

A Database Publication

# ATARI USER

Vol. 2 No. 4

August 1986

£1

*Inside  
this  
issue*



**Use your  
micro to  
control  
the mains**



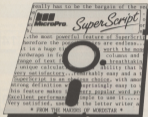
**80 column text on your Atari**  
**Get-It-Right! checksum updated**

**Arcade action: Frank the Fruit Fiend**





# WORDAHOLIC



## SUPERSCRIPT WORDPROCESSING

From the makers  
of WORDSTAR

Most wordprocessing packages will do the bare minimum. SuperScript can't wait to do the lot!

### JUST LOOK AT THIS WORKLOAD

<b>Takes on</b>	writing, storing and editing all your correspondence and other documents.
<b>And</b>	prepares letters, invoices, price lists, contracts, labels etc.
<b>Next</b>	runs personalized mailings with lists or database output.
<b>Then</b>	creates and revises top-quality presentations reports and tables.
<b>On to</b>	typing term papers, technical reports or notes.
<b>And even</b>	adds, subtracts, multiplies, divides and does percentages.
<b>At the same time</b>	giving all round editing and printing facilities such as block-out-paste, insert and typoset, full search and replace.
<b>Not to mention</b>	proof review, layout options and simplified document selection. And much, much more!

### \* SPECIAL MAIL ORDER PRICES \*

COMMODORE 64, ATARI 800/1300E £49.95  
COMMODORE 128, APPLE IIe/IIx £79.95  
UPGRADE FROM EASYSCRIPT £39.95



SuperScript comes complete with Comprehensive User Manual.

## ORDER NOW

By calling MicroPro  
on 01-876 1122 Ext 310  
quote your Accession No. 10

also available from  
all good dealers.



### UPGRADE FROM EASYSCRIPT TO SUPERSCRIPT — JUST £39.95

- \* Improved performance \*
- \* Maths \*
- \* Spelling checker \*
- \* Print or view \*
- \* Integrate with Superbase \*

See Accessories 10/85

NAME

ADDRESS

CITY

POSTCODE

TELEPHONE

ORDER NO.

DATE

NAME

ADDRESS

CITY

POSTCODE

TELEPHONE

ORDER NO.

DATE

NAME

ADDRESS

CITY

POSTCODE

TELEPHONE

ORDER NO.

DATE



Vol. 2 No. 4 August 1988

**Managing Editor:** Derek Macklin  
**Features/Technical Editor:** Andrew Willey  
**Production Editor:** Peter Glover  
**Art Editor:** Heather Sheehy  
**Reviews Editor:** Christopher Payne  
**News Editor:** Mike Cowley  
**Advertisement Manager:** John Biding  
**Advertising Sales:** John Snowden  
**Editor in Chief:** Peter Bransford

**Editorial:** 001-488 8888  
**Administration:** 001-488 8383  
**Advertising:** 001-488 8888  
**Subscriptions:** 001-488 0173  
**Telecom Gold:** 72 8668001  
**Telex:** 268875 MONREF G  
**Cable:** GARDING RL 72 8668001  
**Postal Mailing:** 814888308

**Published by:**  
Database Publications Ltd,  
Europe House, 88 Chester Road,  
Hazel Grove, Stockport SK7 5RY.

**Subscription rates for 12 issues, post free:**  
£12 - UK  
£16 - Europe  
£30 - Overseas (Normal)

"Atari User" welcomes program listings and articles for publication. Material should be typed on computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication by Database Publications Ltd will be on an all-rights basis.

© 1988 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles, listings or advertisements.

"Atari User" and "Atari ST User" are independent publications and Atari Corp (USA) is not responsible for any of the articles they contain or for any of the opinions expressed.

News trade distribution:  
Europeas Sales and Distribution (London), Unit 1, Burgess Road, Inchbush Lane, Hastings, East Sussex TN26 4NR.  
Tel: 0424 450422.

## News

We keep you completely up to date with this month's happenings and new releases around the UK in the Atari 8 bit and ST world.

9

## Software

There's a lot of Belly Hop down at Beer Belly Bum's Brew Biz, as Nuclear Nisk takes a Solo Flight down to Beach Head II... We review them all.

14

## Graphics

Steven Williamson demonstrates how machine code can help to move your Player Missile Graphics up and down the screen.

18

## Gadgets

In the third part of his series on Great Little Gadgets, Len Golding starts looking at the problems involved in controlling mouse apparatus from your computer.

22



## Adventuring

Brillig emerges once again from the dark depths of the computer dungeon to answer more of your adventures questions.

25

## Game

Help Frank the Fruit Fly eat his fill in this eight level Basic and machine code arcade game by G. Bellough.

27

## Checksum

Bowing to tremendous demand, we re-print our popular Get-it-Right! checksum program, with the bonus of a few added features.

33

## Mailbag

You've sent in another varied batch of letters this month, with questions and comments on all aspects of Atari computers.

35

# £25 for you!

## Five Liners

The first Five-Line programs from our readers help to liven up your text display, turn your Atari into a typewriter, and keep track of your variables. Each one wins £25.

40

## 80 Column

An amazing utility to generate 80 column text displays by adding a special character set to a Graphics 8 screen.

42



## 3 ST Roundup

Mike Cowley takes another look at the latest developments on the ST scene - including the promise of a Unix-based system later this year.

## 5 Advice

Andrew Bennett answers more of your questions, and clears up some of the confusion surrounding Atari's numbering system.

## 10 Graphics

Peter Fellows has been looking into one way of using the ST's astounding graphics to help small businesses.

## 14 Reviews

We look at Essex, a text adventure, and Major Motion, a fast action arcade game, in this month's software reviews for the ST.

## 19 Hardware

Gordon Taylor takes a look at the new A S & T one meg disc drive and system stand.

## 20 Accounts

Peter Connors evaluates Cashlink Accounts, a new and powerful accounts package for the ST.

## 25 Database

H & D Base is a remarkable new software package for the ST which offers the same features as Dbase II. Simon Terry investigates.

## 27 American Scene

Our US correspondent reports on last month's Consumer Electronics Show in Chicago, and all the latest state-side releases.

# Compu



This Month's Megad

# Atari 13

130XE · XC12 · ATARI

MINER 2049'er · Eastern Front  
 Invitation to Programming  
 Pole Position · Lone Raider  
 XE Handbook

(RRP £232.14)

# £165



## Blank Disks

Quantity of Blank Disks

10 Single Sided Double Density

£9.99 ~~12.99~~

10 Double Sided

£13.99 ~~16.99~~

50 Single Sided Double Density

£49.99 ~~59.99~~

50 Double Sided Double Density

£69.99 ~~79.99~~



Free with any order + 2 FREE Disks

Free with any order + 10 FREE Disks

Free with any order + 100 FREE Disks

Look WE'VE MO

24 Hour

VISA

Credit Card  
 Order Line

0509-233893

Goods despatched same d

Free

CO

Unit 8, 1

It's Our  
**Birthday**



Special

**BOXE  
PACK**  
**JOYSTICK**

Front  
ing-1  
cler

.00

£3 p+p

**Atari Hardware**

Atari 130XE-128K Computer  
FREE Minor 204P'er

**£94.00**  
£3p+p

Atari 1050-Disk Drive - 10-  
FREE Steve Davis Snooker. FREE Blank Disks  
Pack-RFM-Payoff-Dos 2.5-Demo's

**£127.00**  
£3p+p

Atari 1029-Dot Matrix Printer  
FREE Atariwriter+ 8 Graphics Dump

**£129.00**  
£3p+p

Atari 1027-Letter Quality Printer

**£125.00**  
£3p+p

We Stock a Wide Range of Atari Peripherals  
CALL NOW for Incredibly LOW Prices

SAVE  
AVE-S

**£35**

**£26**

**£40**

**£25**

**MOVED**



**To NEW... Larger Premises!!!**

day . . . Disks 1st class post, Hardware - parcel post

Please send cheques/postal orders to:

**COMPUMART (Dept AU)**

6, Falcon Street, Loughborough,  
Leicester, LE11 1EH



Computer!

