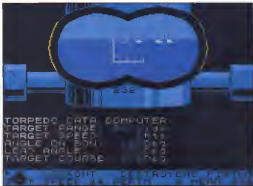
silent routine and you can even receive late orders to destroy a persistent destroyer. When you think the coast is clear, begin to stalk the convoy again.

Your sub can't hold its place so quickly on the surface and you



Your periscope is fitted with a target identifier that you should use to map out the convoy and plan your attack. You should be





should do this whenever possible as it saves the all-important battery power (and that and you lose your suit).

Complete the set scenarios and you're ready for the ultimate



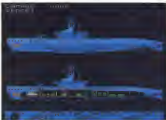
An excellent simulation that will have you sporting with destroyers until the early hours.



missions - the war patrol, when finding the enemy is as difficult as sinking them. Obviously lengthy voyages can be time consuming so you can speed up the action to two, three or four times the normal rate until you're ready for battle.

Really you can make things almost impossible for yourself by adding in skill and reality levels which include limited visibility and torpedoes, zigzagging convoys, no on-ship repair and the worrying expert destroyers.

The game cassette is accompanied by two instruction sheets that explain almost everything from the control keys to convoy routes, tactics and a submarine commander's spook guide to Japanese shipping. When these sheets are loaded there's barely room for the game cassette in the box!



Brian Beckett with news of Digital Precision's new Turbocharge Compiler.

Compiler "Win" does not star Benjamin Franklin. Neither does it feature quirky code snippets or colorful after humanoids. It's plain, Jedi or whatever. Not is it a computer game in the strict sense — although it certainly is in a wider one — but it promises to be just an amazing and probably just as fast) looking. A while ago Liberation Software reviewed the Gliberator, a superRISC compiler to compete with Digital Precision's long-standing and very well received Supercharge. Both set of \$20 and although Gliberator rates some of Supercharge's finer points and is generally size it is a good product which attracted a lot of interest.

In the meantime, Digital was developing Turbocharge. This non-assembly, multi-compiler is due for release at any time. Actually it's been due for release at any time for a while, and some reviewers have taken to needing the company a bit. According to Digital, Turbo will be significantly faster than Supercharge and have numerous additional features and extensions, designed to justify the package's size (the of the "encyclopaedia"). Turbo will set to \$24.95 which includes TurboBook: a package of over 60 new commands designed to extend and otherwise supplement SuperBasic. TurboBook is available now (for \$24.95) but these doesn't seem much point in buying it yet if you're at all interested in getting the Turbo compiler when its ready. Digital has sent me the manual so Turbo must be about due, since they never bother to write me unless there's about to release something. I didn't even get a Christmas card for me to tell you about.

In the meantime the war of words is heating up. Digital has apparently been developing Liberation Software by consistently referring to Gliberator as a "pseudo-compiler". Digital has responded with a denial (possibly of a lawyer's request) stating that the term is simply a technical one referring to the "pseudo code" interpreted by Gliberator compared to the "real assembly code" produced by Turbo's compiler (ie Supercharge) and the up-coming compiler to run on all compilers. Note: The funny bit was that this gentlemanly quibble was printed in a typeface and layout so neat that it probably takes a high resolution microscope to read it at all. Really neat.

about the competition emerge regularly from both sides and are likely to increase as time goes by. Although it might prove a bit confusing to all those potential customers out there, it promises to be loads of fun for reviewers bored with standard PR releases.

Obviously we'll be looking at Turbo when it does come out. But for a sneak preview there are some of the review features. Digital is promising basically to include a bit before writing about a product. I haven't seen yet but as Digital requires reviewers high quality packages for the GL, I have little doubt that the product will live up to its promise. First of all, Turbo will be multi-tasking (as is Gliberator) and — a real favour to the user — will lack that infernal Hence socket system found on Supercharge. It promises to be much faster than Supercharge and is said to be far more user-friendly with an extensive menu and built-in flexibility of use. It will adjust for many structural faults in the programs under compilation and let operators make it you're working with an unexpanded GL, lengthy programs are compiled in sections as would be expected but there are a number of features designed to minimise the drawbacks and difficulties of the task. Turbo tests will be able to both read and write into other Turbo tasks. Error warnings will be reported by line, statement and position. This is just the bare surface of what's promised but gives a hint of what's coming over the horizon in the (probably) very near future.

I have, however, seen the material which, although nobody is going to go out and spend \$25 for a bunch of A4 sheets, is an impressive feature in its own right. The whole thing is some 350 pages (including the TurboBook manual) and is very well written in a readable and humorous style. There are some cartoons and a glossary which includes a definition of "reviewer" that makes up for its inaccuracy what it lacks in basic human kindness. Clearly designed to amuse as well as inform, the point is that Digital obviously wants to sell its product for as wide a circle as possible and has gone to a

considerable effort to produce supporting documentation that will enable non-programmers (or those less familiar with all the technical ins and outs and various specialist traps) to get the best out of it. There are detailed explanations of some of the background maths and supporting functions which describe what is going on and much of the manual is clearly aimed at those not intimately acquainted with the GL. Professional software packaged for all users is a virtue that Supercharge had and Digital (which always takes the approach) deserves full marks for doing this at a matter of cost. The Turbo manual scores in this respect.

If you want another GL to play around with the networking facility (which has set of (big) is dead) or simply because you haven't yet got around to buying one. I've seen them for sale at \$100 on the High Street. Push it a bit and you can probably get one for a fair bit less if the retailer has gotten sick of the thing on his shelves (and I suspect most have) and wants to get rid of it. More importantly, it ought to be time to get shopping for a backgrounders manual, expansion and/or also interface. Although most get by no means all of the GL software is designed for an unexpanded machine, just about all the good stuff seems a lot better at work with extra RAM. There's a lot of competing memory expansions on the market and most of the companies are probably beginning to fill with the idea of getting rid of any surplus stock. As the market starts to gloom.

At the last couple of MicroLink, I noticed quite a few Medco disc interfaces. RAM modules in service. Medco made an excellent package, but easily ran into difficulties, but it looks like some of the original stock of modules has surfaced. If you get a chance to get one already (and I wouldn't pay much for one now) get it because it's probably the best designed GL expansion package I've seen. You will have to find someone to maintain or repair it but, if you can get it cheap enough, it's well worth it as long as you make sure it works to begin with.

Q & A COLUMN

but that memory between 4000 and 4600 is free for your printer interface software. The ZX printer and similar printers can only be used with the 486+ computers, and require no additional software to run them. DO NOT use the INITIALISE option when using the ZX printer.

WJ Left

Return to the Document menu

RJ Left

If a document has been created and a return has been made to the Document menu, this is the option that can be selected to return you to the word processor.

WJ Ctrl

For 128k users only this will print a list of text files held in RAM memory.

WJ Customise

The word processing mode of Spectrum will initially use three different colours to define the various areas of the file as it appears on the screen. The main text will be white on a dark blue background, the text entry window will be white on a black background and the pointer will be light blue. Should this combination of colours not be to your liking then you may alter them using the option. New values can be saved permanently using 'Save Program'.

Note that when text is saved the destination of saved is saved with it so that when loading back into Spectrum the colours may be different to those you may be using.

This option also allows you to redefine the F sign for your printer. The code is initially set to 128 for the SHIVA printer.

RJ BRASH

Again for 128k owners only, this option will set the field in BR4 and BR5 which are you want to erase. To do this option is selected in error just press **INITIAL** to return to the Document menu.

If you enter a filename that is not valid then the program will break with an error report. Should this happen enter gate menu/1 to reset!

Word Processing

Once you have mastered the menu system then you can begin to enter text using the word processor.

Using 'Create' to file your text you will be taken into the word processing mode, the screen will clear and you will see the following information displayed on the top of the screen: CPU: 00001 LN: 0001 CR: 161 CAPS: NW: NG: LPAR: 00000 BR4R: 0000

CR: will keep a record of the current cursor column.

LN: will keep a record of the current cursor line.

CR... will indicate when

Stop/stop mode is selected. EM... will indicate when a MODE is selected.

CAPS... will indicate if caps lock is on or off.

NW... will indicate if word wrap is on or off.

NG... will indicate if insert mode is on or off.

LMAR: will indicate the column value of the right margin if set. RMAR: will indicate the column value of the left margin if set.

The main text screen will be blank except for a series of dots at the beginning of each line. Towards the bottom of the screen you will see a black line, this is the text window in which all typing is carried out. Above this window there is the text line line to remind you of the number given to your document below the window is the end of text marker line. A cursor will be flashing back/white at the beginning of the text window.

Initially your text file is just one character long, a new line character code. If represented by a dot will be seen under the flashing cursor. All margins and text will be reset and insert mode will be on (indicated by the white bar option) INS in the status line) and word wrap will be off.

In word processing there are three distinct modes which give access to the various functions.

1) Text Mode

This is the standard mode on entry to word processing in which normal text entry is carried out. Approved (see a keyboard buffer) to ensure that all line presses are read so that with the latest typeset will not be able to cut type the program.

Use the keyboard as you would in normal typewriter. Typing will always appear in the text window so that you do not have to search the screen to find out where you are. When the cursor reaches column 29 the screen will scroll left to keep text entry in view of all lines.

Text entry will continue on line one until BR4R is pressed to indicate a new line is required (see margin). You will notice a dot placed at the end of each line, this indicates a new line character and is part of the text, that is it can be erased to join two lines together or inserted into a line to split it into two separate lines.

In this mode the MV VIDEO key has a special function in that if pressed it will display a HELP page for that mode. These HELP pages give basic information on all the main functions available in that mode. Text mode: **Line/line**

R Arrow: The cursor can be moved around the text using the four arrow keys. Left and right arrows will move the cursor one

character at a time through the text in the present direction. The up and down arrows will cause the screen to scroll down and up to place the required text line into the cursor window. When moving up and down through the text the cursor will try to stay in the same column, but if the line to be moved into the text window is shorter than that column then the cursor will move to the end of that line (this is similar to program editing on the 128k Spectrum). Note that the keyboard is buffered and if the arrow keys are held down too long you will overshoot the required position.

R Delete: This key will delete the character behind the cursor and backspace to that position closing up the text.

RJ Two Video Move: In the next TAB position (see 1 MODE for the method of setting tabs).

RJ Graphics: This is the second mode available and is mainly concerned with the printer control characters (see GRAPHICS MODE).

WJ Left Mode: The third mode available mainly concerned with the text format and coding (see RT MODE).

WJ Left: Return from word processing to the Document menu.

In text mode all the normal characters are available using CAPS SHIFT and SYMBOL SHIFT. CR/LF/CR/LF will send carriage in the normal way and indicate that it is on by placing a white flash against CAPS in the status line.

2) Graphics Mode

This will be of interest if you have a dot matrix printer that has standard controls to select different text formats. Selecting GRAPHICS will place a white block against BR4 in the status line.

MV VIDEO will produce a HELP page giving the control code letters and their original functions. You will see that for most functions UPPER case is used to switch on the function and lower case to switch it off. There are 30 characters available all of which can be defined to suit your printer.

Printer control characters do not affect the line length of the text sent to the printer.

There are also seven long characters that can be printed in this mode. These characters can be used to place brackets around your text! The following keys are used and as you will see each is supposed to make them easy to find.

G	W	R	+	-
A	S	D	+	-
Z	X	C	+	-

The codes used for these characters are initially for the SHIVA CP40 printer but can be

changed to fill your printer with a few simple BASIC commands. The initial codes are:

```
[ 134 | 135 | 140 | 157  
 136 | 138 | 142 | 158  
 133 | 139 | 141 ]
```

Each code has two addresses as follows: the first address is for UPPER case, the second for lower case so in fact you can re-program 22 new characters.

```
34020:130  
34021:134  
34022:157  
34023:151  
34024:149  
34025:136  
34026:134  
34027:157  
34028:157  
34029:159  
34030:133  
34031:138  
34032:133
```

CHARS LOCK also works in this mode.

This mode also has a 'test' view which will give a display of the current screen at 128 characters per line. This will be in the form of a graphics display with a letter represented by a line. This can be used to check the format of your text. To obtain this view mode press key 1. The display will remain on until another key is pressed when a return to TEXT mode will be made.

3) Ext. Mode

This mode is entered by pressing the EXTEND MODE key and will be indicated by a white block against the in the status line.

MY WORD will display two HELP pages to give you a summary of the various functions available.

The EXTEND MODE key should not be held down once E MODE has been selected.

The characters W:\ | [] are available in this mode and the copyright symbol which is not included has been replaced by the symbol the symbol is available in text mode using BRACK, SPRINT.

Arrow keys have a special function in this mode as follows: UP will take you to the start of the text file.

DOWN will take you to the end of the text file.

LEFT will take you to the left margin or if not set the beginning of the line.

RIGHT will take you to the right margin or if not set the end of the line.

W (W) will set the left margin to the current cursor value, and display that value against LMR in the status line.

R (R) will set the right margin to the current cursor value, and display that value against RMR in the status line.

V (V) reset the left margin to zero.

V (V) reset the right margin to zero.

W | Insert mode ON This will be

indicated in the status line by placing a white block against MR in the mode characters will be inserted into the text displacing the text to the right to make room for that character.

INS in this mode will produce a new line in the text displacing the text down one line to make room. The cursor will flash backwards.

W O Overprint mode ON This will place a block against MR in the status line. The cursor will flash backwards; text entered in this mode will overwrite the current text. If the right margin has been set then this mode will not allow printing to continue on the next line until ENTER is pressed. This has been done to stop overprinting of text on the next line without realising that it is OK to do so. A reminder that the right margin has been set will be printed on the screen should you attempt to exceed the right margin.

W (H) Word wrap ON This will be indicated by a long white block against WR in the status line. There will also be a change of cursor colour, the white block will change to blue. That is INSERT ON WR ON = BLACK/BLUE INSERT OFF WR OFF = BLACK/WHITE

WR ON WR OFF = BLACK/WHITE
WR OFF WR OFF = RED/WHITE

Word wrap will only function if the right margin is set and insert mode is on. Should a word continue from the end of a line to the beginning of the next line then that word will automatically be transferred to the next line.

W (J) Word wrap OFF This is indicated by a short white block against WW in the status line. Words will remain split between the end of one line and the start of the next providing that the right margin has been set.

W (D) Down word) Selecting this function will produce a display of the number of words in your text that will not include any graphics printer control code letters or block markers. The count is instantaneous and will remain on screen until a key is pressed.

W W Word find This will search through your text FROM the CURSOR POSITION to find a word or phrase of your choice. You will be prompted to enter your search text and whenever you search the file until it finds the required word/phrase and place the cursor at the start of that word. Should the required word/phrase not be in the text then the message WORD NOT IN TEXT will be displayed and the cursor will remain in its original position.

W (S) Set DEL This will set up a tab stop at the current cursor position. The cursor will flash back to show that the command has been accepted.

Up to 22 tab stops can be set. DEL may not be set with current values of less than 255. DEL is reset DEL This will remove a tab stop. The border will flash red to indicate that the tab has been removed.

W (S) Set of TABS The command will remove ALL the set tabs and the border will flash green to indicate that the command has been accepted. W (S) Set block start This command will place a block start marker at the current cursor position. It will be indicated by a forward facing arrow. W (S) Set block end This command will place a block end marker at the current cursor position and will be indicated by a backward facing arrow.

Any previously set block markers will be removed before a new block marker is set. W (S) Set block This will remove any set block markers from the text.

Block markers can be treated as a character in the text file, that is they can be erased or re-positioned by inserting text into the gaps. This will have no effect on any of the block functions. Spaceword will allow for any change in position of these markers.

W (C) COPY BLOCK This command will copy a set block to the current cursor position opening up the text to make room. A block can be any length from just one character to the whole text file. If there is not enough room left in the text file to make a block copy then a message to that effect will be given and the copy command will be ignored.

The block copy will be free from block markers. Thus will remain set on the original block.

W (M) MOVE BLOCK This will move a set block from its current position closing up the text to the current cursor position opening up the text. This command will work even on a full text file. The block will be moved complete with its block markers.

W (Z) ZEROT BLOCK This find command will erase a set block from the text file closing up the text.

Block markers can almost be ignored but will not work if a) the cursor is placed inside the block to be MOVED or COPIED.

b) the start marker is positioned AFTER the end marker of there is insufficient text room to accept block COPY.

As an example of the text that can be created in a file, these instructions were produced on a Commodore and took 994 lines of which 222 were produced in one text file (222 words). The remaining 772 lines were produced in another text file (264 words).

IMPOSSABALL



It's Impossible! A new dimension added to the bouncing ball puzzle game by Newton.

Impossible!
Newton
£8.95

Like all the best games Newton's Impossible! is simple to play but hard to master. It's one of those games that sort of sneaks up on you after a couple of attempts — you may not get very far at first, but as you move up against each new obstacle you don't realise how to get through, so you go back to the start for one more try and before you know it you're hooked.

The basic idea behind the game is very simple, but it's been

excellently implemented by programmer John Phillips. You control the movements of a bouncing ball over a rotating three-dimensional obstacle lit

an attempt to complete the course within a time limit, and at the same time as avoiding the deadly obstacles you also have to 'squeeze' a series of cylinders



ball are scattered over the course and which one of those surrounded by some of the more deadly traps and barriers.

These two easier courses in the game each feature three levels, though even the first of these isn't exactly a push-over. At the start of each course the ball is set behind the starting line for each course and, since the timer doesn't start until you've crossed the line, this means that you're given an opportunity to bounce the ball around a bit (and get the hang of controlling its movements) before you start the game properly — a nice idea, and one which allows you to get started calmly (unlike a lot of games wherein, if the ball doesn't bounce you to level a 99 page manual before you can get started).