**0509-262259**

**HOTLINE**

# Compumart

**SAVE**  
**£140**

**ST Megadeal**

**ATARI 520 STM** **PACK**

Only ~~£449.00~~ **£349.00** 3p&p inc.

**520 STM - 1/2 Meg. Drive**  
128K Rom-TOS/GEM on Rom-Mouse-S/ware  
Plugs into your TV. via the onboard RF Modulator

**CUMANA**

**3 1/2" - 1 Megabyte Disk Drive**

**£155.00** £3p-p

**20-3 1/2" SS DISKS**  
**WORTH £40**

**FREE**  
**WITH EVERY**  
**STM PACK**



**Top Quality**  
**3 1/2" Microdisks**

**BOXED IN 10's - LABELS etc....**  
OFFER WARRANTY

**SINGLE SIDED**  
**135 T.R.I.**

**£19.00** + £1.50p

**DOUBLE SIDED**  
**135 T.R.I.**

**£24.00** + £1.50p

**3 1/2" Microdisks**

**Call Now....**  
**For BEST**  
**Prices on the**  
**FULL ST Range**

**Now... Software Too!!**

**WE'VE MOVED**  
To NEW... Larger Premises!!!

Goods despatched same day. .... . Orders 1st class post, Hardware - parcel post

Please send cheque/credit orders to:

**COMPUMART (Dept 888)**  
Unit 8, Falcon Street, Loughborough,  
Leicester, LE11 1EM



**Compumart**  
**0509-262259**  
MORNING

**24Hour**  
**184**  
**Credit Card**  
**Order Line**  
**09-230000**

# Atari 8 bit market is booming

THE Atari 8 bit market is booming as never before, keeping computer shop cash registers jingling up and down the country at a time when business is normally slack.

Defying the traditional summer sales lull, turnover in both hardware and software sectors has been brisk during the past couple of months, an Atari Observer survey has revealed.

Dealers say there are two main reasons why Atari 8 bit machines and software are moving when other brands are not.

One is the spectacular sale of 100,000-80000 machines through Dixons over Christmas which boosted the Atari 8 bit user base to a massive 300,000 in the UK.

The other is that software houses are responding to this - and the ongoing success of the 130XE - by producing a steady supply of new programs at affordable prices.

Where the Atari 8 bit owner could expect to pay about £18.95 - and sometimes as much as £29.99 - per game a year ago, he can now get quality titles for £9.99, with some as

cheap as £2.99.

Retailers report that lower prices have dramatically increased unit sales of Atari 8 bit software, making it possible for youngsters to buy games with their pocket money instead of having to save up for weeks.

Although some software publishers were initially unhappy at lower profit margins per title, most say this has been more than compensated for by vastly increased sales.

Marketing director of Atari main dealership Silvio Shop Tony Deane summed up the current situation: "Owners of Atari 8 bit machines need have no fear for the future.

"The entire trade is very pleased with the volume of recent business, coming at a time when the market is normally slow. In fact we have had a couple of really good months.

"A key factor has been that the Atari 8 bit software scene has changed completely, with a

vast number of low-priced titles becoming available in response to the massive user base.

"What we are seeing is something similar to the way the Spectrum market went, with publishers starting to produce software at sensible prices.

"This has created a brand new buyer market, particularly among children, which is to everyone's benefit. Kids are now able to buy a cheap game each week during their summer holidays where before this was beyond their means.

"The new software situation is exactly what the public and the dealers want - inexpensive titles and vastly increased unit sales.

"The huge success of the Dixons deal at Christmas has ensured continuing support from software houses for the Atari 8 bit machines.

"In fact this support is actually increasing momentum as more publishers climb on the bandwagon".

## Latest releases

APL08000 for the Atari ST has been released by MicroAPL.

Widely used on main and mainframe systems, APL is now entering the micro market in a diverse set of applications, including financial modelling, statistics, market research and insurance.

APL functions can exchange data with other Atari applications. As well as the standard APL08000 product, which uses the special APL character set, MicroAPL also has a version which uses plain English keywords to replace the special symbols.

It costs £170.

\*\*\*

CONVERSIONS of four adventure games for Atari 8100 users have been released by Pellico Software, Rick Hanson, Project Theius, Island of Acan and Embar Seven will only be available on disc.

Embar Seven costs £17.99 with the other three at £14.99.

\*\*\*

FIVE new adventures - Nightmare Maze, Sorcerbal, Garia, Diamond Mine, and Castle Assault - have been released by Blue Rabbit for the Atari 800 range.

All cassette versions will cost £1.99 with a games disc costing all five at £9.99.

\*\*\*

A HIT to stay overnight in a featured house format, the Ghosts of Clock of Death, a new adventure for the 8000, and 8000L from Argus Press Software.

Once in the house the doors slam shut and here comes the Battlemoon from Vragus IV! The cassette costs £2.99.

\*\*\*

A TEST of your knowledge of Britain comes in a new flying adventure for users of the 800800 XLS and 130XE.

Released by Aristocrat, Map Britain has you as the pilot of a helicopter seeking locations like towns, cities, rivers and counties. Prints are scored when the right place is found in a certain time.

Price: £6.99.

## WHAT THE STARS SAY

A NEW program for the Atari 8 bit range, Paranormal from Riverside, is a four part destination horoscope.

Company director David Edwards said: "A lot of research has gone into this. It not only predicts your future but indicates your present and past."

After feeding in your time and date of birth a full prediction is produced. In ESP a check can be made on your psychic abilities and Reincarnation - fortunate - reveals if you have lived before and as what.

Numerology assesses your character and Bio-Rhythms records your mental, physical and emotional powers.

Price: £4.

## Bundled STs snapped up

BUNDLED Atari ST machines are selling like hot cakes at a North London retail outlet. An average of 30 520STM and 1040STT packages a day are being snapped up by customers at Twitler Computers in Southall. There are six packages to choose from, four including the 520ST and two the 1040ST. The 1040ST package aimed at small business users is slightly outselling the 520ST one, bought mainly by home users, reports Twitler managing director Anagh Ghazi.

Prices range from £375 for a 520ST bundled with 800k disc drive and mono monitor to £950 for a 1040ST with built-in 1mb drive and colour screen.

"We find the bundling of ST machines has been a tremendous success", says Ghazi told Atari Observer. "Customers usually know what they want the machine to do and how much they want to spend, but they are often confused about which add-ons to buy".









Steven Burke...celebrating

## Birthday offer

TO celebrate its first birthday Computer, the Atari mail order company, is offering an Atari ST pack for £445 — a saving of £140.

The pack consists of the 52001M with 500K drive and 25 3.5in disks.

Company managing director Steven Burke said: "We have just completed a move to a new warehouse in Loughborough and as it is also our birthday we decided to make special offers to celebrate".

For readers of Atari User the company is offering two free disks with every box of 10 free.

## TEENAGERS SUE OVER ATARI GAME

A MILLION dollar suit has been filed against the Atari Games Corporation by two teenagers from Redwood City, California.

The boys are claiming that the bestselling arcade game Paperboy released by the Warner Communications-controlled company had been written by them.

Mark Casar, 14, and Robin Hallingford, 18, say they sent an outline of the game to Atari back in 1983.