At you cross the starting line you move from right to left across the screen and the obscured landscape of the course opens smoothly along to reveal each new set of obstacles.

Poles apart

On the first course this consists of spike-topped poles which jut up from the ground or hang from the ceiling, and placed both (which look like distorted rings). The cylinders that have to be 'squashed' are column-like structures projecting from the floor and ceiling, and 'squashing' them simply involves leaping into them from above or below, forcing them to retract into the surface. On later courses though, there are all sorts of deadly traps (most of which are moving and require expert judgement and planning to get through).

Surprisingly for a game where you're controlling a moving object and seeing against the clock, *Impossible* doesn't require dazzling arcade-trained reflexes. Despite the time limit there are places where you're better off taking things slowly and thinking your way through an obstacle rather than just bolting through as fast as you can. But when the clock's ticking away and you need to hustle down the home straight to the finishing line, control of the ball is very simple, and the way in which the pace of the game can vary makes it that much more addictive.

One of the really nice features of the game is that it presents you with each course and then just lets you get on with it. There isn't a single, fixed method of solving each obstacle and there's enough flexibility in the game to allow each player to develop his or her own style of playing. The only suggestion I can make that



might improve the playability of the game is that after you've completed a course it might have been a good idea to give you a password that allows you to enter the next level whenever you want. As it is, you've got to go through all the course in a fixed order and even when you've got the first level sorted out you're still got to go through it each time in order to get onto the later ones.

Flicker free

The graphics and animation are about as good as I've seen on the Spectrum. All the objects are large and finely drawn, and the animation is perfectly smooth. There's not the slightest flicker despite the fact that the whole screen display has to swing backwards and forwards very quickly in order to create the impression of bounding movement. The checkered pattern on

the landscape helps with the 3D perspective and also helps when it comes to judging distances and heights of jumps past obstacles, and, as the ball bounces along its shadow follows along, changing as growing as the height of the jumps varies.

There have been other games released recently (*Revolution* and *Action* refile) but *Impossible* beats them all in terms of graphics and playability, and it makes a nice change from shoot 'em ups too.



A budget BMX racing game that's fun from start to finish

BMX Simulator
Codemasters
£1.99

While BMX racing may not have the same status of glamour as motorcycle or speed petro racing, when it comes to computer simulations BMX comes into its own. Formula one racing simulations are often a bit of a left-feeler, your accelerator may be solving 248 mph but the more action doesn't reflect it, but with BMX racing a simulation can get closer to the real thing and the intricate detail and control required to negotiate the course make it ideal for those who want to use their joystick for more than model-air play.

BMX Simulator is an excellent game that is surely destined to become a budget hit. There are seven different courses, seen from an overhead view and these offer a multitude of daunting features becoming harder with each successive track to qualify for the next track you must complete the course within a given time limit — this is fairly generous for the first course but becomes more stringent with track two.

There are options for one or two player games. When playing the computer rider it's gratifying to find that he's very fast and wipes out fairly often, though he does have the advantage in that whenever you collide with him you are alerted the one to go flying. Crashing is illustrated by a well animated

BMX SIMULATOR



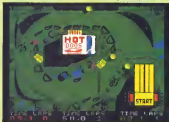
sequence where your rider and bike fall comically in spectacular fashion. Restarting the race can be a problem with a delicate touch needed to steer yourself clear of the object you've run into, otherwise you crash again... and again...

The two player game brings out the full potential of the simulation and most importantly there are options to play with two joysticks (temptingly not!) this was a wise decision as anyone playing the keyboard option would be at a real disadvantage. Two options have

been built in to pep up the after-race debate, all who cut up who on the track being, there is an option replay function which will play back the race, as if you are really into detailed analysis you can also watch the whole race again in slow motion. It's not sure how often these would be used, after the event has worn off but they can cheer additional nonchalance.

Above all, BMX Simulator is very realistic and the bike reacts to the terrain exactly as it would on a real track — the ripples, turns, bumps and water splashes are not let there for decoration, they can be used for your advantage.

BMX Simulator will appeal to all those who like race games even if the idea of BMX racing seems a bit tame. Everything about the game's presentation is highly professional from the graphics to course design — another game to add to Codemasters' increasingly impressive track record.



VIEWPOINT

David Nowatnik tests a new 3D graphics package for the QL.

**Viewpoint
Rubicon Computer
Systems
£9.95**

If you have ever been fascinated by the ability of computers to display three dimensional images, then an excellent new software product for the QL will permit you to try out the concepts for yourself. 'Viewpoint' is a machine code program which allows you to construct 3D wire objects, then display the object in two dimensions (the monitor screen). Objects can be viewed from different viewpoints, and you can zoom in and zoom out, change the perspective, achieve some hidden line removal, and save and load object data to and from associative or disc.

A printer dump of the screen display is also possible, and Fig. 1 shows the screen dump on an Epson printer of just one of the example objects supplied with Viewpoint.

The product comprises a single microcassette cartridge and a 20 page 5" x 3" manual. It is demonstration of the simplicity of use of Viewpoint that the small manual more than adequately covers the operation of the program. The program is menu-driven, with options appearing in the menu window at the base of the screen. Five main menus are available, obtained by pressing one of the function keys.

While the program itself is easy to use, 3D graphics, in concept are rather tricky. Here there are few buzz words to understand: world coordinates, spherical polar coordinates, cartesian base, vector, angles O and O' built in to Viewpoint is an option to have some of these terms displayed graphically. This can be done at any time, even while you are constructing an object. Pressing 'R' and 'C' displays a simple cube in three dimensions, while 'V' and 'O' display the axes of a spherical polar coordinate system. Within these helping options, you can play around with vector R, and angles O and O'. Once you have gained a feel for these, you can begin to constructing your own object.

Objects can be built up by drawing lines between two points. A number of 2D (circular box, polygon) or 3D (sphere) shapes are drawn automatically at the cursor position, with sizes specified by the user. Several drawing aids are included, such as the addition of axes and/or a calibrated grid. Two vertices (coordinated previously defined) can be joined by a simple join command. The mirror image of an object can be produced with the user defining the plane of the mirror. And, text can be added, this text remains fixed on the screen display, and does not move if the viewing position of the object is changed.

Mirror Images

Pressing 'S' brings a search facility into play, in which the computer tests the nearest defined vertex from the current cursor position. The search area is only a 9x9 grid of the world coordinates, which, in effect, means that the cursor has to be visually on the vertex point for it to be found. A bug in the review copy meant that the program crashed if a search was requested when no vertices had yet been defined, this bug the

screen. With large objects, this can be quite time consuming. In particular, multi colour displays are re-drawn quite slowly compared to monochrome objects. However, to save time, Viewpoint allows the user to temporarily turn off the multi colour facility. The re-highlighted objects slightly can be redrawn much faster.

For the more experienced user, Viewpoint has a number of help facilities to aid object construction, such as listing (screen and/or printer) all fixed vertices or surfaces of the object under construction.

For a printer screen dump, Viewpoint has the relevant control codes built in for Epson and dotmatrix printers.

If you do not have a printer compatible with either of these printers, then the codes relevant to your printer can be entered as a set of 8 bytes. Unfortunately, there is no way of saving these codes, so if you do have to enter special codes, you must do so every time you use Viewpoint. During review I had a few problems with achieving a printer dump on my Epson 8300. This turned out to be related to my printer's dip switch positions. So my Epson turned out to be



Screen dump of a 3D object created from Viewpoint.

supplier promise to remove it all future copies. Each line or shape can appear in any user defined colour on Viewpoint's black background, switching between 4 and 8 colour modes can be accomplished with the graphics, diagrams being redrawn to the same scale as soon as the mode has been changed.

In moving the view position relative to the object, the object has to be redrawn on the

(according to Viewpoint) non-Epson compatible. It was a simple matter to use the built (holding one switch position) but worth noting if you are using Viewpoint and experience a similar problem.

Overall, Viewpoint is an impressive package, and, at a RRP of £19.95, provides excellent value for money. Rubicon Computer Systems, 11 Barnwood Road, Sheffield S7 2SL.

THE VALUE OF

TABLES

Pete Cooks, author of *Tou Geff and Academy*, lets us in on some of his machine code secrets.

Listing 1

```

1 SET (initial table for screen addresses... check cursor code before running)
20 SET address = 40000; REM #FF00 on line
30 FOR row TO 255: REM Number of screen lines on Spectrum
40 SET address = 40000; REM 1st row of screen
50 SET address = 40000; REM 1 line within a block
60 SET cursor = 0; REM #000000 screen colour
70 SET address = 40000; REM 1st screen character block
80 SET cursor = 0; REM #000000; REM #000000; REM #000000
90 FOR row = 1 TO 255: REM #000000; REM #000000; REM #000000; REM #000000
100 SET address = 40000; REM #000000; REM #000000; REM #000000; REM #000000
110 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
120 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
130 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
140 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
150 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
160 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
170 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
180 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
190 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
200 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
210 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
220 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
230 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
240 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
250 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
260 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
270 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
280 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
290 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
300 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
310 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
320 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
330 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
340 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
350 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
360 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
370 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
380 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
390 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
400 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
410 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
420 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
430 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
440 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
450 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
460 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
470 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
480 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
490 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
500 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
510 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
520 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
530 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
540 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
550 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
560 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
570 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
580 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
590 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
600 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
610 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
620 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
630 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
640 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
650 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
660 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
670 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
680 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
690 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
700 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
710 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
720 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
730 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
740 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
750 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
760 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
770 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
780 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
790 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
800 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
810 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
820 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
830 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
840 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
850 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
860 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
870 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
880 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
890 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
900 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
910 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
920 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
930 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
940 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
950 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
960 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
970 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
980 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
990 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000
1000 SET cursor = 0; REM #000000; REM #000000; REM #000000; REM #000000

```

One of the most effective techniques in machine code is the lookup table. In basic lookup tables correspond roughly with single-dimension arrays holding tables of data, for example —

```

10 DIM A (20)
20 FOR row TO 20
30 READ A(row)
40 NEXT
50 DATA (whatever you want)

```

In machine code lookups are much more common as they give a way of evaluating complex functions without writing masses of complex code. Two good examples of the use of lookup tables that spring to mind are *3M Tables* and *SCREEN ADDRESS*.

Screen Addresses

One of the biggest hurdles to graphics programming on the Spectrum is the awful layout of the display file. Anyone watching a *SCREEN* loading will have seen that the display lines are stored in a non-sequential fashion and a great deal of time and effort is needed to write a fast and efficient routine to walk out the screen address of any line.

A much simpler way is to generate a lookup of screen addresses in memory. I normally use a short BASIC program to do this. Listing 1.

As an additional check that the program is working the line —

NO POKE ADDRESS

fills the last byte of each line of the screen as the program executes its address.

When the program has finished type *GC00 130* to save the table.

I normally store a table like this as high in memory as possible but if your computer is stored at a high address this might be the value lookup to somewhere else (it's best to use a multiple of 256 for reasons explained later).

Having got the table into memory screen addressing now becomes much easier. To get a screen byte address we only need a short segment of code like the one in Listing 2.

In fact we can do even better than this by entering the table part of a register of 256 bytes. In that case the low byte of the table address is always 0 (Listing 3).

Although this looks as long as the first version there are several improvements. First the DE register pair is not used. Second most of the instructions are single byte instructions which execute much faster and take up less space on a 256.

Finally note one 2 simple examples using the table lookup the first is a fancy clear screen this won't touch the attributes (Listing 4) and the second is a routine to paint a set window (Listing 5).

The beauty of lookup tables for screen addresses is that, with the table in memory, graphics routines become much more straightforward and dozens of routines in a long program can use the same table.

Listing 2

```

10   DEB #D000
20   ENT #0000
30 YTABLE DEB #FF00 (whatever look up is
37  ;
40  ;
50  ;
60  ;
70  ;
80  ;
90  ;
100 GETTAB LD   L,B
110      LD   HL,0
120      ADD HL,HL ;2 bytes in each table value
130      LD   DE,YTABLE ;base of the table
140      ADD HL,DE ;first offset into table
150      LD   A,(HL) ;pick up low byte
160      ADD A,A ;add on column offset
170      LD   B,A ;take low byte of DE
180      INC HL ;point HL to high byte
190      LD   B,(HL) ;pick it up
200      EX  DE,HL ;transfer to HL
210      RET
220  ;

```


ACROSS THE PONID

Mark Fendrick previews America's largest Sinclair show.

Finally I want to answer those of you who have written to ask why this column has not been appearing every month. This was not because Byron had asked for less columns, but because of the situation concerning the Sinclair market here in the United States. In short, since Sinclair has left our part of the market, and Amstrad has no interest in promoting Sinclair technology in North America, nothing much has happened. The number of Sinclair dealers has remained about the same — or even had its numbers reduced. New sales of Sinclair computers — mostly the QL — continue to be in constant flux. Computer owners who have decided to upgrade from their BS 2064

There are still a handful of new products being developed, but not in sufficient numbers to appeal to new home computer buyers, and lack of IBM compatibility turns most potential customers off. The incentives which were 32 Cities personal loans have proven unworkable and many computers which last sold fine by the dealer drive at the retail order customer in an unworkable condition.

But before you all start thinking that I have given up on the Sinclair computers, let me assure you that nothing is further from the truth. More than three years ago I promised that North American Sinclair computers will always be able to find support here, and other than dwelling on what is not, we shall see out all the new happenings and report on them here in ZX Computing Monthly.

Last year one event proved more of a shot in the arm than any other happening. That event

was the first annual Midwest Times Sinclair Computer Fair held in Cincinnati, Ohio.

Vendor who attended the event all come away from Cincinnati with the feeling that their still is a great deal of interest amongst the Sinclair computer owners in keeping their wits about. A few new products made their appearance at this show and a number of extremely well attended lectures rounded out the weekend's schedule. All predictions of the long awaited demise of the Sinclair product in the States were once again put to rest.

Mid West Fest

While other shows were planned across the country, none ever got off the drawing board, but the Second Annual Midwest Times Sinclair Computer Fair has been announced.

Frank Davis, one of the organizers of last year's smashing success has indicated me that this year's gathering will be even bigger and offer more to the dealer and end user than the first fest.

The site for the 1987 affair will be the Holiday Inn North in Indianapolis, Indiana. This location has many improvements over last time. Since many attendees come by way of air (more than had ever been expected) the long drive from the airport and mine mine for the first side to the office. The Holiday Inn is not only much closer to the airport, but the hotel supplies a shuttle service from the airport square feet of show space dedicated to the show as opposed to the 3000 which was taken up in Cincinnati. There will be tables available for vendor and user groups as that of (possible) aspects of the Sinclair computing community will once again be represented.

Because the lectures and

discussions proved so very popular last year, two rooms have been reserved for this purpose. The plans call for the discussions which proved popular to be repeated at various times throughout the course of the weekend. (Once again I have been asked to lead one or two of these discussions. I hope to have the pleasure to meet many of you at that time.)

One of the more popular events which could have used room this last year was a swap meet in which participants could swap any form of software/hardware which they wanted. At last year, arrangements have been made to assure that there will be no illegal copying of copyright software. Those coming to Indianapolis should be prepared to bring those items which they wish to swap, and possibly may be turned into something you could use. This may be the opportunity you have been waiting for to obtain that product you never purchased because it was taken off the market.

The date for this once a year event is the weekend of May 2nd and 3rd.

For more information concerning the show (and including ideas for tables or dealer, distributor and user groups, contact show organizers Frank Davis, 631 East Main Street, New Indiana 46070. Please include a \$2.00 to help defray the cost of postage. I hope to see all of you there in Indianapolis.

I have a number of new software packages which have gathered assembly which I will report on in the next few columns. Among the programs are music synthesizer for the TG 2064, a number of new telecommunications products for the entire product line, and a step in the direction of desktop publishing on the QL. I have been informed that there are a few products which will be introduced in the next few months, and as soon as I receive my review copies I will give you my impressions.

For those of you who want to get the know about new products that you have heard of, or new applications that you have discovered, let me remind you how to get in touch with me. If mail I can be reached at:

Mark L. Fendrick
Post Office Box 2393
Bloomers, NJ 07004-0992
USA.

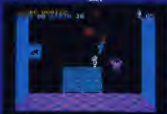
Electronically I can be found on THE SOURCE (BGAASZ) Computers (7447) 32549 DELPHI (A4487) (N089C), by name to MCI MAIL, or via telex (via WU) 520312475. Feedback for review may be sent to the above address as well — hint!

to the 1980s and 1990s, and the 1990s. Quality and price were some things that were also the most important when it came to the best of the best. The games in that issue had the quality of the best of the best. The games in that issue had the quality of the best of the best. The games in that issue had the quality of the best of the best.

Most of the time, it was the quality of the graphics that was the most important. The games in that issue had the quality of the best of the best. The games in that issue had the quality of the best of the best. The games in that issue had the quality of the best of the best.



Game 8



Game 10



Game 11

Despite the slip, and ending your way through the deadly traps which occupy the enigmatic chambers.

By the time they produced *Ultimate Adventure*, the team were in two mind-sets. They had perfected a style of 3D graphics that was revolutionary and which has more or less become adopted as the standard way of presenting 3D graphics. *Merlin*, *Colman*, *Sevendin*, *Mines*, *Prodigy* and many other games are shiny examples of the format that all have pioneered, and *Ultimate Adventure* is no exception in its combination of problem solving, difficult obstacles, and detailed graphics.

Today *Ultimate Adventure* has progressed from this point and while they've produced some great games since they've been unable to get back to the point that they reached with the games in this collection. Except these games were so influential, people who have only gotten into computing in the last couple of years are likely to be more familiar with the large number of games that have copied *Ultimate Adventure*, but these games are still among the best of their various years, so this compilation is worth getting if you're at all interested in classic videogames.

GREAT



ULTIMATE GAMES

BIG

SUBSCRIPTION SAVINGS FOR YOU!

Take advantage of these fantastic money saving subscription offers to these magazines. Hurry, this amazing offer is for a limited period only.

UNITED KINGDOM

OVERSEAS

(Postmaster please post)

	Normal Price	Sale Price	Please Tick	Normal Price	Sale Price	Please Tick
ASB Computing	£24.00*	£18.00	<input type="checkbox"/>	£28.00*	£20.00	<input type="checkbox"/>
Computer Games	£28.00*	£13.00	<input type="checkbox"/>	£24.00*	£13.20	<input type="checkbox"/>
Your Commodore	£28.00*	£13.00	<input type="checkbox"/>	£24.00*	£13.20	<input type="checkbox"/>
ZX Computing Monthly	£28.00*	£12.00	<input type="checkbox"/>	£28.00*	£14.40	<input type="checkbox"/>
Citizen's Band	£28.00*	£12.52	<input type="checkbox"/>	£28.00*	£13.80	<input type="checkbox"/>
Ham Radio Today	£27.00*	£14.40	<input type="checkbox"/>	£24.00*	£16.80	<input type="checkbox"/>
Electronics Digest	£24.00*	£ 8.30	<input type="checkbox"/>	£24.00*	£11.20	<input type="checkbox"/>
Electronics Today International	£28.10*	£14.40	<input type="checkbox"/>	£22.00*	£18.00	<input type="checkbox"/>
Video Today	£28.00*	£12.52	<input type="checkbox"/>	£24.00*	£16.80	<input type="checkbox"/>
Which Video?	£28.00*	£12.52	<input type="checkbox"/>	£24.00*	£16.80	<input type="checkbox"/>
Photography	£28.00*	£12.00	<input type="checkbox"/>	£24.00*	£18.00	<input type="checkbox"/>
Photoplay	£27.00*	£14.28	<input type="checkbox"/>	£22.00*	£17.20	<input type="checkbox"/>
Clocks	£28.00*	£24.04	<input type="checkbox"/>	£24.00*	£28.00	<input type="checkbox"/>
Woodworker	£28.00*	£12.52	<input type="checkbox"/>	£24.00*	£18.00	<input type="checkbox"/>
Popular Crafts	£27.00*	£14.28	<input type="checkbox"/>	£24.00*	£13.20	<input type="checkbox"/>
Wine-maker & Brewer	£28.70*	£10.98	<input type="checkbox"/>	£24.00*	£13.80	<input type="checkbox"/>
Aeron modeller	£28.10*	£20.08	<input type="checkbox"/>	£28.00*	£22.20	<input type="checkbox"/>
Military Modelling	£28.00*	£12.00	<input type="checkbox"/>	£24.00*	£18.00	<input type="checkbox"/>
Model Boat	£28.10*	£12.00	<input type="checkbox"/>	£28.00*	£18.00	<input type="checkbox"/>
Radio Control Model Cars	£29.10*	£14.00	<input type="checkbox"/>	£24.00*	£17.20	<input type="checkbox"/>
Model Engineer	£29.40*	£22.00	<input type="checkbox"/>	£28.00*	£28.00	<input type="checkbox"/>
Radio Control Boat Modeller	£ 8.00*	£ 7.60	<input type="checkbox"/>	£24.00*	£ 8.20	<input type="checkbox"/>
R C M & E	£28.00*	£12.00	<input type="checkbox"/>	£28.00*	£18.00	<input type="checkbox"/>
Radio Control Scale Aircraft						
Quarterly	£ 9.70*	£ 8.60	<input type="checkbox"/>	£24.00*	£ 8.20	<input type="checkbox"/>
Radio Modeller	£28.10*	£12.00	<input type="checkbox"/>	£28.00*	£18.00	<input type="checkbox"/>
Sea Classics International	£20.00*	£ 8.00	<input type="checkbox"/>	£28.00*	£18.00	<input type="checkbox"/>
Scale Models International	£28.00*	£12.00	<input type="checkbox"/>	£28.00*	£18.00	<input type="checkbox"/>
Your Model Railway	£28.00*	£12.00	<input type="checkbox"/>	£28.00*	£28.00	<input type="checkbox"/>

(Offer ends 30th April 1987)

Please enclose my subscription(s) with the invoice

I enclose my cheque/postal order for £ made payable to Agas Special Publications Ltd

or debit from my Access/Barocard No.

valid from to Signature

Name

address

Send this form with your remittance to: **Subscriptions Savings Offer - S.O.F.S.**

INFOPRINT LTD., Floor Nine, 77 The Barbours, Great Westwood, Bore, WPI 166.

DEBUGGING ARTIST II

Flummoxed by the bugs in Artist II? Carol Brooksbank rides to the rescue.

You may have seen in the review of *Sotchi* that I mentioned that there are a number of bugs in the program. Certainly, if you have a copy, you will have met some of them.

This is such a good graphics program without the bugs that I decided to see how many I could remove, and now have my copy in good working order. As most of the changes are quite small, I can see you would like to check your copies up too, so here it is how it is done.

First, a few bugs that I have left alone because their effects are easily corrected or avoided. After using the 'Up' mode to work on the part of the screen normally hidden by the join menu, you will sometimes find that a couple of lines from the bottom of the screen are transferred to the top when you return to the 'Down' mode. This is easily corrected by using the 'Window' option to scroll the whole screen. When using the 'Enlarge' mode to work in detail on small sections of the screen the program may crash if you run the cursor off the right-hand edge of the enlarged window. Avoid running the cursor off the edge. When you need to work off the right-hand side of the enlarged area, move the window over a bit so that the area comes to the centre of the window. If you are on the right-hand edge of your picture, move the cursor slowly so that you can control it, and keep it inside the window.

I mentioned the save/load bug in the type font design option in the review. The 'Save' option saves the wrong block of code, and 'load' crashes the program. Follow this procedure to save/load successfully.

Select the save/load option, and give the file name when prompted. Use the ESCAP key to return to BASIC. Do not start the tape if saving; enter as a direct command.

LET Statement

number = 32385 for font 1
 31500 for font 4
 30835 for font 5

Enter 0010

BA62	E5	PUSH HL
BA63	C5	PUSH BC
BA64	F5	PUSH AF
BA65	CD78BA	CHKBUS
BA6B	20BF	JR NZ, CHKBUS
BA6A	F1	POP AF
BA6B	01BFFF	LD BC, FFBF
BA6E	ED79	OUT (C), A
BA70	0EBB	LD C, BF
BA72	ED79	OUT (C), A
BA74	0EBF	LD C, BF
BA76	ED79	OUT (C), A
BA7B	C1	POP BC
BA79	E1	POP HL
BA7A	C9	RET
BA7B	01BFFF	BUSY
BA7E	ED7B	LD BC, FFBF
BAB0	E6C0	IN A, (C)
BAB2	C9	AND CO
		RET

Busies in above case of the Kempston II interface.

72 to load from microdrive/disk
74 to load from tape
80 to save to microdrive/disk
84 to save to tape

Proceed as usual to save/load

The most serious bugs are in the painting operations, in both the main program and the Pagermaker. The following changes should be made before saving a back-up customized copy.

Kempston II

The last command on LINE 97 of the main program BASIC is
GOPT : SEM CHKS NOT IN

My interface is not intelligible by this, I am not sure whether this is a quirk of my particular interface, or whether it is always the case that NOT IN is not recognized. If your interface is not intelligent after responding 'Y' to 'KEMPTON II???' then change the last command on LINE 97 to **GOPT : SEM CHKS :**

Enabling the screen dumps

The shaded screen dumps work, but the ordinary large and small ones do not if you are using Epson or compatible printers. For the technically minded, this is because there is a mistake in the Epson bit-image graphics codes. The program makes use of the ESC * control code, which allows the bit image mode to be

selected by a number which follows. This number can be 04 or 6, or 5 on some of the Epson models. The program uses '7'. A call to Epson confirmed by suspicion that there is no such mode, so the last one (ie corrected by FORKING) a valid number into the locations concerned. You can experiment to see the effect of the various numbers, but I found 1 the most satisfactory. The only problem is that the large screen dump does not quite fit on the page - about two character squares jut out off the right-hand edge - but none of the other modes produces a large dump which comes anywhere near to filling the page.

Here are the FORKs you need.

FORK 65186,1 (Small screen dump)

FORK 65144,1 (Large screen dump)

You could be juggling with the values of the two pairs of bit-image code. I could not find a more satisfactory combination than those in the program, but if you have more patience than I, the locations and present values are:

65181 (S) (Small dump)
65182 (S)
65183 (S) (Large dump)
65184 (S)

The main program can now be saved. As I have no Epson disc unit, I changed a line 95 to make the program continue.

IS THERE THIS ANSWER TO OLDSCHOOL SPECTRUM "GOLF" CODE

SAVE "GOLF" LINE 96-SAVE "GOLF"
CODE 24500,24500

PageMaker

The separate PageMaker has the same 58-image limit. If you have a Kempston II interface, ignore the instructions in the handbook for loading a customised copy and proceed as follows.

Load the PageMaker from the tape. As soon as the instructions for customising appear on screen, stop the tape and use the BREAK key (ie, before the blocks of code are loaded). Change the number of LINE 20 to LINE 31. Add

28 COPYFORM CODE 0

Enter 20 TO 30, start the tape, and the blocks of code will be

loaded. When the whole program is loaded, use the BREAK key to go to PAGE, and enter PAGE 44004,4.

You cannot juggle with the 58-image data in this program as they are transmitted by loading A with a number, followed by 20 000. To change the 0 data, you would need to replace a one-byte instruction, 20H A, with a two-byte one. The data in the main program is fixed in a table of values which can easily be changed.

You can now save a backup copy of the program with:

SAVE "PageMaker" LINE 20;
SAVE "COPY FORM 28" 2000,2000;
SAVE "COPYFORM CODE"
40000,4000; SAVE "COPYFORM"
CODE 44000,400; SAVE "write-
test" CODE 27000,40.

Your customised PageMaker will work perfectly, except that you must remember when making

The Writer real file, the maximum number of characters per line must be 62, not 64 as stated in the handbook. A full Atlas I screen still covers 32 characters, so you will need to take account of the fact that a screen takes slightly more than half the width of the page.

If you wish to use the PageMaker, you will have a copy of The Writer, and those with a Kempston II interface will also have found that this does not work. This is because, despite the handbook, the code in The Writer is for Kempston I. To get a Kempston II going with it, you need the bit of code in figure 1, and to load it into the program of 40716.

I don't normally think it is a reviewer's job to debug commercial software, but this suite is so useful that it is worth the time and effort to get the programs working properly.

READ ONLY

Spectrum +2 Machine
Language for the
Absolute Beginner
Melbourne House
£8.95

This book contains the best and clearest explanation of the 260 instructions that I have ever come across. In that respect it is an excellent book for the beginner, but the author seems to have forgotten that the newcomer to machine code also needs to be shown how to string the instructions together to make the computer actually do something. The book is peppered with short routines, but most of them simply state that if you load a couple of registers with numbers and add or subtract them, you get the right answer. The beginner might be forgiven for thinking that you could do most of this in half the time with a calculator.

There is an excellent chapter on hex, binary and decimal arithmetic, and it shows clearly how to calculate the conversions from one to the other, which is just as well. Most of the examples are in decimal, there is a decimal loader and no hex loader, but the tables of opcodes and instructions are in hex and there is no conversion table. It seems a certain recipe for disaster to expect the beginner to calculate conversions between conventions.

There is a splendid chapter on interrupts which makes the use of vector addresses (initially) clear, but again the examples only show how to load

it to a table on every interrupt, and the chapter ends by saying that there are other things you can achieve with interrupt chain routines, but they need some programming experience.

The author states briefly over keyboard reading and printing to the screen, but the chapter on sound makes the whole book worthwhile. At last we are given courses which explain something and show the reader how to build on the examples to make their own routines. This chapter, and the sections on memory paging, will be useful to anyone, even the more experienced machine code user, who is new to the ZX4 and +2. Despite the title, I think Joe Pritchard is more familiar with the ZX4 model because he talks of simplifying the sound via the expansion boards which, of course, the +2 does not have, and there is no mention of the +2's sound generator.

If you are new to the ZX4 or +2, then get this book. If you are a machine code novice, buy it for its clarity, but get another of the same title with better examples. This one tells you all about 260 instructions but nothing much about programming.

Conor Brookbank



STREAMS AND CHANNELS

This article contains not one, but two new channels which you can use on your Spectrum. The first is a channel which enables users of the Spectrum 128 (or the 128+2) to be able to use the ZX Printer — or some other compatible printer such as an AlphaStar — even in 128K mode.

If you don't have a Spectrum 128 then you don't need this new channel at all, because you can use the ZX Printer anyway.

The second is not really a new channel at all, but a modification to an existing channel. It enables the Spectrum to be able to

If you thought that you couldn't use the ZX Printer with your new 128 then Tony Baker can prove you wrong.

communicate freely with a GL via the local Area Network available from the ZX Interface One. Of course communication between these computers via the network is already possible, and David Hozosnik has recently been doing an excellent series in ZX Computing on that very task. It is not my intention to duplicate any of his material, but merely to remove a couple of deficiencies in the channel itself. We'll return to this later meanwhile back to the ZX Printer.

Figure 1

12 + 00	[1 byte] =	0,0000	Address of channel 128 origin.
12 + 01	[1 byte] =	00000,0	Address of channel 128 target.
12 + 02	[1 byte] =	0,0000	Start of channel (in 128).
12 + 03	[1 byte] =	0,0000	Default 128K mode/128K use channel.
12 + 04	[1 byte] =	0,0000	Length of clear buffer routine.
12 + 05	[1 byte] =	0,0000	Length of channel info (in 128K).
12 + 06	[1 byte] =	0,0000	Header flag.
00 01	Set of leading zeros will read for keywords, next instruction.		
00 02	Set word.		
00 03	Set word.		
00 04	Set word.		
00 05	Set of 128K status in 0K, next instruction.		
00 06	Set of 128K status in 0K, next instruction.		
00 07	Set of exactly two additional parameters required, next instruction.		
00 08	Set of exactly two additional parameters required, next instruction.		
12 + 00	[1 byte] =	0,00000	0 constants of print size in buffer.
12 + 01	[1 byte] =	0,0000	Start of buffer (in 128).
12 + 02	[2 bytes] =	0,0000	See printer buffer.
0000	0,0000	00 0000	is a one-byte wide buffer.
01		00 0	
02		00 0	before destination of buffer starts.
000	0,00000	00 000	
01		000 0	Start address of channel info.
10000		00 00,0000	
00		00 0,0	is: points to see printer buffer, handle interrupt.
01		00	
0000		00 0,0	
01	0,0,0,0,0	000 00	
00000		000 000,0,000,0,000	Output var var to 00 00,000.
01		00 00	
0000		000 0,0,0,0,0	Output all right now.

one, the best possible place for this buffer would be as part of the channel information block for the channel — that would mean that the area occupied by the new printer buffer would be relinquished whenever you closed the channel.

We shall call this new channel "Z" for ZX Printer. Take a look at Figure One — it shows the structure of the channel information block for the new channel. As you can see, the first eleven bytes store complete standard information, using the standard which was developed throughout this series. (ZX+60766 contains the constant 0000, which identifies this as being a user-defined channel. This means that it may be opened or closed using some of the software listed last month (ZX+000, ZX+00C) and ZX+000 also store information in the same format as the "W" channel gives last month — that is so that we can support more of last month's subroutines. As you shall see, a collection of new channels is far more advantageous than a single new channel.

Firstly, the new printer buffer itself runs from (ZX+00) onwards, and is 1024 bytes in length (most of the old printer buffer) — the variable Z_LENGTH (ZX+00C) will be zero if the printer buffer is completely empty or non-zero otherwise. If the new buffer is full it will contain 20h Z_LENGTH (ZX+00C) minus constant of 20h — the width of the buffer. It is so that the W0 and control-control routines last last month may be expanded successfully.

The program begins at the label Z_OPEN, which is the routine to perform all of the peripheral tasks necessary to open the channel. In fact this merely consists of testing whether or not the buffer is empty and printing a newline if it isn't.

Newline

The Z_NEWLINE routine may be compared to the ROM routine COPY_BUFFER at address 00C0. Its purpose is to transfer the contents of the buffer to the ZX Printer last, before finally erasing the previous buffer contents. The routine Z_EMPTY does the actual erasing. You will notice a couple of differences between the ROM routine COPY_BUFFER and the routine Z_NEWLINE. First, the new printer buffer is used instead of the old one, and secondly the subroutine Z_COPY_LINE is called instead of the original COPY_LINE.