So they were somewhat surprised two years later when they came across the game in a

A NEW wave of games products is about to be launched for the ST market in the UK on the basis of sales of the machines over-topped unabated.

The latest breakthrough has been made by Peter Connors, a regular contributor to Atari User, with a program called Flying Start which he is selling for £10.

"I bought an ST to take the place of my aging BBC Micro, but I still wanted to be able to access services like Protext", said Connors. "I don't set out to write a commercial package, just something that I could use myself".

But now there is a good chance Flying Start could be taken up by one of the major computer software publishers, and it has already been

enthusiastically received by Microsoft.

STAIR has entered the modern era in the United States with a 128K board full duplex Hayes-compatible device priced under \$300.

It will serve Atari's entire range of ST and 8 bit machines and is expected to appear in the shops within weeks.

But the machine is unlikely to be seen in the UK due to the long drawn out process of obtaining specifications approval by BASF.

enthusiastically received by Microsoft.

The package will work with any V21/V22 modem and has full colour or monochrome text and graphics. It can handle scrolling bulletin boards and is fully menu driven.

Connors is currently working on enhancements that will lead to a V.1-1.03 version to follow the existing V-1.01 product. It is "somewhat higher priced".

Meanwhile modern manufacturer Minolta Technology is putting the final touches to a powerful integrated ST cinema software package called SuperTern which it intends to launch on the Atari stand at the PCW Show in September.

It will have standard Atari and Viewdata/Protext modes of operation based on the GEM environment with VTI80 emulation. Users can select from low or high resolution displays.

SuperTern will have capability for telesoftware downloading, file transfer, print frames to printer, and carry an in-built telephone directory.

Exact now will be selectable

between 300 full duplex up to 1400 full duplex and the complete package is expected to cost just under £30.

Bundled with Minolta's new Hayes-compatible WS4000 modem, it will be offered as a complete communications for the ST at under £200.

## Trivial Pursuit for Atari

TRIVIAL Pursuit, currently the world's most successful board game, is to be released by Comark for Atari users early next year.

With 10 million copies having been sold worldwide — three million in the UK — there was fierce competition among software companies to obtain the rights to produce a home computer version of the game.

Domark's joint managing director, Mark Strachan, said: "We're delighted that the Tevie people chose us".

The game will be very similar to the board version with the computer displaying the board and running the game.

A minimum of 3,000 questions will be included with the program, some with text only, but nearly half with sound and graphics.

To make replacement questions easier to market Domark is using a new system, UniLoad, which will mean that other versions can use the same data file.

Strachan said: "We have already had 40,000 advance orders".

It is expected to sell for about £14.95.



# ATARI Users...

## High Quality Blank Discs at low low prices

Micro Resources (UK), Superior Quality Discs carry a Lifetime Guarantee. They come in their own Library case and represent real product excellence in highly competitive prices.

05/00-45 sp	_____	Per box of 10
05/00-48 sp	_____	£7.90
05/00-48 sp	_____	£30.00
05/00-48 sp	_____	£10.00

### Seaside 5 1/4" Microfiche Lifetime Guarantee

05/00-100 sp	_____	Per box of 10
05/00-100 sp	_____	£81.75
05/00-100 sp	_____	£30.44

### Seaside 5 1/4" Floppy Disc Lifetime Guarantee

05/00-48 sp	_____	Per box of 10
05/00-48 sp	_____	£11.00
05/00-48 sp	_____	£15.00
05/00-48 sp	_____	£15.00

Authorised Seaside Distributor

Micro Resources offer special quantity discount prices for Schools, Offices and Education Centres. Prices start at 50p per disc in bulk packed lots, fixed charges on call up with your Access or Services to number. Orders are despatched within 24 hours. Prices include delivery but exclude VAT. An hour after-noon service.



## MICRO RESOURCES Ltd.

South Hill House, 11 Liverpool Gardens, Worthing,  
Sussex BN1 1 1LN  
Telephone Worthing (0903) 213174

# DATABASE SOFTWARE

**BRITAIN'S MOST INNOVATIVE SOFTWARE HOUSE**

*As part of our further development into business software  
we are seeking the following full-time personnel:*

### COMPUTER PROGRAMMERS

You will write software for 6502, 280 and 68000 based home and business computers in high and low level languages. As an active member of a programming team you will be jointly responsible for the design and development of various exciting software packages. You will be expected to show enthusiasm to master new machines as they appear on the market. Fair and initiative are qualities which we expect in all our programmers. You must have a proven programming ability and a determination to complete your work to the highest standards.

### QUALITY CONTROLLERS

You will work as a member of a team and be expected to interact closely with the programmers during all stages of software development. This will involve aiding program design, testing and bug tracing, constructive program appraisal and assistance in the production of documentation. Ideally you should be reasonably conversant with several different business packages.

For an application, *please* apply in writing to:  
**The Personnel Manager, Database Software, Europa House,  
68 Chester Road, Hazel Grove, Stockport SK7 5NY**

## Platform yawns

**Program:** *Sam & Max's Morbo* (A)  
**Price:** £7.99 (catalogue) £5.99 (retail)  
**Supplier:** American Software, c/o US Gold, one 2 and 3, Holford Way, Holford, Birmingham B5 7AR. Tel: 021-250 2388

Sorry to have to report that, despite the intriguing title, this game is nothing more than another one off the seemingly interminable production line of platform games. In fact, the only unusual thing about it is the title. Score zero for originality.

The game is described on the packaging thus: "32 fast action screens of action-packed exciting fun as Burt wanders through the Blewing Factory".

There may well be 32 screens but after a few plays I was so bored that I had no desire to put that statement to

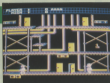
the test. The screens I did manage to play through were certainly neither fast-action, exciting or action-packed. Slow-action, monotonous and yawn-packed would be more apt.

The Burt of the alternative title is a blooby pink figure who can run left and right (taking a bit of time to get up steam), and can jump, sit down and fire bullets.

The game begins outside the factory where there are separate entrances for shipping, production and control, each of which leads to a different part of the factory.

Basically, each screen is a series of stages dotted with doors through which the meemies, without which no platform game is complete, appear.

Back shooting means looking exactly the same as the last shooting meemie. Kill one and



another eventually appears from the same door.

As well as meemies, platforms and doors, there are moving trucks, escalators, rolling barrels and blowers, the latter gently floating Burt upwards to a higher platform. Moving off a certain edge of most screens usually takes you to another screen.

Don't ask me what the object of the game is - the minimal instructions were silent on that point. About the only thing they do tell you is that the game can be played by one or two players.

The gameplay was dull, the graphics mediocre, the use of colour unimaginative, the animation basic and the sound below average.

Granted it is inexpensive, but if you really want a cheap but enjoyable platform game, better by far are *Duke's Gooies* and *Nuclear Rick* from the same company.

**Bob Chappell**

Sound	4
Graphics	4
Playability	4
Value	5
Overall	4

## Platform fun

**Program:** *Nuclear Nick*  
**Price:** £7.99 (catalogue) £4.99 (retail)  
**Supplier:** American Software, c/o US Gold, one 2 and 3, Holford Way, Holford, Birmingham B5 7AR. Tel: 021-250 2388

**SURPRISE,** surprise. *Nuclear Nick* is yet another platform game. Nothing wrong with that provided the game challenges and entertains - and this one certainly does.

You control Nick, the usual running, jumping bumbler, whose job it is to visit around various platforms grabbing the goodies and avoiding the bad-ies.

The basic villains of the piece are parading red-poly robots whose very touch spells lightning and fire out there. Although Nick has no weapons, he can strike back by setting what appear to be - I hope you're ready for this -

reflective hamburgers!