In fact Z_COPY_LINE is itself very similar to the ROM's COPY_LINE routine (at 00F4). The only difference in evidence is

0000		LD A,00	
0001		RD (00),A	Fetch printer buffer size.
00		LD	Buffer information.
0005	Z_LENGTH	LD B,00	
00		LD B,Z	Use address of channel info.
0008		LD B,(00)	
00		LD B,(B,0)	Use pointer to use printer buffer, 10-00.
00		LD B,Z	
00	Z_LENGTH+000	LD (00),A	Store next byte of buffer.
00		LD B,0	Use pointer to next byte in buffer.
000E		LD B,(Z_LENGTH+000)	Use value of buffer.
00000000		LD Z,(Z_LENGTH)	Signal "flashing" open not required?
00000000		LD (Z_LENGTH),A	Send a newline to buffer.
00		LD	Return.
00		LD (00)	
00	Z_LENGTH+100	LD A,Z	Use new buffer to output.
0000		LD B,0	
00		LD A,Z	Use BZ (last 2 ones) 00-00 (overflow)
0000		LD B,0	Use BZ (last 2 ones) 00-00 (overflow)
0000		LD (00),A	Fetch printer buffer size, but still
			Use open for last two ones.
00		LD B,Z	Use "last 2 ones" flag
			Use BZ,00.
0000	Z_LENGTH	LD (00),A	Send channel status 0000 printed.
0000		LD A,Z	
0000		LD (00),A	Fetch printer buffer size.
00		LD	Buffer information.
00000000		LD B,(Z_LENGTH)	Copy the new printer buffer.
0000		LD (00),A	Output B, 0000 — 0000 repeat.
00	Z_LENGTH+000	LD A,(Z)	Use value of address.
00		LD B,0	
0000		LD B,(Z_LENGTH)	Include BZ pointer not corrected.
00000000		LD (00),A	Send next printer buffer to next.
00000000		LD (Z_LENGTH+00),A	Copy to print next one from buffer.
00		LD (00)	
00000000	Z_LENGTH+000	LD (Z_LENGTH)	Use pointer to head of channel information area.
00000000		LD (00),A	
00		LD A,Z	Use new of channel to search for.
0000	Z_LENGTH+000	LD (00),A	Use pointer to next char info area.
0000		LD B,0	
00		LD B,Z	Erases with copy set if channel not found.
00000000		LD A,(Z_LENGTH)	Use new of channel printed to.
00		LD B,0	
00		LD B,Z	Erases with BZ printing in channel information area, and copy next, if search successful.
00000000		LD B,(Z_LENGTH)	
00000000		LD A,(Z_LENGTH)	Use length of channel info.
0000		LD (Z_LENGTH+00)	Copy back to routine search.
00000000	Z_LENGTH	LD (00)	
00000000	Z_LENGTH	LD (Z_LENGTH)	Use pointer to channel info's area.
00000000		LD B,(Z_LENGTH)	Use BZ (overflow, control pointer, VLA, and some control).
00		LD B,0	Return if these complete.
00		LD B,Z	
0000		LD A,Z	
0000		LD A,(Z_LENGTH)	Copy with printer characters.

000	IF $HL, \text{L_OPEN}$	Jump with HL to L_OPEN_OPEN .
0000	DB 00	
0001	IF $HL, \text{L_CLOSE}$	Jump with HL to L_CLOSE .
0002	CALL L_WRITE	Write a message.
0003	IF L_OPEN	Go jump to L_OPEN .
0004	DB 14	
0005	IF L_CLOSE	Jump with contents of HL to L_ .
0006	DB 1,00	In the HL there are.
0007	IF L_L_OPEN_OPEN	Jump with "success content".
0008	DB 4	
0009	DB 0,00	In the HL there are.
000A	IF L_L_L	Jump with HL to L_L_L .
000B	DB 4,4	In HL (bit 0 or OPEN), or FF (bit 1 or CLOSE).
000C	DB L_L_L_L	In "flag" byte, the complement of parameter to use.
000D	DB 0	In all bits reset, except bits 0, 1, 2 (OPEN) or bit 3 (CLOSE) will be taken from "flag" byte, but complement of parameter to use.
000E	DB L_L_L_L	In 3 (OPEN) or bit 2 (CLOSE) of "flag" byte will be compared with parameter, only to test.
000F	DB L_L_L_L	
0010	IF L_L_L	Jump with contents of HL to L_L_L .
0011	DB 1,0	In HL there are.
0012	IF L_L_L	Jump with contents of HL to L_L_L .
0013	DB 0,0	In HL there are.
0014	IF L_L_L_L_L_L	Jump with HL to L_L_L_L_L_L .
0015	DB 0,0	In HL there are.
0016	IF L_L_L_L	Jump with HL to L_L_L_L .
0017	DB 0,0	In HL there are.
0018	IF L_L_L_L	Jump with HL to L_L_L_L .
0019	DB 0,0	In HL there are.
001A	IF L_L_L_L	Jump with HL to L_L_L_L .
001B	DB 0,0	In HL there are.
001C	IF L_L_L_L	Jump with HL to L_L_L_L .
001D	DB 0,0	In HL there are.
001E	IF L_L_L_L	Jump with HL to L_L_L_L .
001F	DB 0,0	In HL there are.
0020	IF L_L_L_L	Jump with HL to L_L_L_L .
0021	DB 0,0	In HL there are.
0022	IF L_L_L_L	Jump with HL to L_L_L_L .
0023	DB 0,0	In HL there are.
0024	IF L_L_L_L	Jump with HL to L_L_L_L .
0025	DB 0,0	In HL there are.
0026	IF L_L_L_L	Jump with HL to L_L_L_L .
0027	DB 0,0	In HL there are.
0028	IF L_L_L_L	Jump with HL to L_L_L_L .
0029	DB 0,0	In HL there are.
002A	IF L_L_L_L	Jump with HL to L_L_L_L .
002B	DB 0,0	In HL there are.
002C	IF L_L_L_L	Jump with HL to L_L_L_L .
002D	DB 0,0	In HL there are.
002E	IF L_L_L_L	Jump with HL to L_L_L_L .
002F	DB 0,0	In HL there are.
0030	IF L_L_L_L	Jump with HL to L_L_L_L .
0031	DB 0,0	In HL there are.
0032	IF L_L_L_L	Jump with HL to L_L_L_L .
0033	DB 0,0	In HL there are.
0034	IF L_L_L_L	Jump with HL to L_L_L_L .
0035	DB 0,0	In HL there are.
0036	IF L_L_L_L	Jump with HL to L_L_L_L .
0037	DB 0,0	In HL there are.
0038	IF L_L_L_L	Jump with HL to L_L_L_L .
0039	DB 0,0	In HL there are.
003A	IF L_L_L_L	Jump with HL to L_L_L_L .
003B	DB 0,0	In HL there are.
003C	IF L_L_L_L	Jump with HL to L_L_L_L .
003D	DB 0,0	In HL there are.
003E	IF L_L_L_L	Jump with HL to L_L_L_L .
003F	DB 0,0	In HL there are.

STREAMS AND CHANNELS

the effect of passing the NAME key — in my routine L_PRINT is called instead of WRITE_FILE . This removes any possible chance of corrupting the old printer buffer area.

A general purpose subroutine is included next — SEARCH_CH_ALL . If it is entered with the A register containing the name of a channel, the routine will search through the channel information area looking for a channel with the name. If it finds one it will return with it pointing to the channel information block and the carry reset, otherwise the carry will be set. We shall make use of this subroutine later.

Z_PRINT

Z_PRINT is the routine which "prints" a character held in the A register onto the new printer buffer. Note that the subroutine of address HDF comes from last month's article and will sort out all keywords, control characters, and will deal with both the carriage control and the "no" function. On entry into the subroutine the C register will contain one WRITE or OVER parameter, or the X coordinate of any AJ parameters. The Z_PRINT routine itself is really quite simple, keeping in mind that it has to deal with ASCII characters, block graphics, and MODE , as well as WRITE , WRITE , AJ and WRITE . Follow it through to see how all the various cases are dealt with.

Finally (for this channel) we have Z_OPEN , which opens the channel and attaches it to a device. On entry the A register must contain the device number to which the channel is to be attached and the subroutine will do the rest. It makes use of another subroutine from last article — the OPEN_NEW routine which will open user defined channels.

Channel Q

When communicating with the QL via the normal network channel you will notice a couple of problems. The first is that on the Spectrum the code for "enter" is OD (hex), whereas on the QL the code for "enter" is OA (hex). This means that, for instance, PRINT AJ on the QL cannot be matched by INPUT AJ on the Spectrum, the vice versa. It is normally necessary to use PRINT AJ , WRITE on the QL in order that the Spectrum may use INPUT AJ . Conversely, INPUT AJ on the QL needs to be matched by PRINT AJ , WRITE on the Spectrum.



What I intend to do is to modify the "M" channel attached to a particular string so that the deficient one removed is a modified "M" channel. These problems disappear FROM AS on the Spectrum may be matched by INPUT AS on the QL, and vice-versa. Other surprising advantages turn up. If you left a Spectrum program over a modified network channel than the QL, will be able to LOAD the program with the single command LOAD NET_2 (assuming that the Spectrum is station number 2) - or to transfer the program directly onto a QL. Alternative ready for loading later with the single QL command COPY NET_2 TO MAIN_1.PROG.

The second problem is a little more complex, but not much so. The Spectrum command INPUT AS actually expects to receive a string expression, not the contents of an actual string. However, this is not the case with INPUT LINE. Consequently for us, we seldom notice this. Because the Spectrum provides the surrounding quotes - this converts the string text into a string expression and everything is fairly easy - with one exception: if the string pointed over the network by the QL contains a quoted character then the Spectrum will be unable to evaluate the result of a string expression and will halt with error C. Response in BASIC. The cure to this problem is very simple. If, during Spectrum INPUT or INPUT LINE a quoted character is received from a modified network channel THEN IT MUST BE converted to two CHARACTERS. THE QUOTED CHARACTER, in order that the expression will still evaluate.

Let's take a look at the subroutines now. We begin with SHADOW_LOAD which will call a subroutine in the Shadow ROM whose address is DE. It performs this by loading the Spectrum into memory and we are returning to the Shadow ROM with a Spectrum subroutine, and then simply copying out the Shadow ROM on completion.

Q_PRINT is the routine which "prints" a character over the modified network. As you can see it is extremely simple, merely testing for an "error" character and converting such to OA. Before using the original Shadow PRINT subroutine to print the character.

Q_INPUT is more complicated, though not much so. Really, the routine tests the error return address to see whether the character input is due to an INPUT command, or due to an INKEY function. If it turns out to be merely INKEY?

LD	LD A,(DE)	Use byte from buffer.
000000	LD A,(A,DE+1)	
0001	LD A,(A,DE+2)	Copy of error flag.
0002	RR A	
0003	RR A,(A,DE+1)	
0004	LD A,(DE+2)	Copy of Spectrum no.
0005	LD	
0006	RR (A)	Use byte from program from, with error and Spectrum data into account.
0007	LD (DE),A	Store in buffer.
0008	RR A	Check pointer into program from.
0009	LD A,(DE)	
0010	RR A,A	Use pointer to appropriate byte in buffer for next use of character.
0011	RR A	Use pointer into program from.
0012	RR A	Use pointer to next byte to use from main character into buffer.
0013	RR A,(A,DE+1)	Use character just printed.
0014	RR A	Send the next flag.
0015	RR	Return.
0016	RR (A)	
0017	LD A,(DE)	Use stream number to attack character.
0018	LD A,(DE)	
0019	RR (A,DE+1)	Search for another "Q" character.
0020	LD A,(A,DE+2)	Send to the output.
0021	LD A,(A,DE+1)	Use character into buffer.
0022	LD A,(A,DE)	Use output address.
0023	LD (A,DE),A	Use input address.
0024	LD A,(A)	Use length of channel in's.
0025	RR (A)	
0026	RR (A,DE)	Open the channel.
0027	LD (A,DE),A	Send the flag.
0028	LD (A,DE),A	Specify buffer width - 255 char.
0029	LD (A,DE),A	Start the buffer, and return.
0030	RR (A)	
0031	LD A,(DE),A	Start string address to shadow ROM.
0032	LD A,(DE),A	Start Shadow character address.
0033	LD A,A	Use error.
0034	LD A,A	Start error signalling feature to shadow ROM address.
0035	LD A,A	Jump to call Shadow subroutine.
0036	LD A,A	
0037	LD A,(DE),A	Use pointer to channel information.
0038	LD A,A	Send character to print.
0039	LD A,A	
0040	LD A,(A,DE+1)	Jump when character in "buffer", use Q's code for "buffer".
0041	LD A,A	
0042	LD A,(DE)	Use Shadow output address.
0043	LD (A,DE),A	Use program output to Shadow ROM.
0044	LD A,A	Use character just printed.
0045	LD A,A	Send the next flag.
0046	RR	Return.
0047	RR (A)	
0048	LD A,(DE),A	Use pointer to channel information.
0049	LD A,(DE),A	Use pointer to error return address.
0050	LD A,(A)	
0051	LD A,A	
0052	LD A,(A)	Use error return address.
0053	LD A,(A,DE)	
0054	LD A,A	
0055	LD A,(A,DE),A	
0056	LD A,A	

STREAMS AND CHANNELS

0000		00 00,00	
0004		00 0,0,0000,0	Copy of low 0000 channel.
0008	0,0,0000	00 0,0(0000)	
000C		00 0,0(0-00)	
0010		00 000,0000,00	000 address input from shadow 000.
0014		00 0,0,0	000 character just input.
0018		0000 00	Mask the flag.
001C		00 00	
0020		00 00,0,0000,0	Copy address into A to do "rotate".
0024		00 0,0,0	Use "rotate" "rotate" character.
0028		00 00,00	Remove the flag.
002C		00 0,0,0	Use character to address.
0030		0000	Set zero.
0034	0,0,0000,0	00 00,0(00,0)	Repeat the channel count down to 000 or 00,0000.
0038		00 00,00	Done the address 00,0000.
003C		00 0000	Now count over-rotate address plus.
0040		00 000,0(0,0)	Remove pointer to over-rotate addr.
0044	0,0,00,000	000 000,0(00,000)	Load 000 character from network.
0048		00 0,0,00,0000	Copy 00 character 00.
004C		00 0,0,00,0000	Copy 00 will writing.
0050		00 000000 00	Repeat 0, 00 of 000.
0054	0,0,00,0000	00 00	
0058		000 0	Address (0000 to now finished) if character is "rotate".
005C		0000 000,0(00,000,0)	Draw the character to the 0000 000.
0060	000,0000	00 0,0(0,0,0)	
0064		00 00,0,00,0000	Low byte of 0000 0000 0000.
0068		00 00,00	
006C		0000 0,000,000,000,0)	Build-up 0000 character.
0070		00 0,0,00,0000	Go back to end of 0000.
0074		000 0000	
0078	0000,0	000 000,000,000,0)	Use 0000 data for stream channel.
007C		00 0,0,0	
0080		00 0,0	
0084		00 0,000,0000	Draw 00 character out to net.
0088		00 00,0(0000)	Use pointer to base of channel information area.
008C		000 00,0,0	Use pointer to last byte of read info.
0090		000 00,0	
0094		000 00,0	
0098		000 00,0	Use pointer to 000 byte of read info.
009C		00 0,0,0)	Use base of channel.
00A0		00 00	
00A4		00 0,000,00	Copy into B channel on "0".
00A8	000,0000	000 000000 00	Repeat 0, 0000 channel.
00AC		000 00	
00B0		00 00,0,0	
00B4		00 00,0,0	
00B8		00 00,0,0	
00BC		00 00,0,0	
00C0		00 00,0,0	
00C4		00 00,0,0	
00C8		00 00,0,0	
00CC		00 00,0,0	
00D0		00 00,0,0	
00D4		00 00,0,0	
00D8		00 00,0,0	
00DC		00 00,0,0	
00E0		00 00,0,0	
00E4		00 00,0,0	
00E8		00 00,0,0	
00EC		00 00,0,0	
00F0		00 00,0,0	
00F4		00 00,0,0	
00F8		00 00,0,0	
00FC		00 00,0,0	
0100	0,0,000	00 000	Use stream reader to address.
0104		00 0,0,0	Copy to zero channel "0" and attach it to address area.

These things are just as simple as for Q_0000. The shadow input routine is called by the read character, and if that character is OA then it is converted to "rotate".

If streamin we are dealing with INPUT (as opposed to INKEY) then the first thing we must do is address (rotate) use of the 0000 routine in the ROM which we do by entering the machine stack as far as the main address from the 0000 call and then supplying a new routine to replace 0000. We can call Q_0000 as a sub routine to fetch a single character from the network (returning OA to 000) and the flag will tell us if the character is valid or not. Then we can call a ROM subroutine.

ADD CHAR_1 to actually insert the character into the input line. Quote characters must be inserted twice, not once (except during INPUT 000). Finally, if an "enter" character is received then we may return, or if from the 0000 routine with the finished string expression in the INPUT area of RAM.

MC0007_N is the routine which will convert a "N" channel to do as it told. On entry the A register must contain the stream number of an existing "N" channel which must have been opened normally. In fact a channel (00+0000) and 00+0000 will normally contain the value 0000 which, if called, will page in the shadow ROM and then call the actual output to input subroutine addressed by 00+0000 and (00+0000) respectively.

The MC0007_N subroutine will merely convert these 0000 to the addresses of Q_ROM and Q_INPUT (leaving the addresses of the shadow routines unchanged). The shadow routine will then only be called when required by Q_ROM or by Q_INPUT.

Finally a five byte demonstration of the "Z" channel. Simply calling Z_0000 from 0000 will open a new channel "Z", and attach it to stream number four.

Channel B

In the next and final part of this series we shall examine the use of the existing channel "B" which may only ever be used in machine code. We shall also be creating a new channel, also called "B" (if for RAM-disk) which will enable you to create RAM and WRITE files in the Spectrum (B) RAM disc area, just as you can on the harddrives. Be then, keep smiling, and don't lose anything seriously.

■ His mouth was moving into the big line! That is as far as the size of illustrations goes. No more little snippets. Now we're talking about full screen illustrations.

But fast the bad news: big pics mean big memory blocks to store them — 800 bytes (memory space) to be precise if you want them in glorious Monochrome II style, enlarging a "Mugs" style comic strip drawn by a relatively short program. This may not matter left alone for the moment if cost is, and see how we cut size and recall the illustrations.

And at all, how many can you store in memory? Answer — 5, with about 84 left for the driver program (128 bytes will obviously do much better than that). Next, how do you store them? In principle, what you have to do is to transfer the contents of each byte of the display (the D_PXL) and attributes (the ATTR) when the screen picture is stored in high memory, knowing where you're getting it. You could do this by reading each byte, then flipping the contents up, but this would be quite slow. The solution is a short machine code routine which uses the instruction LDH. This stands for load high memory address special which probably doesn't

Table 4.

0	1	2	3	4	5	6	7
278	40	52	274	0	42	22	112
0	2	22	224	42	22	224	22
27	2	42	2	42	22	42	224
274	2	22	224	22	22	22	22
22	2	22	22	22	22	22	22
27	22	42	2	22	22	22	22
2	2	22	22	2	22	22	22
27	2	22	22				

make you much the wiser. In essence, you set one counter to the number of bytes to be transferred, another to the start of the D_PXL, and a third to the destination address. LDH does the first, continuously transferring bytes until the first counter is reduced to zero.

The necessary code is included in Program 4, which will make the transfer for you, modify the code to make it work in reverse, then save it along with the picture bytes. Type it in, and let's try it out. If you have some pic ready (made with a drawing utility such as Tom Sawyer's Light Screen Designer) you could use these: 01=write the "rainbow" and legal SCREENS at the beginning of your "horizontal" tape will be quite satisfactory.

Run the program, and you'll be asked how many pics you want to save. Let's try two. Answer the prompt for a title with the first pic, then play the tape. The picture will load in more or less instantly and be transferred to high memory; the address of which will be given. Make a note of this, and the POKE number as well. You'll need this information later. Repeat for the second picture, and there you are. You can then save the data for the two pics plus the machine code to drop them back down.

To make use of this routine you will need a subroutine in your program such as: **POKE POKE ADDRESS, %RANDOMIZE USR ADDRESS, RETURN**

Where you have an variable X to the POKE number you noted for that picture, before you SCREEN %RANDOMIZE USR POKE X, but don't think of the wrong moment. You'll miss it.

Memory squeeze

How all this is easy well if you have a "128" or don't need other pics. How can you squeeze the presented data into the Spectrum's pint pot? One way is to have a smaller quant in other words, perhaps you would like to copy with a few colour drawing (RGB and RGB), which takes up 256 bytes less. You could go further by having only a top third, or two heights (store picture frame coordinates) use this system. Storing the bottom of the screen (close by first) the RGB given in Program 1 will tell you the changes you will need to make for these variations. If you want to mix several types of illustrations, in other programs you'll need to POKE 65544 with the appropriate value shown in the line 180 6544 before you call the routine.

This technique obviously doesn't help if you really want full screen illustrations, but there is a way around that problem if you're prepared to sacrifice a little speed for a great saving in space. If you type in the following: **LOAD "SCREENS", POKE 65544 TO 65596, PRINT IN 0=Address "SCREENS", PAGE 1, PAUSE 20, NEXT 1**

Press ENTER, then LOAD in a picture. You'll see a series of numbers appear at the bottom of the screen as the program fills it away through the D_PXL, and later the ATTR if you wait long enough. You'll notice that the numbers 0 and 255 (and later the gamma attribute, for example 56 if the background is black on white) occur more often than any others. This is because most of a picture is either blank space (0) or white (255), which much of the attribute and gamma unchanged. Knowing this, it is possible to compress the data for a picture by storing, for example a line of 32 zeros as 032. Using this technique even a complex picture, such as the "rainbow" SCREENS is compressed to about half its usual length. This is what the compressor routine (Program 3) does, here it is and let's try it out.

Run the program, and answer



RANDOM MEMORY

PROGRAM 1

```

10 REM PICSAVE
20 INPUT "No. of pics to store";N
30 POKE 53756,N LET A=53757-N
4012-12:POKE repeat 6912 with
5040 for top third of SCREEN.
6000 for top third, 6144 for
FILE only
70 CLEAR; LET N=PEEK 53756
LET A=53757-N:6912-11:POKE 6912
replaced as above
80 LET A=6912 PRINT "CLEAR"
90 FOR I=1 TO N: INPUT "Title
of pic " I";T:AS LOAD A":I
:ASCENDING
100 RANDOMIZE A LET L=PEEK 5336
70 LET N=PEEK 53671:POKE 53359,N
80 POKE 53358,N:RANDOMIZE USP
60358
90 PRINT "Data start for pic -
" I";I="A":POKE A,L:1:1
LET A=A+6912:POKE repeat 6912
as before
100 NEXT I:POKE 53358,33:POKE
53359,47
70 STOP SAVE "pics"CODE 8.N+
6912+12
100 DATA 17,0,0,33,0,64,1,0,27
27,176,201:POKE repeat 27 with
34 for FILE only, 18 for top
two thirds, 0 for top third only
9999:Call with - POKE 53360,
high byte of data start address.
RANDOMIZE USP 60358

```

PROGRAM 2

```

1 REM COMPACTOR
2 INPUT "Title" :AS IF A=""
THEN STOP
30 INPUT "Attribute" :AS AT
40 POKE 53756,N:POKE 53757,N
50 LET A=53756-N
60 POKE 53756,N:POKE 53757,N
70 LET A=53756-N
80 POKE 53756,N:POKE 53757,N
90 POKE 53756,N:POKE 53757,N
100 IF A="" THEN GO TO 1000
1100 IF A="" THEN GO TO 1000
1200 IF A="" THEN GO TO 1000
1300 IF A="" THEN GO TO 1000
1400 IF A="" THEN GO TO 1000
1500 IF A="" THEN GO TO 1000
1600 IF A="" THEN GO TO 1000
1700 IF A="" THEN GO TO 1000
1800 IF A="" THEN GO TO 1000
1900 IF A="" THEN GO TO 1000
2000 IF A="" THEN GO TO 1000
2100 IF A="" THEN GO TO 1000
2200 IF A="" THEN GO TO 1000
2300 IF A="" THEN GO TO 1000
2400 IF A="" THEN GO TO 1000
2500 IF A="" THEN GO TO 1000
2600 IF A="" THEN GO TO 1000
2700 IF A="" THEN GO TO 1000
2800 IF A="" THEN GO TO 1000
2900 IF A="" THEN GO TO 1000
3000 IF A="" THEN GO TO 1000
3100 IF A="" THEN GO TO 1000
3200 IF A="" THEN GO TO 1000
3300 IF A="" THEN GO TO 1000
3400 IF A="" THEN GO TO 1000
3500 IF A="" THEN GO TO 1000
3600 IF A="" THEN GO TO 1000
3700 IF A="" THEN GO TO 1000
3800 IF A="" THEN GO TO 1000
3900 IF A="" THEN GO TO 1000
4000 IF A="" THEN GO TO 1000
4100 IF A="" THEN GO TO 1000
4200 IF A="" THEN GO TO 1000
4300 IF A="" THEN GO TO 1000
4400 IF A="" THEN GO TO 1000
4500 IF A="" THEN GO TO 1000
4600 IF A="" THEN GO TO 1000
4700 IF A="" THEN GO TO 1000
4800 IF A="" THEN GO TO 1000
4900 IF A="" THEN GO TO 1000
5000 IF A="" THEN GO TO 1000
5100 IF A="" THEN GO TO 1000
5200 IF A="" THEN GO TO 1000
5300 IF A="" THEN GO TO 1000
5400 IF A="" THEN GO TO 1000
5500 IF A="" THEN GO TO 1000
5600 IF A="" THEN GO TO 1000
5700 IF A="" THEN GO TO 1000
5800 IF A="" THEN GO TO 1000
5900 IF A="" THEN GO TO 1000
6000 IF A="" THEN GO TO 1000
6100 IF A="" THEN GO TO 1000
6200 IF A="" THEN GO TO 1000
6300 IF A="" THEN GO TO 1000
6400 IF A="" THEN GO TO 1000
6500 IF A="" THEN GO TO 1000
6600 IF A="" THEN GO TO 1000
6700 IF A="" THEN GO TO 1000
6800 IF A="" THEN GO TO 1000
6900 IF A="" THEN GO TO 1000
7000 IF A="" THEN GO TO 1000
7100 IF A="" THEN GO TO 1000
7200 IF A="" THEN GO TO 1000
7300 IF A="" THEN GO TO 1000
7400 IF A="" THEN GO TO 1000
7500 IF A="" THEN GO TO 1000
7600 IF A="" THEN GO TO 1000
7700 IF A="" THEN GO TO 1000
7800 IF A="" THEN GO TO 1000
7900 IF A="" THEN GO TO 1000
8000 IF A="" THEN GO TO 1000
8100 IF A="" THEN GO TO 1000
8200 IF A="" THEN GO TO 1000
8300 IF A="" THEN GO TO 1000
8400 IF A="" THEN GO TO 1000
8500 IF A="" THEN GO TO 1000
8600 IF A="" THEN GO TO 1000
8700 IF A="" THEN GO TO 1000
8800 IF A="" THEN GO TO 1000
8900 IF A="" THEN GO TO 1000
9000 IF A="" THEN GO TO 1000
9100 IF A="" THEN GO TO 1000
9200 IF A="" THEN GO TO 1000
9300 IF A="" THEN GO TO 1000
9400 IF A="" THEN GO TO 1000
9500 IF A="" THEN GO TO 1000
9600 IF A="" THEN GO TO 1000
9700 IF A="" THEN GO TO 1000
9800 IF A="" THEN GO TO 1000
9900 IF A="" THEN GO TO 1000
10000 IF A="" THEN GO TO 1000

```

Now you have your compacted codes separately on tape you need to save them as one long code length. Do this

the "title" prompt if you haven't got the table I supplied in the January column, you can calculate it yourself. Say the picture is drawn in blue ink on yellow paper; the value would be 1 (ink) + 4 (paper) = 5-49. Now load in the SCREEN and will have one on the program load over and the computing may take some time, so go and make a cup of coffee, or have a stroll around the garden. You've been harassed over that VDU for too long anyway! When the transfer is complete the number of bytes that the picture has been compressed into is displayed. Make a careful note of this and the file you use to save the compacted code to tape. Repeat with a new attribute value and SCREEN until you have all you need, then just reply to the "title" prompt with END.

Clyde Bish presents tips on full screen graphics.

Table 2

	LD	HL, 1000	
	LD	DE, 00000000	(POKEs before call)
LOOP:	LD	A, (HL)	
	CP	000	check for filled byte
	JR	Z, FILL	
	CP	0	check for blank
	JR	Z, R000	
	LD	(HL),A	
	INC	HL	
	DEC	DE	
	LD	A, 0	
	CP	00	100 for 2 1/2 screens, 10 for 1 1/2 screens
	JR	C, LOOP	
LOOP1:	LD	A, (HL)	
	CP	attribute	replace with SET if FILE only
	JR	Z, R000	attribute start here, POKE after using
	LD	(HL),A	
	INC	HL	
	DEC	DE	
	LD	A, 0	
	CP	01	
	JR	C, LOOP1	
	SET		
R000:	INC	DE	
	LD	A, (HL)	provide to ship attributes not to be altered
	POP	HL	
	LD	B, 0	
	LD	C, 0	
	LD	D, 0	
	LD	E, 0	
	LD	F, 0	
	LD	G, 0	
	LD	H, 0	
	LD	L, 0	
	LD	(HL),A	
	POP	HL	
	JR	LOOP	
R100:	INC	DE	
	LD	A, (HL)	provide to ship FILE bytes not to be altered
	POP	HL	
	LD	B, 0	

Q-Liberator

David Nawotnik reviews
the latest BASIC
compiler for the QL.

**Q-Liberator
Liberation Software
\$69.95**

Not long ago, there was only one BASIC compiler for the QL, now there are three. The first, SuperCharge, was reviewed in ZX Computing a year ago. Next on the scene was Q-Liberator, closely followed (if not, in theory) by Turbo from Digital Precision.

Q-Liberator is supplied in an attractive white-cassette-sized case, with two microfiche cartridges and an 80 page manual. Of the two cartridges, one is a fully working copy; the other is a 'reader'. The ingenious security protection system in Q-Liberator prevents users obtaining working copies of one of the essential compiler files by all means apart from use of the reader. The user can obtain up to five working copies from this reader, configured to any chosen storage device. Once a working copy is created, it can be used without hindrance from any other security system.

Using Q-Liberator is simpler than the basic program loads the compiler, run time routines and SuperBASIC extensions used by the compiler. Once these are installed, the BASIC program to be compiled is loaded as normal, and the program compiled with the 'LIBERATE' command.

Q-Liberator compiler is two-pass; the first pass produces a working copy of the BASIC program, the second pass produces the compiled code. The user can select the second pass to occur automatically after the first, if sufficient RAM is available.

There are several options available for the compiled code. In program development, Q-Liberator will produce a line number table, disk memory usage statistics, which help to identify errors. These options can be suppressed in producing the final version, to save space. The sun area option can be incorporated into the final code, producing a totally independent program, or they can be lost. Despite the latter option you can save a lot of space if you have

several Q-Liberator compiled programs running concurrently. Extended SuperBASIC commands can be incorporated into the Q-Liberator produced code, up to eight machine code sets of new commands can be added, but you'll need to have a little knowledge of these routines (the start of the initialisation routine, and the start of the name table relative to the start of the code) to supply the compiler with the appropriate parameters. Other compiler options include setting heap, stack, and input buffer sizes, and switching on/off integer arithmetic modes. These options are available by inserting single ROM lines into the SuperBASIC program.

A number of SuperBASIC extensions are supplied with Q-Liberator for use specifically with the compiler. Of particular interest are commands to turn on/off the cursor in the compiled program, and a useful set of error trapping commands which can replace the mixed bag of commands in the various ROM versions of the QL.

The producers of Q-Liberator state the claim for their product that it is fully compatible with SuperBASIC (bearing in mind that there are several commands, e.g. LIB which make no sense in a compiled code) in reviewing the product several programs were compiled. All compiled versions worked well with little or no modification required to the BASIC programs to achieve successful compilation. The compiler picked up a missing END-SUBC in one program, which Q-Liberator did not seem to read the libcode area and warning messages allowed easy interpretation of this and other minor problems.

The first thing to note is the difference memory size can make to operating speed with more RAM. BASIC and compiled programs work faster. The test programs (compiled with Q-Liberator with all default options set) produced significant savings on everything except the ingenuity test. This is not surprising, as ingenuity calculations require a lot of ROM time - whether in BASIC or as a compiled program. Inclusion of the run time module into the Q-Liberator compiled program makes no significant difference to operating speed.

The time saving on integer math was disappointing.

SuperBASIC doesn't perform true integer mathematics; integers are converted to floating point numbers before they are used so a very large saving with use of variables with the '%_ suffix on compilation was to be expected. It seems that Q-Liberator will not correctly interpret the '%_ in a variable without the integer maths directive switched on.

Surprisingly with the integer directive, the simple loop takes longer to perform, so code is needed with this directive to optimize operating speed. Integer mathematics with trigonometric functions is invaluable, but the compiler compiled and ran integer file lengths without complaint.

For comparison, the same routine was compiled with SuperCharge and the operating speeds are about the same. SuperCharge has the edge on the simple loop and integer maths. Q-Liberator does slightly better with the FOR-NEXT loop and ingenuity test. SuperCharge failed to compile BASIC procedure correctly (it provided an integer result, but that had to be altered to provide results for comparison).

All file sizes were obtained using the MPQ option in IDE. Surprisingly, all Q-Liberator files were larger than the corresponding SuperBASIC file (even without incorporating the run time module), and their size not predictable from the size of the original SuperBASIC file. Incorporation of the run time module adds another 16 to the size of the compiled program. For the shorter jobs, BASIC file programs, compilation took 180 seconds with microchips and 116 seconds with RAM disks. In every case, loading the compiled program was much faster than the corresponding SuperBASIC program. This was particularly noticeable with the longer programs.

With its very close compatibility with SuperBASIC, and simplicity of use, Q-Liberator offers an excellent advantage over SuperCharge. And for those who simply want their SuperBASIC programs to load and work faster, Q-Liberator has a lot to offer. If it is not priced a little more reasonably,

Liberation Software, 43 Clifton Road, England upon Thames, Surrey

MINDPLAY

Peter Sweasey is thawed out temporarily to review Silicon Dreams and the first GAC adventures.

Though the idea of Month may be questionable as you read this, the magazine is currently several inches deep in snow. No PC bonuses for me, I just have to content myself with making show issue versions of Arpan, the barbaric editor, then jumping up and down on them.

It is so cold at the moment that my faithful Spectrum (this particular reader is over four years old) has been affected. It led in its usual position overnight but the keyboard behaves as if smelt off it permanently being pressed, the remedy is to place the machine in the snow cupboard for ten minutes.