I was asked because *US Gold* set no traditions with me review copy. But given the title, the flashing of said burgers, the fact that Nick immediately starts to pulsate and glow after feeding his face, and the dire consequences for the robots, it seems a fair deduction.

That being said, the game deserves to win the equivalent of the Jules Rimet trophy for daffy plots, not to mention the master of questionable taste (no pun intended).

Gathering down a jump-burger gives Nick several seconds worth of reflective indignation, enabling him to neutralise as many robots as he can reach before his digestive system returns to normal. Once victorious, Nick can progress to the next screen.

Further points can be amassed by gathering up small



suspended bricks (nuclear waste), anticlock tables (I along the way. Nick has five lives and 30 screens to cross, each longer than the last.

While the robots and burgers are always the same in appearance if not quantity, each screen has a different configuration of platforms.

The graphics are crisp and attractive and, though there is no music, sound effects are generous and, I think, appropriate (how you hear exploded a reflective burger?).

The game is simple to get

into, while the difficulty of the screens range from taking candy from a baby to explosive delight.

Despite its dubious scenario and questionable content, *Nuclear Nick* is a pretty good platform game. And at the cheap asking price, it's a genuine bargain.

**Bob Chappell**

Sound	7
Graphics	8
Playability	8
Value	8
Overall	8



## It's a great war...

**Program:** Beach-Head '91  
**Price:** £24.99  
**Supplier:** Access Software, c/o  
SoftSource Europe,  
514-518 Aston Road,  
Aston Road, Birmingham  
B9 2AR. Tel: 027-328  
3585.

As you'll have gathered from the title, this is the long-awaited follow-up to the very successful Beach-Head. And for the money, this is better than the original.

Subtitled The Dictator Strikes Back!, it is a two-player game with your opponent being either another player or the computer. Against the computer, you pick a master which side you play, for the computer will control the other.

Like its predecessor, Beach-Head II has four distinct sequences. The first, Attack, is probably the best in it: the Allies drop troops from a helicopter ready to begin the assault against the Dictator's stronghold.

The troops have to be parachuted evenly behind a

four-section rear wall from which they must advance down the screen to a two-section wall.

From here they scramble towards the door of the enemy's lab and out of reach of the machine-gun which all the time has been rattling away at them leaving a postmarked battlefield.

Men can be sent over the wall as a diversion and grenades can be lobbed at the gun, but the ultimate objective is to get at least one man to the door.

In Escape, escaping footpads have to flee across an open courtyard. The Dictator has four methods of stopping them - dropping stones from a wall, placing mines from two-storey buildings in the courtyard, and sending out tanks and rocket-launching trucks. The Allies try to fend off all these attacks with a captured machine gun.

Escape has the footpads go in a helicopter which is under attack from the Dictator. Obviously this is the weakest of the four scenarios, though



will prove exciting to play.

The final sequence Battle, has the Dictator (the Dragon) and the head of the Allies (J.P. Straker) facing each other in a head-to-head across an underground river.

Mounted on long platforms, the two bull ponies (sharpened wooden sticks, it says here) at each other for several rounds.

The throws can be straight or curved and small speech bubbles appear to let you know the rival's innermost thoughts.

First one to get incited with four ponies is a stay and falls off the platform and into the river.

There is an option to

practice any screen and an automatic demo mode to show you how it should be done.

I missed the speech synthesis of the Commodore 64 version but even so, this is still a superb game.

The animated action and use of sound are a treat. The game offers bags of thrills and spills and is one of the best releases for the Atari this year. Not to be missed.

**Bob Chappell**

Sound	.....	A
Graphics	.....	B
Playability	.....	B
Value	.....	A
Overall	.....	B

## Good simulation

**Program:** Solo Flight 2  
**Price:** £9.99 (unrated),  
£7.99 (rated)  
**Supplier:** US Data, c/o P and  
P, Millard Way, Millard  
Birmingham B1 4AR. Tel:  
021-284 3388

SOLO Flight is an American real-time flight simulator which has been around quite some time. This is an updated version, providing a better cockpit control layout and a built-in flight trainer.

The simulation is based around the light monoplanes of the late 1920s and early 1930s, particularly the Ryan S-7 series, and offers flying practice together with a Mini Pilot game in which you can put your learned flight skills to

a considerably sterner test.

The screen display is unusual. The bottom half is taken up with the expected cockpit controls while the remainder provides a view not only of the landscape but also, oddly enough, of your plane.

Your viewpoint is from just above and behind your plane, from which a shadow is cast at low altitudes. You can also switch to a left, right or backward view from the cockpit.

While it might seem a bit strange at first, it works out quite well and makes flying the plane that much easier.

The comprehensive and clear instrumentation includes indicators for throttle, airspeed, altitude, artificial horizon, vertical velocity, Rps,



and VQR needle. There is also an altimeter, fuel gauge, lights for brake, landing gear and engine overheat, OME (oil) measuring equipment, LS (loss) scope indicator and magnetic compass.

In Flying Practice mode you can select the American state you will fly across (Kansas, Washington or Colorado) and the weather conditions: clear, windy or IFR (Instrument Flight

rules - low cloud).

Or if you prefer, you can simply practice landing.

You can also be accompanied by an on-board flying instructor. The instructor takes the form of text messages such as "You are too high. Push forward," which appear across the bottom of the screen.

Such messages repeat until you obey it, but the lullian can

be switched off at any time by a single command. There is no speech synthesis in the Amiga version.

In Mail Run mode your job is to deliver sacks of mail across different states to their destinations in as short a time as possible. You can select the level of difficulty from student (easiest) to commercial pilot (most difficult) and decide which destinations and how



much mail and fuel to carry. As the game progresses

weather conditions deteriorate and, depending on the skill level selected, your plane may develop mechanical and instrument malfunctions.

The graphics are fairly satisfactory and the display a little slow and jerky.

Even so, this unusual flight simulator is pleasantly easy to get to grips with, particularly given the presence of the onboard instructor. The Mail

Run option is an added bonus. Enjoyable and entertaining, the new Solo Flight has much to commend it.

Bob Chappell

Sound	5
Graphics	7
Playability	8
Value	8
Overall	8

## The big topper

Program: *Ballyhoo*  
Price: £27.98  
Supplier: Infocom, c/o Data Distribution, 7-8 The Arcade, Watney Road, Slough, Berks SL1 4 4XX, Tel: 07-038 0203.

WHAT connection is there between a sucker, an Annie Oakley and a First of May? All three info shouted "Discus!" give yourself a cuddly top. To put the rest of you in the picture, a sucker is a clown-gate, an Annie Oakley is a talent and a First of May is a novice circus performer.

Now the only reason I

happen to be knowledgeable about circus lingo is because I've been playing *Ballyhoo*. Infocom's latest standard level text adventure, in *Ballyhoo*, all the action takes place in a circus where the stardom of this particular big top is matched only by the usefulness of the performers.

Learning after the show was over, I managed to eavesdrop on a conversation between the circus owner, Mumbur, trying to spill it backword), and a private detective. It seems that Chateau, the owner's daughter, has been kidnapped and might have

As you budge along in the wake of the outflanking crowd, you thrust through your memories of this evening. Your experience of the circus, with its ballyhooed promises of wonderment and its ultimate disappointment, has been to sink your teeth into a candy apple whose fruit is rotten.

Never will the outrageous prices, the ill-conceived costumes, and the

been stashed somewhere on the circus lot. Since the gumshoe appears to be about as competent as Clouseau, I decided upon a little sleuthing of my own.

The adventure is full of fun and originality and I got to meet many interesting characters. For instance, there is Tina, a lady who for she-occupies two locations at once! When I tried to get a little closer to her massive bulk I was told "The slope's too steep!"

There is also to Conrade Thurst, a diminutive Russian, and the innately named Chuzzle, a curly clown, than first means the see while a visit to Andrew Jerry (half man, half woman) proves a most confusing experience. And having my bumps, fish, palm read and being placed in a typocric trance by Rintshaw the Incomprehensible are not things I shall forget in a hurry either.

Then there are the animals. Fancying my chances as a lion tamer, I stooped boldly into the lion's cage to put them through their paces. Exit stage right, stooped to pieces but somewhat wiser. Maybe there was better luck to be had with Mahler, the fearsome gorilla. The question was — should I

enter his cage wearing that ridiculous gorilla costume I'd found lying about?

Drawing a veil over my experiences with Mahler, I turned to my exploits as a high wire artist. Here I had the opportunity to excel — and would have done it some doubtabout hadn't removed the safety net when I wasn't locking. Result — back again to being the human plaything in one of the sideshows.

The mystery and danger intrinsic to the plot of *Ballyhoo* is well-seasoned with wit and humour. I particularly enjoyed an encounter with the Egress — a rare but fearless mammal — and, elsewhere, being able to vent my feelings during the moment between doing something painful and actually feeling the pain.

Yes, Infocom has done it yet again. *Ballyhoo* is a superb adventure, full of intrigue, originality and intelligent humour. Don't miss it.

Bob Chappell

Sound	9/10
Graphics	9/10
Presentation	10
Value	10
Overall	10



# Now let's get things moving...

## Part Four of STEPHEN WILLIAMSON'S series on player missile graphics



**THIS** time I will show how to move players around the screen by using animation techniques.

Program 1 demonstrates the basic method for moving players. The first section, lines 50 to 180, contains the standard player missile initialization routines that you will be familiar with if you have followed this series so far.

POKE 165,0 in line 50 switches off the cursor, so if you stop the program and want to switch the cursor back on, either press Reset or type POKE 165,2.

Moving a player horizontally is straightforward. You simply increase or decrease the value held in the horizontal register corresponding to the player you wish to move. A list of

these registers was given last month, but in case you do not have a copy handy, they can be summarised as follows:

- Registers 53248 to 53251 are the horizontal registers for players 0 to 3.
- Registers 53252 to 53255 are the registers for missiles 0 to 3.

One problem with the horizontal registers is that you cannot PEEK them to find out the current horizontal position of the player or missile stripe. If you type

**POKE 53248, 140:  
PRINT PEEK (53248)**

you would expect the return to print the number 140, the value you have just stored in location 53248. Instead

you get the number 0. So where has 140 gone? The Antic chip has taken 140 into its inner workings, altered the horizontal position of the player stripe accordingly and then immediately reset the value held at 53248 to 0 before the PRINT PEEK (53248) instruction has been handed. This is all very inconvenient if you want to keep track of the horizontal co-ordinate of the player stripe.

For most animation procedures we need to assign a variable to hold the current player stripe position. Line 170 of Program 1 uses the variable MOVX2 to store the current horizontal value of the player 0 stripe, which is initially 130.

Line 180 assigns another variable,

```

10 DIM B(255)
20 DIM P(4)
30 DIM M(4)
40 DIM S(4)
50 POKE 165,0
60 GOSUB 1000:GOTO 100
70
80 POKE 165,2
90 GOSUB 1000
100 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 110
110 B(I)=0
120 NEXT I
130 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 140
140 P(I)=130
150 NEXT I
160 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 170
170 M(I)=0
180 NEXT I
190 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 200
200 S(I)=0
210 NEXT I
220
230 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 240
240 GOTO 250
250 NEXT I
260
270 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 280
280 GOTO 290
290 NEXT I
300
310 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 320
320 GOTO 330
330 NEXT I
340
350 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 360
360 GOTO 370
370 NEXT I
380
390 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 400
400 GOTO 410
410 NEXT I
420
430 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 440
440 GOTO 450
450 NEXT I
460
470 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 480
480 GOTO 490
490 NEXT I
500
510 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 520
520 GOTO 530
530 NEXT I
540
550 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 560
560 GOTO 570
570 NEXT I
580
590 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 600
600 GOTO 610
610 NEXT I
620
630 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 640
640 GOTO 650
650 NEXT I
660
670 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 680
680 GOTO 690
690 NEXT I
700
710 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 720
720 GOTO 730
730 NEXT I
740
750 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 760
760 GOTO 770
770 NEXT I
780
790 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 800
800 GOTO 810
810 NEXT I
820
830 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 840
840 GOTO 850
850 NEXT I
860
870 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 860
860 GOTO 870
870 NEXT I
880
890 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 880
890 GOTO 890
890 NEXT I
900
910 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 900
910 GOTO 910
910 NEXT I
920
930 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 920
930 GOTO 930
930 NEXT I
940
950 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 940
950 GOTO 950
950 NEXT I
960
970 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 960
970 GOTO 970
970 NEXT I
980
990 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 980
990 GOTO 990
990 NEXT I
1000
1010 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1010
1010 GOTO 1010
1010 NEXT I
1020
1030 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1030
1030 GOTO 1030
1030 NEXT I
1040
1050 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1050
1050 GOTO 1050
1050 NEXT I
1060
1070 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1070
1070 GOTO 1070
1070 NEXT I
1080
1090 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1090
1090 GOTO 1090
1090 NEXT I
1100
1110 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1110
1110 GOTO 1110
1110 NEXT I
1120
1130 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1130
1130 GOTO 1130
1130 NEXT I
1140
1150 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1150
1150 GOTO 1150
1150 NEXT I
1160
1170 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1170
1170 GOTO 1170
1170 NEXT I
1180
1190 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1190
1190 GOTO 1190
1190 NEXT I
1200
1210 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1210
1210 GOTO 1210
1210 NEXT I
1220
1230 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1230
1230 GOTO 1230
1230 NEXT I
1240
1250 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1250
1250 GOTO 1250
1250 NEXT I
1260
1270 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1270
1270 GOTO 1270
1270 NEXT I
1280
1290 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1290
1290 GOTO 1290
1290 NEXT I
1300
1310 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1310
1310 GOTO 1310
1310 NEXT I
1320
1330 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1330
1330 GOTO 1330
1330 NEXT I
1340
1350 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1350
1350 GOTO 1350
1350 NEXT I
1360
1370 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1370
1370 GOTO 1370
1370 NEXT I
1380
1390 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1390
1390 GOTO 1390
1390 NEXT I
1400
1410 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1410
1410 GOTO 1410
1410 NEXT I
1420
1430 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1430
1430 GOTO 1430
1430 NEXT I
1440
1450 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1450
1450 GOTO 1450
1450 NEXT I
1460
1470 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1470
1470 GOTO 1470
1470 NEXT I
1480
1490 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1490
1490 GOTO 1490
1490 NEXT I
1500
1510 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1510
1510 GOTO 1510
1510 NEXT I
1520
1530 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1530
1530 GOTO 1530
1530 NEXT I
1540
1550 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1550
1550 GOTO 1550
1550 NEXT I
1560
1570 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1570
1570 GOTO 1570
1570 NEXT I
1580
1590 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1590
1590 GOTO 1590
1590 NEXT I
1600
1610 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1610
1610 GOTO 1610
1610 NEXT I
1620
1630 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1630
1630 GOTO 1630
1630 NEXT I
1640
1650 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1650
1650 GOTO 1650
1650 NEXT I
1660
1670 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1670
1670 GOTO 1670
1670 NEXT I
1680
1690 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1690
1690 GOTO 1690
1690 NEXT I
1700
1710 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1710
1710 GOTO 1710
1710 NEXT I
1720
1730 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1730
1730 GOTO 1730
1730 NEXT I
1740
1750 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1750
1750 GOTO 1750
1750 NEXT I
1760
1770 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1770
1770 GOTO 1770
1770 NEXT I
1780
1790 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1790
1790 GOTO 1790
1790 NEXT I
1800
1810 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1810
1810 GOTO 1810
1810 NEXT I
1820
1830 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1830
1830 GOTO 1830
1830 NEXT I
1840
1850 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1850
1850 GOTO 1850
1850 NEXT I
1860
1870 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1870
1870 GOTO 1870
1870 NEXT I
1880
1890 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1890
1890 GOTO 1890
1890 NEXT I
1900
1910 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1910
1910 GOTO 1910
1910 NEXT I
1920
1930 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1930
1930 GOTO 1930
1930 NEXT I
1940
1950 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1950
1950 GOTO 1950
1950 NEXT I
1960
1970 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1970
1970 GOTO 1970
1970 NEXT I
1980
1990 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1990
1990 GOTO 1990
1990 NEXT I
2000
2010 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2010
2010 GOTO 2010
2010 NEXT I
2020
2030 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2030
2030 GOTO 2030
2030 NEXT I
2040
2050 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2050
2050 GOTO 2050
2050 NEXT I
2060
2070 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2070
2070 GOTO 2070
2070 NEXT I
2080
2090 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2090
2090 GOTO 2090
2090 NEXT I
2100
2110 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2110
2110 GOTO 2110
2110 NEXT I
2120
2130 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2130
2130 GOTO 2130
2130 NEXT I
2140
2150 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2150
2150 GOTO 2150
2150 NEXT I
2160
2170 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2170
2170 GOTO 2170
2170 NEXT I
2180
2190 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2190
2190 GOTO 2190
2190 NEXT I
2200
2210 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2210
2210 GOTO 2210
2210 NEXT I
2220
2230 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2230
2230 GOTO 2230
2230 NEXT I
2240
2250 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2250
2250 GOTO 2250
2250 NEXT I
2260
2270 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2270
2270 GOTO 2270
2270 NEXT I
2280
2290 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2290
2290 GOTO 2290
2290 NEXT I
2300
2310 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2310
2310 GOTO 2310
2310 NEXT I
2320
2330 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2330
2330 GOTO 2330
2330 NEXT I
2340
2350 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2350
2350 GOTO 2350
2350 NEXT I
2360
2370 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2370
2370 GOTO 2370
2370 NEXT I
2380
2390 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2390
2390 GOTO 2390
2390 NEXT I
2400
2410 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2410
2410 GOTO 2410
2410 NEXT I
2420
2430 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2430
2430 GOTO 2430
2430 NEXT I
2440
2450 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2450
2450 GOTO 2450
2450 NEXT I
2460
2470 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2470
2470 GOTO 2470
2470 NEXT I
2480
2490 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2490
2490 GOTO 2490
2490 NEXT I
2500
2510 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2510
2510 GOTO 2510
2510 NEXT I
2520
2530 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2530
2530 GOTO 2530
2530 NEXT I
2540
2550 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2550
2550 GOTO 2550
2550 NEXT I