Several months ago I plugged a then fledgling adventure club called Adventure Probe. Well I'm glad to see the magazine has grown considerably since then,

and you get a lot more for your £1 cover fee (or 75p if you take out a year's subscription). As Sandra Sharkey (the magazine's editor) says, reader feedback has inspired it tremendously. There are various hints, reviews and opinions, but what Probe seems especially strong on is help. Several pages are crammed full of readers' names, followed by lists of games that they are prepared to help with. Help and solutions can be bought from Probe themselves for the pitifully sum of a first class stamp (shared with each issue is Adventure Ships, which is a catalogue of amateur games for sale, some of which sound most intriguing. Something which particularly caught my eye was an ad for the Ra — an addition to the Gull which, though not written by Glast, adds full sentence interpretation and a proper WAS command. Generally interaction between readers seems better than in most such clubs, so if you're an adventure fiend, it could be worth buying a trial copy (send £1 for a sample issue to Adventure Probe, 13 Marton,

lead, Blythfield, Wigton WMB 640).

A product called the Adventure Builder System also received its first month courtesy of Ollie's occasionally feeble budget division, Alpha Omega. For just under a five, it may appear to offer a bonus Gull, but no such luck. It provides you with a set of routines and structures for you to write your game around — in BASIC. The coded routines are much better than BASIC would be, speed apparently being that language's main disadvantage. This really is a pleasing little GAC, the Gull and Ollie's soon-to-be-leaved Professional Adventure Writer all enable you to write much faster and evade complicated adventures, with almost the same flexibility as BASIC but less of the hassle. BASIC programming is time consuming and wasteful of memory — with wily Gull-type utilities were designed in the first place. Besides, ASB is only £4 cheaper than the recently re-priced Gull. If people want to write for pleasure, I think they will pay the extra, if they want to write commercially they'll have to



HELPLINE

I've been having a little new-year cleanout down here — and I've discovered something of a backlog of old problems to be dealt with. I'm sending apologies to anyone who has had to wait for a reply. I should have cleared the backlog by the time you read this, so you will not have to wait so long in future.

P. Lopez from Abingdon asks how to pay tribute to Nero in **Samurai Banzai**. Anthony Quinn kindly supplies the answer: simply **WAIL**. Another guide from Mark Hitchcock, who needs the words to climb out of the BR in Interceptor's **Affinity**, once he has removed the ceiling panel: it's **CLIMB OUT OF LIFT**.

David Brown of Standard-Le-Hope would like to enter the holy tower in **Occams' Never Ending Story**. As far as I know, it's impossible in Fort One — you certainly do not need to. At the end of Fort Three you'll find yourself outside it — **UNLOCK DOOR** with the gold key from part two found west from the future planet. Thanks to Charles James for a Never Ending solution.

Nigel Dawes — who describes himself quite accurately as 'the bore' who is still playing **Planet Of Death** — wants to know how to get the mirror without breaking it. From the cavern it housed in, go to the adjoining wind tunnel and use **GET** with the gloves you'll find there. With these, pick up the man and drop him. As a result, he'll be removed from the mirror, so you can shoot him without incurring bad luck.

Some **Hobbit** queries need someone called 'Vandepoorter' from Belgium willing to find the magic ring in the dark, stuffy passages. From the **DEATH** dungeon, go west out the window (use **SA** to walk for the goblin to appear), **N**, **S** and **E**. Use Acton from the Isle of Wight has broken through the system with the **Three Clowns**. How do I go through the magic door? You need to wear the ring, then **ERADIANE CRACK**, keep **WAITING** and watching the ring **obscurely**; eventually the door will open at an all-seeing **Go NE**. Immediately this happens, follow the door closes again. Forgive me, Mark McDonald, who'd like to kill the dragon without anyone's help. I think this is impossible: you need to pick up Lord at Long Leen already — **CHERRY SAUCE**. Then head to the dragon's treasure hall, avoiding him on your way if he's there.

when you arrive then **SA** to **WARD "SHOOT DRAGON"**. It he's not go east, wait for a few moments (waiting the ring might help), then east again and he should be there.

Adam Cochrane needs help in **Orion Through The Outback Gates**. The important thing is not to go out through the window to start with. **DESS**, take the letter, **radiation suit** and **Cherry Blossom**, move the chest from **Maggie's Den** to your own, take and wear the belt, then out into **Evening Street** and **BUY BOMBS**.

K. Ingham wants to board a train in **Double Boys The Moreby Jewels**. I've never played this so I'm indebted to the help of Christian John Hunter. In the garden of your house, **GET INVOLVE** (as making out your door), and **OPEN** in Go to the railway station, **GET** and **SLIP THROUGH**, **GET YOUR USE TOLLY** in **SHOW SCULL**, **ENTER PLATFORM** and **BOARD TRAIN "C"** is also stuck in **Elfo's Castle Gaidin**, which again I have not played, but which again John Hunter helps with. **USE GEMMADE** in the passage of rats. Avoid the radio tower, you don't need to visit it to complete the game. John also helped a reader out with **Games Of Endos** (the solution is too complicated to print here) so really thank indeed.

WIP

Anybody out there called 'Vero Cruz' or 'Questprobe 27' Please send me a solution if you have — I'll be happy to help with another game in return.

Wave goodbye to adventuring blues . . .

with new system, HelpLine automatic. No additives or

preservatives! And now of special sale prices — every solution must get it! You have a really naggy problem, troubles with computers in **Dracula**, or are unable to ring that bell in **Hunchback** — well **HelpLine (HL)** can help.

Yes, your super software **EXCEL** has devoted this simple system — even the **Homster** could use it!
Step 1 Take a pen
Step 2 Fill in the coupon
Step 3 Send it off to the address below
Follow these and before you can say **Ad**, your poses will have been solved. Please read the short print here . . .

British correspondents, please enclose a stamped, addressed envelope if you want a personal reply rather than wait some months for the magazine to come out. If you are writing from abroad, just enclose an envelope — I'll add the postage. I try to respond within two months but I can take longer (on the other hand you might receive an immediate reply). **ONLY DEAL WITH ADVENTURES** but arcade games, or technical problems (write to **Crusades** about those), or arcade adventures (20-page games included, not even **Heley** on the **Magick**). Please put the name of the game you're writing about on back of the envelope. And **PLEASE** don't write letters posing for general solutions — I just don't have the time to write them out for everyone. They can be answered, however, from **EXCEL** like **Adventure Probe** — see this month's introduction.
The address is: **HelpLine, 21 Computing, No 1 Golden Square, London W1R 3AS**. How many other magazines solve your problems like this **ARD**? Leave your hands silky smooth!

Title	_____
Company	_____
Problem	_____
I can help solve	_____
Name	_____
Address	_____



Apocalypse Gold

suffers. Korzhov, described as a "barren outlier" on the book cover, is not a card but a dog, in fact a cello case to hold his previous successes, a record of progress, setbacks and success. Quirk: Walter Wanderland looks you in the camera's eye — might as well — who's not considering a function-free button, which has never been in contact with the rest of mankind, yet developed of its own will.

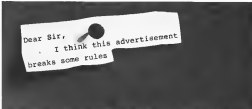
Immediately you notice the problem — address labels — considering the utility name — which are faster, better and much bigger than in real life.

games, and remain on screen rather than scroll up the page as you play. In Apocalypse Gold, but poor. Creative is. Walter Wanderland is neither game nor story consistently up to the design standard of the other four. Four years past the first window is too small, with no border, representing it from the graphics. Removes everything scrolls out of sight so soon you have to keep looking, but this covers the picture to show again on top of itself with a resulting unnecessary entry for display is block on white for an idea, down into Walter

Wanderland) with no re-designed characters. Combined with slightly sluggish playing time, the second two titles get into a bit of a rut. The thing like an idea on the strength, like (Buffed) Game Of Magic.

An advantage of SACED games over British ones is the rolling facilities you can enter commands in upper and lower case punctuation, and punctuation. Also there's games sometimes require information, with more than two key words. However simple, their technical superiority for actual ideas, but on this point by today's standards. Description in text is better, but still in Walter Wanderland and simple, almost childish in Apocalypse Gold. (Apocalypse Gold, but poor) Creative is. Walter Wanderland is neither game nor story consistently up to the design standard of the other four. Four years past the first window is too small, with no border, representing it from the graphics. Removes everything scrolls out of sight so soon you have to keep looking, but this covers the picture to show again on top of itself with a resulting unnecessary entry for display is block on white for an idea, down into Walter

In short, what we have here are two unremarkable adventures, which could have been written any year or two ago. They would make about a third of a good quality, interesting, budget game, however, right or not. It is a ridiculous price for such mediocrity so don't buy. What I find surprising is that these were the best of the many games available, were part of a collection. It is a pity that some of such games, especially those on the hardware of personal computers and Amiga, are not available. You can play elsewhere.



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 1, Brook House, Torrington Place, London WC1E 7HX

The space between individual words represents the standard of advertising.



RAINBIRD

Rainbird are probably the only software house never to get a bad review from ZX. So what's their secret?

When I went along to visit Rainbird Clive Edgway had just rolled up his desk to keep the safety officer happy — my desk had everything wrong with it. Desperately Clive's bit of floorpace looked like an oasis compared to the heap that passed for my own desk. In fact Rainbird's offices, in a modern office block in the West End of London, seemed a lot cleaner and more well organised than many of the software houses I've visited in the past. But then, whereas most UK software houses began to back bedroom outfits and slowly expanded into small companies as the piles of paperwork and machinery got out of hand, Rainbird came to the industry from the opposite direction. As part of the television group of companies (which also includes Festival, Beyond and Coln), Rainbird are actually a small offshoot of the vast British television organisation but like most software houses they've managed to avoid too much office clutter, and remain fairly informal.

Clive is Rainbird's PR Manager but she is also an ex-computer journalist having worked for Computer and Video Games and an obscure computer called Sinclair User. Former software reviewers moving into PR raise a potential question: Do their wholeheartedly push a product they would have given a less than stellar review in their former role? Fortunately that problem doesn't arise with Rainbird since I don't remember any of their products ever



getting a bad review. Unlike many companies that make similar claims, Rainbird have genuinely built up a reputation for releasing a small number of high quality products.

This is probably due to the fact that the label was set up to be the quality part of the Edgway group.

"We're the top end of the range — high quality", Clive told me. "We'll look at anything. Some of the best games have

landed on software houses' desktops as complete games, but if someone comes to us with a scenario or an idea that we like we'll develop it. But we wouldn't bring out just anything — it's got to be something that's not worth our £14000 price range we won't develop it."

But isn't it difficult to continually maintain a lead when you're dealing with a machine like the Spectrum which has been taken about as



for as it can go to town?

Yes, but converting those from big machines to big big products conversions for about ten machines and some of our games still on the Amiga and I and are converted down onto the Spectrum so we sometimes manage to do things that we didn't think could be done on the Spectrum. Standart has now changed speech which we didn't think could be able to do. I don't think there's a single bit of memory left in there.

An obvious example of this 'everything downwards' policy is Air Strike. The graphics were first brought to the Spectrum in terms of scans and menu-driven controls. Formerly only seen on such more powerful business machines. And you're not always a case of converting down onto the poor little Spectrum. Sometimes

collaborations with level 9 (the levels of Darkness and Silicon Dreams (Hoggs)) started on the Spectrum and have moved



The Knight Orc Cometh

The last two Level 9 products from Rainbird were upgraded versions of earlier releases, but coming next is Knight Orc, a brand new LP game in three parts, and something of a departure from this old style. Is it you play an Orc setting out to gain revenge on the humans who spend all their time victimising Orcs in fantasy novels and games (there's a twist in there though, but I'm not going to tell you about it). The game features a number of independent characters that you'll have to deal with, and whose help you'll need if you're going to complete the game so you can't just go around killing everything in sight.

After making the coffee, Clive



swings and spears through a whole range of machines and are doing very well in both America and Europe.

shuffled me into the computer room where I met Gary (also ex-CW) and Paul (our resident wit — he does naturally odd things). The Spectrum version of the game wasn't yet complete, but Paul (the wit resident) was, 'I'm the best wit' booted up the graphics for the Amiga version to show me.

Pete Selwyn, our resident adventure fiend, will be pleased to hear that LP have adopted a totally new style for their game graphics in Knight Orc. They've commissioned a number of pieces of original artwork and digitised them. The result, on the Amiga anyway, is pretty impressive, with some pictures that look almost impressionistic, and others that have almost photographic clarity of detail (fingers crossed their work look too bad on the Spectrum).

Knight Orc is due out for the Spectrum shortly, as is the 12th version of Magnus's Scroll's adventures, The Power. In the meantime though I decided that I should get back and tidy up my own desk in case the safety officer dropped by.

GAUNTLET



The arcade conversion set to make converts of us all.

Gauntlet
US Gold
£9.95

This is the game that everyone has been waiting for. During a wait for the official version of Gauntlet from US Gold, the rest of the software industry has developed an overwhelming conviction to produce Gauntlet imitations. The choice of "based on the clones" has brought us Stone (Atari), Dandy (Electric Dreams) and Druid (Preston).

So, has the reward been commensurate to the effort that Gauntlet fans and clones? The short answer is no. Despite approaching the game with expectations that were bound to disappoint, I can report that the

official Gauntlet is by far the most enjoyable and addictive of them all.

The original arcade game could be played by up to four players. On the Spectrum of course, this is whittled down to two (thankfully both can play with a joystick) but it is the option of cut and slash teamwork as you clear dungeons after dungeons of enemies that adds a whole new dimension to playing a Spectrum game. You would expect this two player option to slow down the action but astonishingly there is no loss of speed and the action is terrifically fast.





THOR De Warrior



TIFFIA De Valkyrie

You have a choice of four characters and, if playing the two player game, it's best to choose characters that have complementary characteristics. Thor, for instance, is a warrior with excellent muscle power and good in close combat but a dunce when it comes to casting spells. You might want to team him with Merlin who gets pummeled badly in hand to hand fighting, but can use his magic power to wipe out monsters and the generations that spawn them.

Then the Valkyrie's speciality is hand to hand combat and Tiffia the Elf relies on his magic charms to make up for moderate abilities of fighting.

As for adversaries, the dungeons are overflowing with them, ranging from ghosts to Drags, whose fatal charms will drain you of health, unless you have the spell to kill him. In addition there are Gynks, Demons, Lobbers and Squeakers all with their own characteristics and each requiring a slightly different approach to dispatch them most effectively.

It worried me that due to the complexity of Counter (99 levels) there is a lot of time spent re-winding and loading from tape. If you don't find Counter complex this could be a drag but if you become addicted these breaks can become a welcome respite from the un-rewarding battle.

Dungeon plays

The skill of the game of course is to fight Gorgs, Goblins, Gnomes and punch your way through those 999 plus levels. It's not a game for dedicated strategists. For one thing the systems of Dungeons decides threatening you leave little time for planning ahead. These are, however, plays you'll need to implement if you want to survive. Unless you are an advanced player you can forget about the points tally — worry instead about your health score. Go for all the food you can pick up (some of it is poisoned but after a while you can spot the difference) initially I found myself shooting food by mistake in the heat of the fray (not a good idea)

If cornered by monsters try to destroy the generations, you can't draw a screen completely until you have less staying of roomfuls of magic ghouls may make you feel good but think of your health which will be sapped by each individual contact.

Counter holds a lot of surprises, the graphics are vivid and maintain their clarity even on a screen teeming with detail. The speed of the game will satisfy even the hard-core arcade player, the lifespan of the game will keep the veteran conscious happy and there are enough elements to it to satisfy those who want something more than just shoot things. Of course if you just like shooting things and know somebody else who likes shooting things too, then this game will seem very close to heaven.



THE DISCOVERY COLUMN



Advice for Discovery Disc owners from John Wass

I suppose I've asked for it, but over the last few months quite a few enquiries have been made about the capacity of Discovery. Principally, I suppose, because I mentioned that it is possible to add a Double Sided Double Density (DDDD) 5.25" drive. So, to get things straight, let's look at just what there is inside that black metal box.

The Drive

As fitted to all standard Discovery units, this is a 40 track single sided three and a quarter inch drive known in the computer trade as a quarter megabyte drive. The tracks are recorded as concentric circles. The mechanism reads 40 tracks on just the one side of the disc. So one old three and a half inch disc which fits in a four and a half inch drive will fit the mechanism. It is possible to put tracks through closer together than the 40 track disc without there being interference between one track and the next; the mechanism is only able to record 40 tracks on it.

There is a power supply unit included on Discovery's circuit board which powers both Discovery and Spectrum. The power supply which came with early Spectrums had so little power to spare that there was not always enough to work Optima fully synchronous systems, so in some cases these would not work. The Discovery was a refinement of this system and its PSU now has plenty of power to spare.

The interface contains a read only memory (ROM) which indicates error messages and is paged in on receiving one of its

own commands. If you look at the Spectrum's memory map, you will see that the Spectrum ROM occupies the first 128K. Discovery ROM occupies the last 8K, and is interfaced in and out externally with the Spectrum ROM. As you know, the Spectrum ROM can be Paged to see what's in each address, but if you try to Paged it when into an address, it takes no notice. Similarly, the Discovery ROM contains uneditable information. The method of operation is exactly like interface 1, however data can operate instead of microdrives, and, as the code is written rather differently, the track codes which operate these machine code are different. So the two are compatible from any BASIC program, but not in respect of machine code.

Formatting

One of my correspondents recently was very upright because the capacity of a quarter megabyte drive is only 128K. This, he guessed, is grossly misleading. And so it is, but there is a good reason. Information recorded on a cassette tape is recorded sequentially, you have to grind through the lot until you come to what you want. A disc immediately accesses your chosen program. This is because each track on the disc is further divided up by the operating system into sectors or blocks, according to a set plan dictated by the software (the firmware on the Discovery ROM), the Directory or Catalogue of the disc. This directs the drive to read so many sectors, or alternatively, to move to so many sectors at particular locations. The number of tracks used is critical or a compromise. Imagine a disc divided up into only one sector, and imagine trying to save 12

programs on it. As you can't avoid the thing to start with, you'd have to load every program, which you could access instantly. However, it seems that you could index this program almost 256K long by a bit of fiddling about. Imagine dividing the thing up into one byte sectors. You would need a catalogue miles long for the thing to read and whilst economical up to a point, access times would be enormous. So the choice of sector length, etc. is a compromise between speed and waste. Discovery is pretty flexible, but reasonable access speeds and wastes relatively little of the disc.