```

```

1000
1010 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1010
1010 GOTO 1010
1010 NEXT I
1020
1030 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1030
1030 GOTO 1030
1030 NEXT I
1040
1050 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1050
1050 GOTO 1050
1050 NEXT I
1060
1070 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1070
1070 GOTO 1070
1070 NEXT I
1080
1090 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1090
1090 GOTO 1090
1090 NEXT I
1100
1110 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1110
1110 GOTO 1110
1110 NEXT I
1120
1130 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1130
1130 GOTO 1130
1130 NEXT I
1140
1150 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1150
1150 GOTO 1150
1150 NEXT I
1160
1170 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1170
1170 GOTO 1170
1170 NEXT I
1180
1190 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1190
1190 GOTO 1190
1190 NEXT I
1200
1210 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1210
1210 GOTO 1210
1210 NEXT I
1220
1230 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1230
1230 GOTO 1230
1230 NEXT I
1240
1250 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1250
1250 GOTO 1250
1250 NEXT I
1260
1270 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1270
1270 GOTO 1270
1270 NEXT I
1280
1290 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1290
1290 GOTO 1290
1290 NEXT I
1300
1310 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1310
1310 GOTO 1310
1310 NEXT I
1320
1330 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1330
1330 GOTO 1330
1330 NEXT I
1340
1350 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1350
1350 GOTO 1350
1350 NEXT I
1360
1370 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1370
1370 GOTO 1370
1370 NEXT I
1380
1390 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1390
1390 GOTO 1390
1390 NEXT I
1400
1410 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1410
1410 GOTO 1410
1410 NEXT I
1420
1430 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1430
1430 GOTO 1430
1430 NEXT I
1440
1450 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1450
1450 GOTO 1450
1450 NEXT I
1460
1470 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1470
1470 GOTO 1470
1470 NEXT I
1480
1490 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1490
1490 GOTO 1490
1490 NEXT I
1500
1510 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1510
1510 GOTO 1510
1510 NEXT I
1520
1530 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1530
1530 GOTO 1530
1530 NEXT I
1540
1550 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1550
1550 GOTO 1550
1550 NEXT I
1560
1570 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1570
1570 GOTO 1570
1570 NEXT I
1580
1590 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1590
1590 GOTO 1590
1590 NEXT I
1600
1610 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1610
1610 GOTO 1610
1610 NEXT I
1620
1630 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1630
1630 GOTO 1630
1630 NEXT I
1640
1650 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1650
1650 GOTO 1650
1650 NEXT I
1660
1670 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1670
1670 GOTO 1670
1670 NEXT I
1680
1690 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1690
1690 GOTO 1690
1690 NEXT I
1700
1710 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1710
1710 GOTO 1710
1710 NEXT I
1720
1730 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1730
1730 GOTO 1730
1730 NEXT I
1740
1750 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1750
1750 GOTO 1750
1750 NEXT I
1760
1770 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1770
1770 GOTO 1770
1770 NEXT I
1780
1790 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1790
1790 GOTO 1790
1790 NEXT I
1800
1810 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1810
1810 GOTO 1810
1810 NEXT I
1820
1830 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1830
1830 GOTO 1830
1830 NEXT I
1840
1850 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1850
1850 GOTO 1850
1850 NEXT I
1860
1870 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1870
1870 GOTO 1870
1870 NEXT I
1880
1890 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1890
1890 GOTO 1890
1890 NEXT I
1900
1910 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1910
1910 GOTO 1910
1910 NEXT I
1920
1930 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1930
1930 GOTO 1930
1930 NEXT I
1940
1950 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1950
1950 GOTO 1950
1950 NEXT I
1960
1970 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1970
1970 GOTO 1970
1970 NEXT I
1980
1990 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 1990
1990 GOTO 1990
1990 NEXT I
2000
2010 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2010
2010 GOTO 2010
2010 NEXT I
2020
2030 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2030
2030 GOTO 2030
2030 NEXT I
2040
2050 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2050
2050 GOTO 2050
2050 NEXT I
2060
2070 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2070
2070 GOTO 2070
2070 NEXT I
2080
2090 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2090
2090 GOTO 2090
2090 NEXT I
2100
2110 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2110
2110 GOTO 2110
2110 NEXT I
2120
2130 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2130
2130 GOTO 2130
2130 NEXT I
2140
2150 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2150
2150 GOTO 2150
2150 NEXT I
2160
2170 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2170
2170 GOTO 2170
2170 NEXT I
2180
2190 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2190
2190 GOTO 2190
2190 NEXT I
2200
2210 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2210
2210 GOTO 2210
2210 NEXT I
2220
2230 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2230
2230 GOTO 2230
2230 NEXT I
2240
2250 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2250
2250 GOTO 2250
2250 NEXT I
2260
2270 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2270
2270 GOTO 2270
2270 NEXT I
2280
2290 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2290
2290 GOTO 2290
2290 NEXT I
2300
2310 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2310
2310 GOTO 2310
2310 NEXT I
2320
2330 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2330
2330 GOTO 2330
2330 NEXT I
2340
2350 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2350
2350 GOTO 2350
2350 NEXT I
2360
2370 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2370
2370 GOTO 2370
2370 NEXT I
2380
2390 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2390
2390 GOTO 2390
2390 NEXT I
2400
2410 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2410
2410 GOTO 2410
2410 NEXT I
2420
2430 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2430
2430 GOTO 2430
2430 NEXT I
2440
2450 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2450
2450 GOTO 2450
2450 NEXT I
2460
2470 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2470
2470 GOTO 2470
2470 NEXT I
2480
2490 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2490
2490 GOTO 2490
2490 NEXT I
2500
2510 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2510
2510 GOTO 2510
2510 NEXT I
2520
2530 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2530
2530 GOTO 2530
2530 NEXT I
2540
2550 FOR J=0 TO 3:FOR I=0 TO 255:GOTO 2550
2550 GOTO 2550
2550 NEXT I

```

POKE 165,2:PRINT PEEK (53248):PRINT PEEK (53248)

**Get it right!**

POKE	PEEK	POKE	PEEK
100	130	100	130
101	130	101	130
102	130	102	130
103	130	103	130
104	130	104	130
105	130	105	130
106	130	106	130
107	130	107	130
108	130	108	130
109	130	109	130
110	130	110	130
111	130	111	130
112	130	112	130
113	130	113	130
114	130	114	130
115	130	115	130
116	130	116	130
117	130	117	130
118	130	118	130
119	130	119	130
120	130	120	130



5 to hold the value of STICK(0) which is the status of the joystick. The values are 11 - left, 7 - right, 14 - up, 13 - down and 15 - centre.

Lines 190 and 200 check to see whether S is 11 or 7 - joystick pointing left or right and alters the value of HORIZZ by one according to which direction the player is to move.

Line 210 POK is the value held in HORIZZ into the horizontal register for player 0, to move the player stripe one position to the right or left. Each movement of the player stripe is equal to the width of one pixel.

The display at the bottom of the screen gives the current horizontal value. At a value of 48 the player starts to leave the blue area and moves across the background until, if you continue to move the joystick left, it will move off the screen. You can continue decreasing the horizontal value although this will have no effect on the screen display.

If the value of HORIZZ goes below 0 or above 255 you will get an error message as the program reaches line 210 and tries to put the value of HORIZZ into the horizontal register 33046. This is because you cannot give a number more than one byte or less than 0 into a single memory address.

To avoid this problem, it is normal for a player missile program to contain error trapping routines. Program 1 does not have error routines so that you can experience what happens when you try to move a player too far, but Program 2 does include one.

So much for horizontal movement, which is simple to operate. When it comes to vertical movement things are a little more complicated. It is not actually possible to move the player stripe vertically. It will always stretch from the top of the screen to the bottom, no matter how many or how few pixels are plotted within it.

We saw last month how, when you store the shape data into the appropriate player data area, the distance within that area from which you start to load the data determines the vertical position of the shape. This is why, in line 100, the data for the player shape is loaded 140 bytes from the base of the player 0 data area to make sure that the bug appears part way down the screen.

```

10 DIM HORIZZ(255)
20 DIM STICK(15)
30 DIM SHAPES(10,255)
40 DIM VERT(255)
50 GOSUB 900:GOTO 1,2,3:POKE 700,0
60
70 FOR I=0 TO 255
80   FOR J=0 TO 255:POKE 33046+I,0:NEXT J
90 NEXT I
100 FOR I=0 TO 10000:GOTO 3000:POKE 33046+I,0:NEXT I
110 FOR I=0 TO 255
120   FOR J=0 TO 255:POKE 33046+I,0:NEXT J
130 NEXT I
140
150 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
160
170 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
180
190 IF STICK(0)=11 THEN HORIZZ=HORIZZ-1
200 IF STICK(0)=7 THEN HORIZZ=HORIZZ+1
210 POK HORIZZ
220
230 FOR I=0 TO 255:PRINT I:NEXT I
240
250 GOTO 100

```

Program 1

If you alter line 100 by changing the 140 to 141, the bug will appear one pixel further down the screen. 139 would display a player one pixel further up.

It should now be obvious how to give the illusion of vertical movement. We need a routine to move the whole of the shape data one byte further up or one byte further down in memory. Line 200 performs this function.

First, line 170 assigns the variable VERT to hold the current value of the vertical position of the player in the same way as the variable HORIZZ.

Lines 230 and 250 check for the up and down positions of the joystick and increase or decrease the value of VERT accordingly. If the joystick has been moved up or down then the variable V is set to one. V acts as a flag. When V is one then the vertical movement routine in line 260 is actioned. When V is zero, and therefore no vertical movement is required, the routine in line 280 is ignored.

The routine in line 260 is similar to

```

270 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
280
290 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
300
310 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
320
330 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
340
350 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
360
370 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
380
390 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
400
410 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
420
430 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
440
450 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
460
470 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
480
490 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
500
510 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
520
530 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
540
550 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
560
570 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
580
590 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
600
610 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
620
630 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
640
650 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
660
670 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
680
690 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
700
710 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
720
730 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
740
750 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
760
770 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
780
790 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
800
810 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
820
830 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
840
850 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
860
870 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
880
890 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
900
910 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
920
930 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
940
950 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
960
970 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
980
990 FOR I=0 TO 255:POKE 33046+I,0:NEXT I
1000

```



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

180 allows the data to be re-read in line 280.

You probably won't be very impressed by the Program I demonstration of vertical movement. The bug is not very lively, traveling more like a snail than an insect, with a rippling effect as the pixels are replotted.

There is not much that can be done about this if we are to stick to Basic. The plotting routine in line 280 takes time, and there will always be a noticeable delay as the program keeps updating the information held in the player data area.

If you want to add some zest to the program, you have to enter the realms of machine code.

Program II is an improved version of Program I that includes two machine code routines, one for up movement and the other for down.

Don't worry if you do not have a disc about machine code. The two routines can be copied and included for use in your own player missile programs without having to understand exactly how they work.

If you run Program II it will at first look very much like Program I with the same bug waiting for you to move it with the joystick. But when you move it up or down you will see that the animation is much smoother.

Line 170 reads the data contained in lines 300 to 320 — the machine code routine — and stores it in the unused part of the player missile data area. This unused area is not needed by the player missile system and is 788 bytes long for single resolution players and 384 bytes long for double resolution players — plenty of room in which to store machine code routines. See the player missile data area map in *July's Aster Wars*.

The first routine — 21 bytes long — is placed in the unused area from PBASE upwards, and the second routine starts at PBASE plus 21.

The routines as they stand will control any player or missile up to 20 pixels in height. To use the routine for taller players, alter the number 20 in line 340 to two greater than the player height, and the 20 in line 360 to one greater.