Second disc

When a second disc is added, the drive mechanism pressed circuit board has provision to enable one to configure it as disc 2 and a Random Access Memory (RAM) chip has to be added inside Discovery. This is because the default values in the Discovery ROM all refer to disc 1, 40 track, single side. The RAM chip occupies the memory locations immediately above the Discovery ROM and is paged in with it on sensing the presence of the RAM chip. The ROM implies its default values into the RAM (i.e. RAM, unlike ROM can be altered) it can be Paged as well as Paged. If you read in a DSDD disc in drive 2, assuming that that is the mechanism which you have plugged in there, then the firmware senses that the drive is not 40 track and single sided, and the default values for drive 2 are adjusted in the RAM to match those of the disc read in. Before you can use such a system you must have a disc to read in or a program to configure it, hence the frustration

of some who have filled cars capacity driven but have only been able to get them to ROMM to MSB, the values are set at default and something with the system differently. Finally, if you have only the standard drive fitting the ROM chip is not really going to help you very much. If you will be able to adjust the cable length and one or two other things, but this will never enable you to store a megabyte or more of any more computer correspondents had helped I hope this will help to clear up some of the misconceptions.

I mentioned only a couple of months ago that I had been making which used Discovery's random access facility to Jim from several days, primarily set the DCLISTS. Although the program reads some data and writes on Jim would be the first to admit, it wasn't his idea the bit, and that is the object of the exercise. DCLISTS lists a list of all the programs and associated directory entries and so which are kept in random access file called "discs". Once you have this you can select one item by title, by the type of item or by the disc that it is on (the program is listed in Figure 1). It's pretty self explanatory although I think that I had to put in an extra line - 7 ONM. "P" to get it to print the files in lines 10 and 140 also need to be removed, they are there to remind you that certain parameters need some direction to enable much longer lists to be decompressed.

The Program

The first time that you ROM this program, you will have to do GO TO 29 when the program can find the random access data file "discs" used for in line 15, not suprising as you haven't made one yet. Lines 40-50 and 55 line up the list on the screen rather helpfully if I printed it out (Figure 2) so that you can see what the data file looks like. Each entry is given a number, on the left then followed the name of the entry. Next a letter indicating the type of item. Finally, a number for the disc on

```

10 REM *****
11 J.E. Burns 1984.
20 REM file random,header,info
30 LOAD header,info,msb,msb,msb
40 SET =00000000000000000000
50 GOTO info,header,msb,msb,msb
*****
60 PRINT "RANDOM ACCESS FACILITY"
70 REM *****
80 REM *****
90 PRINT "Header"
100 PRINT "Info"
110 PRINT "MSB"
120 PRINT "MSB"
130 PRINT "MSB"
140 PRINT "MSB"
150 PRINT "MSB"
160 PRINT "MSB"
170 PRINT "MSB"
180 PRINT "MSB"
190 PRINT "MSB"
200 PRINT "MSB"
210 PRINT "MSB"
220 PRINT "MSB"
230 PRINT "MSB"
240 PRINT "MSB"
250 PRINT "MSB"
260 PRINT "MSB"
270 PRINT "MSB"
280 PRINT "MSB"
290 PRINT "MSB"
300 PRINT "MSB"
310 PRINT "MSB"
320 PRINT "MSB"
330 PRINT "MSB"
340 PRINT "MSB"
350 PRINT "MSB"
360 PRINT "MSB"
370 PRINT "MSB"
380 PRINT "MSB"
390 PRINT "MSB"
400 PRINT "MSB"
410 PRINT "MSB"
420 PRINT "MSB"
430 PRINT "MSB"
440 PRINT "MSB"
450 PRINT "MSB"
460 PRINT "MSB"
470 PRINT "MSB"
480 PRINT "MSB"
490 PRINT "MSB"
500 PRINT "MSB"
510 PRINT "MSB"
520 PRINT "MSB"
530 PRINT "MSB"
540 PRINT "MSB"
550 PRINT "MSB"
560 PRINT "MSB"
570 PRINT "MSB"
580 PRINT "MSB"
590 PRINT "MSB"
600 PRINT "MSB"
610 PRINT "MSB"
620 PRINT "MSB"
630 PRINT "MSB"
640 PRINT "MSB"
650 PRINT "MSB"
660 PRINT "MSB"
670 PRINT "MSB"
680 PRINT "MSB"
690 PRINT "MSB"
700 PRINT "MSB"
710 PRINT "MSB"
720 PRINT "MSB"
730 PRINT "MSB"
740 PRINT "MSB"
750 PRINT "MSB"
760 PRINT "MSB"
770 PRINT "MSB"
780 PRINT "MSB"
790 PRINT "MSB"
800 PRINT "MSB"
810 PRINT "MSB"
820 PRINT "MSB"
830 PRINT "MSB"
840 PRINT "MSB"
850 PRINT "MSB"
860 PRINT "MSB"
870 PRINT "MSB"
880 PRINT "MSB"
890 PRINT "MSB"
900 PRINT "MSB"
910 PRINT "MSB"
920 PRINT "MSB"
930 PRINT "MSB"
940 PRINT "MSB"
950 PRINT "MSB"
960 PRINT "MSB"
970 PRINT "MSB"
980 PRINT "MSB"
990 PRINT "MSB"

```

Figure 1

1	=	MUSIC	H	1	9	=	DRABHITS	P	1	17	=	4,11	G	1
2	=	DARTS	A	1	10	=	pa	P	1	18	=	1aster	U	1
3	=	Traffic	B	1	11	=	<<DODD>>	A	1	19	=	T, O, H.	P	1
4	=	HEXODRW	U	1	12	=	uttl	U	1	20	=	disc0huts1	U	1
5	=	frust	B	1	13	=	dotqens	A	1	21	=	NIDM	H	1
6	=	jackpot	B	1	14	=	ratcatch	A	1	22	=	sake6	P	1
7	=	bouncer	A	1	15	=	life	p	1	23	=	cdswhop	A	1
8	=	T, Nat 40k	U	1	16	=	quendian	H	1	24	=	drawer	G	1

Figure 2



25 = TAB Fields U 1	67 = CHARACTER B 4	109= ARTStudio G 6
26 = ackack A 4	68 = estall G 4	110= super cat U 6
27 = headread U 1	69 = ORG P 4	111= test H 6
28 = attachdraw B 1	70 = clock H 4	112= mouse G 6
29 = Taword(r) H 2	71 = sort U 4	113= draw G 6
30 = patience G 3	72 = TRICK H 4	114= ship# B 6
31 = SHUNTING P 3	73 = noise gen H 4	115= card# B 6
32 = engine B 3	74 = roadrunner A 4	116= copyprint U 6
33 = edskey U 3	75 = 3d maze P 4	117= vroner2 G 6
34 = toolkit U 3	76 = cp H 4	118= disclaster F 6
35 = mysterbox P 3	77 = tick-tock G 4	119= DISSEMPH B 6
36 = BETA LISTER F 3	78 = EXPLORE B 4	120= anicator2 B 7
37 = BWDIT B 3	79 = EXERCISE H 4	121= tower# H 7
38 = organ H 3	80 = mendels G 4	122= TLM2 (last) H 7
39 = fade G 3	81 = Windows G 4	123= WRwcds2 F 7
40 = DATA (beta) F 3	82 = betachar H 4	124= DISCL187 F 7
41 = waves G 3	83 = continents H 4	125= Dscs4 F 7
42 = dice B 3	84 = ship# B 4	126= dscs1 F 7
43 = candle B 3	85 = BETA LISTER F 4	127= controls H 7
44 = Beta Basic H 3	86 = BETADISCLR F 4	128= WRwcds # 7
45 = BETADRAW H 3	87 = thick char U 4	129= write H 8
46 = BETADATA F 3	88 = futuristic U 4	130= ADDR F 8
47 = PUZZLE P 3	89 = factors H 4	131= wordmerge H 8
48 = brickyard A 3	90 = TV TEST G 4	132= DISCLIST F 8
49 = exam.exe U 3	91 = 3dHeight H 4	133= dscs F 8
50 = zoom B 3	92 = delater U 4	134= dscs2 F 8
51 = dntcalc F 3	93 = copyprint U 4	135= dscs3 F 7
52 = YES TITLE G 3	94 = ARCHER A 4	136= DISCLIST3 F 8
53 = CLR B 3	95 = super cat U 4	137= dscs3 F 8
54 = clearcorn B 3	96 = Golf B 3	138= rnuuu U 4
55 = flash G 3	97 = GARTS A 3	139= ZZZ
56 = explode G 3	98 = Accidents F 3	140=
57 = unglaf B 3	99 = kitchen A 3	141=
58 = SUND# H 3	100= EFFECT H 3	142=
59 = enlarger B 3	101= CHALLENGE G 3	143=
60 = disclaster F 3	102= DISSEM U 3	144=
61 = alaraclock U 3	103= PG-A10 U 3	145=
62 = wordsearch F 3	104= MDG edit U 3	146=
63 = letter sq F 3	105= arrows G 3	147=
64 = golf B 4	106= solitaire2 B 3	148=
65 = Beta Basic H 4	107= TABCALC F 6	149=
66 = TRUMAN B 4	108= TABDATA F 6	150=

which it'll find. Lines 66 to 68 direct the program to lines 608 (new entries), 603 (new additions), and 604 (new corrections). 645 (gets any number, what's the only?) 699 (pick entry by file, so that you can find which disc it's on), 618 (programs, etc., of any particular category), 619 (what is an any particular disc), 620 (SAMS routine), finally 96 (END).

In line 605 (in Jim's files — the way it's used) to have a miscellaneous (for) list. MORE in lines 695 and 626 are to look and insert cogs and, unfortunately Jim had used both capital and lower case "p" in line 606, and didn't test the putting of the entries in again.

Line 620 (above) contains a number of (PRINT) options for which line 7 is essential if you have a printer.

Inverse print

This program therefore could form a basis for your own custom access program. In this particular listing, Jim has made things more difficult for himself by inserting inverse printing directly into the listing — if that's good, but I don't really make it show up in the program, whereas INVERSE 1 always does (do please remember this when submitting programs). Nevertheless, the program does.

what it sets out to do. Perhaps a complete data base will follow. Who knows?

Errors

The program in last month's column for the resolution to GMS 3 has a bug in it. The following (program line will correct it) 644 # GOTO 646="DAT" THEN GLE, GAT 1; PRINT @:"Any Key Is Return"; PAGE# 6, LBT IN @D8PMB; 32470+356* PEEK; 32470+314*"Aug PEEK; 32470+256* PEEK; 32470+2276+657*A; PEEK@20+286* PEEK; 32470+4276: GO TO 66

CROWIRES

More mysteries from the Spectrum's technical
Twilight Zone with Ray Hides

Midi matters

Q Dear Sir,
As a new reader of your magazine I seek advice which has doubtless been answered before. Being the owner of a Spectrum +2 I have just obtained the linker from the COM2 & Spectrum 486. My main reason for purchasing the +2B was for the MIDI MIDI interface.

Having a MIDI equipped synthesizer I am interested in determining on the MIDI port in which pin is what and what is needed so that a connecting cable can be made. Different the +2B and the +2B+2 appears that there are two connectors used at the synthesizer MIDI port that I do not know what these are either.

On another note, can there any later facts for the +2B which would include the Commodore printer MP1002 under the title cover 1014 to be used?
King regards
A. Roberts, Llaner

A Last thing first by my know ledge there are no options which will allow the connection of Commodore equipment to the Spectrum. Commodore wind their own line and a manufacturer depends to their own standards and which will soon prove us all about everything else. I would strongly recommend buying the MIDI card direct as far as I'm concerned I must admit as to fully having satisfactory knowledge of it but prefer to buy these small units instead of spending hours staring over a hot soldering iron!

Commodore Marketing Ltd. 5 Willow Grove, Llaner, Llanelli, Carmarthenshire, SA31 3LW. I think you can get a 1024 bit DRAM for 19.95. Most logic branches of Buys also stock their wares.

The MIDI port is very limited and all you are likely to be able to do is send the PLAY signal out to your synth. To do anything more interesting such as record, arrange and send to equipment together you need a unit with at least 16 CPU specific MIDI outputs and a clock or sync device.

As you have a real synth I would strongly advise you to invest in a separate MIDI interface and software. These do a very impressive looking MIDI interface priced at £44.95.

Reverend, Research Ltd. of 44 Mount Glen, Wickford, Essex SS16 6ED. Tel: 0202 328193 produce the FS 1024B2 which is the best I have used. It will give you 16 outputs and cost around £150 which will confirm that their 1024B2C interface does the job deep time. I am not available. Unfortunately the two are not software compatible with each

other (I believe I know 808 have a real time package) but they have not yet been on a video review.

808 use of 10 Commodore Read System, Computing West Midlands, Tel: 021 351 2244 and their system costs around £100.

Mode muddles

Q Dear Sir,
I have a Spectrum 486 and for a project I need a floppy disk drive that will interface to Spectrum 486 +2 which reads continuously. There is however one problem and that is that the speed is increased with by a loud noise for which my supplier has no remedy.

I do not like the way in which there is the change in 486 reader on there is an 486 interface for the computer is in GMS LOCK or GMS/LOCK modes I was disappointed in the +2 and I gather that other readers have encountered difficulties I congratulate you then on the article 'Support Not Required' which I read with interest.
Looking to hear from you.
MR. Lupton, Middlesbrough

A I am glad there is nothing I can do about the lack of GMS LOCK or GMS/LOCK modes because that the same seems to be a mystery to all the engineers who you approach from a standard 486 Spectrum to the +2 +2 and I wonder if you modified the (software)?

Once produced an upgrade ROM to replace the existing 486 one in the computer, I tried it out and it worked without any problem, including any unexpected loud effects. If you have not tried the ROM upgrade then I suggest you contact Open and do as best you can.

The upgrade has been done or if the ROM was supplied specifically for the +2B then I suggest you try the Open with Quartz 100 and if that still makes then the fault is in the disc drive and it should be returned for checking and repair.

I think the noise/heat/loud sounds with speakers indicates that your computer has a fault and that should be checked and repaired.

Botronics runaround

Q Dear Sir,
A few months ago I purchased a 0485 printer from Botronics of High Wycombe, Bucks. At first it worked but the machine to work using the remote commands in the manual, but it worked really well when

I used the software supplied with it.

After then a drive unit has stopped working and I cannot fix it myself. I've tried to telephone Botronics but I cannot get a nice operator. I've also tried to contact Botronics with the 0485. Could you please let me if you could whether this company had ceased trading?

What could you tell me the address of Botronics Computer who is in a listed on the back of the printer. Please give your address.
DR. Pate, London W8R

A I'm afraid that Botronics are no longer trading having recently gone into liquidation. Your best bet is to contact one of the many repair companies of which Botronics and Video Vault are highly recommended.

Hard lines

Q Dear Sir,
I have recently bought an Omega 486 drive for my +2B Spectrum and whilst I am happy with its performance I have noticed that it causes a slight interference on the TV screen.

Do you think of your horizontal lines interfering and also when the screen goes through the interference is slight once noticed it can become noticeable.

Would be grateful if you could let pass the line drive and how to get rid of it. I should have the drive in the open in the TV?
Your faithfully
Philip A. Clark, Newcastle.

A You are right this is one of those things that can drive you mad! However there is not much you can do until I suggest that it is due to one of the components of your Spectrum (usually a small, but vital, IC) in the software used on the operational side of the specific line. Any slight change from the usual mode of operation could cause such a thing to occur.

Therefore you give up the idea, changing the position of you logging, not only of the Omega but also of most of the other components that make up your system in relation to each other. May you see the degree of the effect and perhaps cure it.

I would also suggest that you try changing the input impedance of the back of the Spectrum with a resistor (as opposed to bypass capacitor) or some "tune" circuit or using an device to gently remove any bit. You can then notice by experimenting!

THE BETTER LETTER

Modifications to the Tascword character set by John Wase and Colin Laycock

Both Tascword 2 and Tascword 3 utilize the same 94-character character set. Tascword software didn't alter it for Tascword 3, they merely changed the location. In both cases, therefore, Tascword's actual on-screen letters can be changed if one wishes. In this example many of the lower case letters have been redefined. Figure 1 to provide a character set which has much improved legibility when compared with the original Tascword letters.

Tascword 2

For Tascword 2, all you have to do is to enter BASIC after loading and do the POKE's listed in Figure 2. To use the new character set, select the "save Tascword" option from the main menu, and the code thus listed includes the new character set.

Tascword 3

Tascword 3 is only a little less straightforward. Subrout 33784 from each of the addresses given in Figure 2, this is conveniently done by: `LET J = 33784:POKE 4076-2, J:POKE ... etc.`

Vive La France

All this gives you a clue as to how the Tascword letters are stored. Moreover, if you consult in your Tascword manual (or possibly the one for your printer if its not an Epson) you will find that certain character codes corresponding to such things as curly brackets in the English character set correspond to accented letters in, say, the French or German set. It's then not too difficult to incorporate in Tascword 2 (or, in addition to line 40, to switch the printer to say, the French character set, similarly a line 40 or 70 can be readily added to Tascword 3. By appropriate POKE's to the addresses for the curly bracket in the Tascword character set, you can convert this into an accented letter if you'd a French Tascword both on screen and printer.

Tascword Two: Improved character set

POKE 33784-1, 7	POKE 33784, 7
POKE 33785, 7	POKE 33785, 7
POKE 33786, 7	POKE 33786, 7
POKE 33787, 7	POKE 33787, 7
POKE 33788, 7	POKE 33788, 7
POKE 33789, 7	POKE 33789, 7
POKE 33790, 7	POKE 33790, 7
POKE 33791, 7	POKE 33791, 7
POKE 33792, 7	POKE 33792, 7
POKE 33793, 7	POKE 33793, 7
POKE 33794, 7	POKE 33794, 7
POKE 33795, 7	POKE 33795, 7
POKE 33796, 7	POKE 33796, 7
POKE 33797, 7	POKE 33797, 7
POKE 33798, 7	POKE 33798, 7
POKE 33799, 7	POKE 33799, 7
POKE 33800, 7	POKE 33800, 7
POKE 33801, 7	POKE 33801, 7
POKE 33802, 7	POKE 33802, 7
POKE 33803, 7	POKE 33803, 7
POKE 33804, 7	POKE 33804, 7
POKE 33805, 7	POKE 33805, 7
POKE 33806, 7	POKE 33806, 7
POKE 33807, 7	POKE 33807, 7
POKE 33808, 7	POKE 33808, 7
POKE 33809, 7	POKE 33809, 7
POKE 33810, 7	POKE 33810, 7
POKE 33811, 7	POKE 33811, 7
POKE 33812, 7	POKE 33812, 7
POKE 33813, 7	POKE 33813, 7
POKE 33814, 7	POKE 33814, 7
POKE 33815, 7	POKE 33815, 7
POKE 33816, 7	POKE 33816, 7
POKE 33817, 7	POKE 33817, 7
POKE 33818, 7	POKE 33818, 7
POKE 33819, 7	POKE 33819, 7
POKE 33820, 7	POKE 33820, 7
POKE 33821, 7	POKE 33821, 7
POKE 33822, 7	POKE 33822, 7
POKE 33823, 7	POKE 33823, 7
POKE 33824, 7	POKE 33824, 7
POKE 33825, 7	POKE 33825, 7
POKE 33826, 7	POKE 33826, 7
POKE 33827, 7	POKE 33827, 7
POKE 33828, 7	POKE 33828, 7
POKE 33829, 7	POKE 33829, 7
POKE 33830, 7	POKE 33830, 7
POKE 33831, 7	POKE 33831, 7
POKE 33832, 7	POKE 33832, 7
POKE 33833, 7	POKE 33833, 7
POKE 33834, 7	POKE 33834, 7
POKE 33835, 7	POKE 33835, 7
POKE 33836, 7	POKE 33836, 7
POKE 33837, 7	POKE 33837, 7
POKE 33838, 7	POKE 33838, 7
POKE 33839, 7	POKE 33839, 7
POKE 33840, 7	POKE 33840, 7
POKE 33841, 7	POKE 33841, 7
POKE 33842, 7	POKE 33842, 7
POKE 33843, 7	POKE 33843, 7
POKE 33844, 7	POKE 33844, 7
POKE 33845, 7	POKE 33845, 7
POKE 33846, 7	POKE 33846, 7
POKE 33847, 7	POKE 33847, 7
POKE 33848, 7	POKE 33848, 7
POKE 33849, 7	POKE 33849, 7
POKE 33850, 7	POKE 33850, 7
POKE 33851, 7	POKE 33851, 7
POKE 33852, 7	POKE 33852, 7
POKE 33853, 7	POKE 33853, 7
POKE 33854, 7	POKE 33854, 7
POKE 33855, 7	POKE 33855, 7
POKE 33856, 7	POKE 33856, 7
POKE 33857, 7	POKE 33857, 7
POKE 33858, 7	POKE 33858, 7
POKE 33859, 7	POKE 33859, 7
POKE 33860, 7	POKE 33860, 7
POKE 33861, 7	POKE 33861, 7
POKE 33862, 7	POKE 33862, 7
POKE 33863, 7	POKE 33863, 7
POKE 33864, 7	POKE 33864, 7
POKE 33865, 7	POKE 33865, 7
POKE 33866, 7	POKE 33866, 7
POKE 33867, 7	POKE 33867, 7
POKE 33868, 7	POKE 33868, 7
POKE 33869, 7	POKE 33869, 7
POKE 33870, 7	POKE 33870, 7
POKE 33871, 7	POKE 33871, 7
POKE 33872, 7	POKE 33872, 7
POKE 33873, 7	POKE 33873, 7
POKE 33874, 7	POKE 33874, 7
POKE 33875, 7	POKE 33875, 7
POKE 33876, 7	POKE 33876, 7
POKE 33877, 7	POKE 33877, 7
POKE 33878, 7	POKE 33878, 7
POKE 33879, 7	POKE 33879, 7
POKE 33880, 7	POKE 33880, 7
POKE 33881, 7	POKE 33881, 7
POKE 33882, 7	POKE 33882, 7
POKE 33883, 7	POKE 33883, 7
POKE 33884, 7	POKE 33884, 7
POKE 33885, 7	POKE 33885, 7
POKE 33886, 7	POKE 33886, 7
POKE 33887, 7	POKE 33887, 7
POKE 33888, 7	POKE 33888, 7
POKE 33889, 7	POKE 33889, 7
POKE 33890, 7	POKE 33890, 7
POKE 33891, 7	POKE 33891, 7
POKE 33892, 7	POKE 33892, 7
POKE 33893, 7	POKE 33893, 7
POKE 33894, 7	POKE 33894, 7
POKE 33895, 7	POKE 33895, 7
POKE 33896, 7	POKE 33896, 7
POKE 33897, 7	POKE 33897, 7
POKE 33898, 7	POKE 33898, 7
POKE 33899, 7	POKE 33899, 7
POKE 33900, 7	POKE 33900, 7
POKE 33901, 7	POKE 33901, 7
POKE 33902, 7	POKE 33902, 7
POKE 33903, 7	POKE 33903, 7
POKE 33904, 7	POKE 33904, 7
POKE 33905, 7	POKE 33905, 7
POKE 33906, 7	POKE 33906, 7
POKE 33907, 7	POKE 33907, 7
POKE 33908, 7	POKE 33908, 7
POKE 33909, 7	POKE 33909, 7
POKE 33910, 7	POKE 33910, 7
POKE 33911, 7	POKE 33911, 7
POKE 33912, 7	POKE 33912, 7
POKE 33913, 7	POKE 33913, 7
POKE 33914, 7	POKE 33914, 7
POKE 33915, 7	POKE 33915, 7
POKE 33916, 7	POKE 33916, 7
POKE 33917, 7	POKE 33917, 7
POKE 33918, 7	POKE 33918, 7
POKE 33919, 7	POKE 33919, 7
POKE 33920, 7	POKE 33920, 7
POKE 33921, 7	POKE 33921, 7
POKE 33922, 7	POKE 33922, 7
POKE 33923, 7	POKE 33923, 7
POKE 33924, 7	POKE 33924, 7
POKE 33925, 7	POKE 33925, 7
POKE 33926, 7	POKE 33926, 7
POKE 33927, 7	POKE 33927, 7
POKE 33928, 7	POKE 33928, 7
POKE 33929, 7	POKE 33929, 7
POKE 33930, 7	POKE 33930, 7
POKE 33931, 7	POKE 33931, 7
POKE 33932, 7	POKE 33932, 7
POKE 33933, 7	POKE 33933, 7
POKE 33934, 7	POKE 33934, 7
POKE 33935, 7	POKE 33935, 7
POKE 33936, 7	POKE 33936, 7
POKE 33937, 7	POKE 33937, 7
POKE 33938, 7	POKE 33938, 7
POKE 33939, 7	POKE 33939, 7
POKE 33940, 7	POKE 33940, 7
POKE 33941, 7	POKE 33941, 7
POKE 33942, 7	POKE 33942, 7
POKE 33943, 7	POKE 33943, 7
POKE 33944, 7	POKE 33944, 7
POKE 33945, 7	POKE 33945, 7
POKE 33946, 7	POKE 33946, 7
POKE 33947, 7	POKE 33947, 7
POKE 33948, 7	POKE 33948, 7
POKE 33949, 7	POKE 33949, 7
POKE 33950, 7	POKE 33950, 7
POKE 33951, 7	POKE 33951, 7
POKE 33952, 7	POKE 33952, 7
POKE 33953, 7	POKE 33953, 7
POKE 33954, 7	POKE 33954, 7
POKE 33955, 7	POKE 33955, 7
POKE 33956, 7	POKE 33956, 7
POKE 33957, 7	POKE 33957, 7
POKE 33958, 7	POKE 33958, 7
POKE 33959, 7	POKE 33959, 7
POKE 33960, 7	POKE 33960, 7
POKE 33961, 7	POKE 33961, 7
POKE 33962, 7	POKE 33962, 7
POKE 33963, 7	POKE 33963, 7
POKE 33964, 7	POKE 33964, 7
POKE 33965, 7	POKE 33965, 7
POKE 33966, 7	POKE 33966, 7
POKE 33967, 7	POKE 33967, 7
POKE 33968, 7	POKE 33968, 7
POKE 33969, 7	POKE 33969, 7
POKE 33970, 7	POKE 33970, 7
POKE 33971, 7	POKE 33971, 7
POKE 33972, 7	POKE 33972, 7
POKE 33973, 7	POKE 33973, 7
POKE 33974, 7	POKE 33974, 7
POKE 33975, 7	POKE 33975, 7
POKE 33976, 7	POKE 33976, 7
POKE 33977, 7	POKE 33977, 7
POKE 33978, 7	POKE 33978, 7
POKE 33979, 7	POKE 33979, 7
POKE 33980, 7	POKE 33980, 7
POKE 33981, 7	POKE 33981, 7
POKE 33982, 7	POKE 33982, 7
POKE 33983, 7	POKE 33983, 7
POKE 33984, 7	POKE 33984, 7
POKE 33985, 7	POKE 33985, 7
POKE 33986, 7	POKE 33986, 7
POKE 33987, 7	POKE 33987, 7
POKE 33988, 7	POKE 33988, 7
POKE 33989, 7	POKE 33989, 7
POKE 33990, 7	POKE 33990, 7
POKE 33991, 7	POKE 33991, 7
POKE 33992, 7	POKE 33992, 7
POKE 33993, 7	POKE 33993, 7
POKE 33994, 7	POKE 33994, 7
POKE 33995, 7	POKE 33995, 7
POKE 33996, 7	POKE 33996, 7
POKE 33997, 7	POKE 33997, 7
POKE 33998, 7	POKE 33998, 7
POKE 33999, 7	POKE 33999, 7
POKE 34000, 7	POKE 34000, 7
POKE 34001, 7	POKE 34001, 7
POKE 34002, 7	POKE 34002, 7
POKE 34003, 7	POKE 34003, 7
POKE 34004, 7	POKE 34004, 7
POKE 34005, 7	POKE 34005, 7
POKE 34006, 7	POKE 34006, 7
POKE 34007, 7	POKE 34007, 7
POKE 34008, 7	POKE 34008, 7
POKE 34009, 7	POKE 34009, 7
POKE 34010, 7	POKE 34010, 7
POKE 34011, 7	POKE 34011, 7
POKE 34012, 7	POKE 34012, 7
POKE 34013, 7	POKE 34013, 7
POKE 34014, 7	POKE 34014, 7
POKE 34015, 7	POKE 34015, 7
POKE 34016, 7	POKE 34016, 7
POKE 34017, 7	POKE 34017, 7
POKE 34018, 7	POKE 34018, 7
POKE 34019, 7	POKE 34019, 7
POKE 34020, 7	POKE 34020, 7
POKE 34021, 7	POKE 34021, 7
POKE 34022, 7	POKE 34022, 7
POKE 34023, 7	POKE 34023, 7
POKE 34024, 7	POKE 34024, 7
POKE 34025, 7	POKE 34025, 7
POKE 34026, 7	POKE 34026, 7
POKE 34027, 7	POKE 34027, 7
POKE 34028, 7	POKE 34028, 7
POKE 34029, 7	POKE 34029, 7
POKE 34030, 7	POKE 34030, 7
POKE 34031, 7	POKE 34031, 7
POKE 34032, 7	POKE 34032, 7
POKE 34033, 7	POKE 34033, 7
POKE 34034, 7	POKE 34034, 7
POKE 34035, 7	POKE 34035, 7
POKE 34036, 7	POKE 34036, 7
POKE 34037, 7	POKE 34037, 7
POKE 34038, 7	POKE 34038, 7
POKE 34039, 7	POKE 34039, 7
POKE 34040, 7	POKE 34040, 7
POKE 34041, 7	POKE 34041, 7
POKE 34042, 7	POKE 34042, 7
POKE 34043, 7	POKE 34043, 7
POKE 34044, 7	POKE 34044, 7
POKE 34045, 7	POKE 34045, 7
POKE 34046, 7	POKE 34046, 7
POKE 34047, 7	POKE 34047, 7
POKE 34048, 7	POKE 34048, 7
POKE 34049, 7	POKE 34049, 7
POKE 34050, 7	POKE 34050, 7
POKE 34051, 7	POKE 34051, 7
POKE 34052, 7	POKE 34052, 7
POKE 34053, 7	POKE 34053, 7
POKE 34054, 7	POKE 34054, 7
POKE 34055, 7	POKE 34055, 7
POKE 34056, 7	POKE 34056, 7
POKE 34057, 7	POKE 34057, 7
POKE 34058, 7	POKE 34058, 7
POKE 34059, 7	POKE 34059, 7
POKE 34060, 7	POKE 34060, 7
POKE 34061, 7	POKE 34061, 7
POKE 34062, 7	POKE 34062, 7
POKE 34063, 7	POKE 34063, 7
POKE 34064, 7	POKE 34064, 7
POKE 34065, 7	POKE 34065, 7
POKE 34066, 7	POKE 34066, 7
POKE 34067, 7	POKE 34067, 7
POKE 34068, 7	POKE 34068, 7
POKE 34069, 7	POKE 34069, 7
POKE 34070, 7	POKE 34070, 7
POKE 34071, 7	POKE 34071, 7
POKE 34072, 7	POKE 34072, 7
POKE 34073, 7	POKE 34073, 7
POKE 34074, 7	POKE 34074, 7
POKE 34075, 7	POKE 34075, 7
POKE 34076, 7	POKE 34076, 7
POKE 34077, 7	POKE 34077, 7
POKE 34078, 7	POKE 34078, 7
POKE 34079, 7	POKE 34079, 7
POKE 34080, 7	POKE 34080, 7
POKE 34081, 7	POKE 34081, 7
POKE 34082, 7	POKE 34082, 7
POKE 34083, 7	POKE 34083, 7
POKE 34084, 7	POKE 34084, 7
POKE 34085, 7	POKE 34085, 7
POKE 34086, 7	POKE 34086, 7
POKE 34087, 7	POKE 34087, 7
POKE 34088, 7	POKE 34088, 7
POKE 34089, 7	POKE 34089, 7
POKE 34090, 7	POKE 34090, 7
POKE 34091, 7	POKE 34091, 7
POKE 34092, 7	POKE 34092, 7
POKE 34093, 7	POKE 34093, 7
POKE 34094, 7	POKE 34094, 7

Sue Townsend THE GROWING PAINS OF ADRIAN MOLE

PROGRAM BY LEVEL II



This beautiful video, the complete version of "The Growing Pains of Adrian Mole" hosted by Sue Townsend, is both a comedy and a satire. It tells the funny story of a boy being bullied at school in the 1950s. It is not based quite with the usual distinctions, and there are no real lessons to be learned possible through your awareness of multiple cultural questions. The four programs covering 12 episodes of the original television series will be available in a different computer

TM & © 1992 by Mosaic Publishing Ltd.

Produced by Mosaic Publishing Ltd.

The Growing Pains of Adrian Mole is available from a range of software distributors. If you have trouble finding you can buy direct from Mosaic Games. Please contact the Mosaic Games department at the address below or to: Mosaic Games Mail Order, 21 St. James Road, Portland Road, Luton, Bedfordshire LU1 3 2EX. Make cheques or credit cards payable to: Mosaic Games Ltd. Please allow 4 weeks for delivery.

COMPUTER FORMAT	PRICE	AVAILABLE DATE
Amstrad 486/586	£14.95	18 February 1992
Commodore 64/68K	£11.95	18 February 1992
Amiga 500/1000	£11.95	18 February 1992
MS-DOS	£11.95	12 February 1992
MS-DOS 3.86	£11.95	12 February 1992
MS-DOS 3.86/4.0 CD-ROM	£11.95	12 February 1992
Apple II/IIx/IIc/IIc+/IIcX/IIcX+ Plus 2/Plus 2X/IIcX	£14.95	12 February 1992

Name

Address

Money Enclosed



ATARI ST
AMSTRAD

£14-95
£8-95

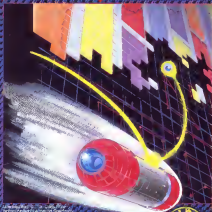


TAITO
COIN-OP

COMMODORE £8-95
SPECTRUM £7-95

TAITO
COIN-OP

WORLD OF GOLD



World of Gold is a golf game that's different from any other. It's a game that's fun to play and easy to learn.

World of Gold
The name of the game



Golden Tee Golf is a trademark of Taito Corporation.
© 1995 Taito Corporation. All rights reserved.

World of Gold is a trademark of Taito Corporation.