Horizontal movement in Program II is handled in the same way as Program I. An error trapping routine has been inserted in lines 230 and 230 that stop *MORZ* becoming less

than 48 or more than 302. This means that the player cannot leave the playfield 0 area and, of course, it also prevents an error message occurring due to the value of *MORZ* going below 0 or above 255.

Lines 280 and 280 control vertical movement by calling up the machine code routines.

*VERT*, the vertical coordinate variable, is first checked to make sure that it is not greater than 32 or less than 212 — when the player is at the top or bottom edge of the playfield — before passing control of the program to the appropriate machine code routine.

The formula for using the routine is:

**AmUSB (machine code address, player address plus vertical position)**

Following this formula, the call-up

routine in line 280 becomes:

**AmUSB(PBASE,  
PBASE+1023+VERT)**

The up machine code routine starts at PBASE and the down routine at PBASE+21.

For the routines to operate correctly, *VERT* must be increased or decreased by no more than one at a time.

After the routines have been called up and the player replotted, *VERT* is adjusted by one ready for the next plotting operation.

Once you know the correct data for the up and down machine code routines then operating them is no more difficult than the pure Basic routine used in Program I — but the effect is much better.

Program III introduces another animation effect and another machine code routine. In this program the

```

100 REM *****PROGRAM II*****
110 REM *****MACHINE CODE*****
120 REM *****UP*****
130 REM *****DOWN*****
140 REM *****MORZ*****
150 REM *****MORZ*****
160 REM *****MORZ*****
170 REM *****MORZ*****
180 REM *****MORZ*****
190 REM *****MORZ*****
200 REM *****MORZ*****
210 REM *****MORZ*****
220 REM *****MORZ*****
230 REM *****MORZ*****
240 REM *****MORZ*****
250 REM *****MORZ*****
260 REM *****MORZ*****
270 REM *****MORZ*****
280 REM *****MORZ*****
290 REM *****MORZ*****
300 REM *****MORZ*****
310 REM *****MORZ*****
320 REM *****MORZ*****
330 REM *****MORZ*****
340 REM *****MORZ*****
350 REM *****MORZ*****
360 REM *****MORZ*****
370 REM *****MORZ*****
380 REM *****MORZ*****
390 REM *****MORZ*****
400 REM *****MORZ*****
410 REM *****MORZ*****
420 REM *****MORZ*****
430 REM *****MORZ*****
440 REM *****MORZ*****
450 REM *****MORZ*****
460 REM *****MORZ*****
470 REM *****MORZ*****
480 REM *****MORZ*****
490 REM *****MORZ*****
500 REM *****MORZ*****
510 REM *****MORZ*****
520 REM *****MORZ*****
530 REM *****MORZ*****
540 REM *****MORZ*****
550 REM *****MORZ*****
560 REM *****MORZ*****
570 REM *****MORZ*****
580 REM *****MORZ*****
590 REM *****MORZ*****
600 REM *****MORZ*****
610 REM *****MORZ*****
620 REM *****MORZ*****
630 REM *****MORZ*****
640 REM *****MORZ*****
650 REM *****MORZ*****
660 REM *****MORZ*****
670 REM *****MORZ*****
680 REM *****MORZ*****
690 REM *****MORZ*****
700 REM *****MORZ*****
710 REM *****MORZ*****
720 REM *****MORZ*****
730 REM *****MORZ*****
740 REM *****MORZ*****
750 REM *****MORZ*****
760 REM *****MORZ*****
770 REM *****MORZ*****
780 REM *****MORZ*****
790 REM *****MORZ*****
800 REM *****MORZ*****
810 REM *****MORZ*****
820 REM *****MORZ*****
830 REM *****MORZ*****
840 REM *****MORZ*****
850 REM *****MORZ*****
860 REM *****MORZ*****
870 REM *****MORZ*****
880 REM *****MORZ*****
890 REM *****MORZ*****
900 REM *****MORZ*****
910 REM *****MORZ*****
920 REM *****MORZ*****
930 REM *****MORZ*****
940 REM *****MORZ*****
950 REM *****MORZ*****
960 REM *****MORZ*****
970 REM *****MORZ*****
980 REM *****MORZ*****
990 REM *****MORZ*****
1000 REM *****MORZ*****

```



*****PROGRAM II*****		
10 00017	10 00017	10 00017
10 00018	10 00018	10 00018
10 00019	10 00019	10 00019
10 00020	10 00020	10 00020
10 00021	10 00021	10 00021
10 00022	10 00022	10 00022
10 00023	10 00023	10 00023
10 00024	10 00024	10 00024
10 00025	10 00025	10 00025
10 00026	10 00026	10 00026
10 00027	10 00027	10 00027
10 00028	10 00028	10 00028
10 00029	10 00029	10 00029
10 00030	10 00030	10 00030
10 00031	10 00031	10 00031
10 00032	10 00032	10 00032
10 00033	10 00033	10 00033
10 00034	10 00034	10 00034
10 00035	10 00035	10 00035
10 00036	10 00036	10 00036
10 00037	10 00037	10 00037
10 00038	10 00038	10 00038
10 00039	10 00039	10 00039
10 00040	10 00040	10 00040
10 00041	10 00041	10 00041
10 00042	10 00042	10 00042
10 00043	10 00043	10 00043
10 00044	10 00044	10 00044



LET'S give your computer the power to control mains appliances. It can be operated directly from the keyboard, by a Basic program, or by external sensors such as light or heat detectors.

There are two independent channels, expandable to four, and each channel can handle 750 watts. For now we'll look at the electronics and theory of operation. Next month we'll describe the practical construction and programming.

Safety is our first consideration because, no matter what design is used, the control box must have a mains cable going in at one end and a joystick lead coming out of the other.

If ever the twin should meet, your computer – or at very least its *Pld* chip – will rapidly depart this life, and may attempt to take you along with it.

This design has two separate stages of protection – an opto-isolator on the input circuit, and mechanical relays to handle the mains voltage output.

Let's look at the opto-isolator first. Figure 1 shows in schematic form how it works – you may recognise the symbols from last month. It has two components, housed together in a light-proof package. The input is a conventional light-emitting diode and the output is a phototransistor which switches off in darkness and on when illuminated.

The only link between input and output is a beam of infra red light, and there's no way that high voltages can cross the gap. Your computer sees only the LED side – all the dangerous work is done by circuitry which it never needs to know about.

Since our controller has two channels we need two opto-isolators, and for convenience we've used a dual type which has everything contained in a single 8 pin package. A second level of mains isolation is

# Controlling mains appliances

The accent is on safety in Part 3 of  
LEN GOLDING's series on using  
your Atari to control devices

achieved by using mechanical relays to do the actual switching. Relays come in many shapes and sizes, but they all work on the same principle – see Figure 2.

A coil of fine wire generates a magnetic field when current flows through it, and the magnetic attraction is used to operate one or more sets of switch contacts. The coil uses little power and can operate on low voltages, but the switch contacts can handle much higher voltages and currents.

Since the coil is heavily insulated from the switch contacts, mains voltages cannot get through to the low-voltage lines.

Figure 3 shows the circuit for the complete mains controller, with its three distinct stages. Stage 1 is the LED circuitry which connects to the computer via a joystick port, and operates on power from pin 7 (+5V).

Control signals are taken from pins 1 and 2 via resistors R1 and R2, to the

base leads of TR1 and TR2 when they are amplified and made to drive LEDs A and B inside the opto-isolator. Resistors R3 and R4 limit the current through each LED to around 9mA.

The two jacks are optional – they connect to the joystick port's analog inputs, so you can use external sensors – like light detectors or central heating thermostats – to control the switching action.

Stage 2 operates on a nominal 12V DC from transformer T1. The power supply is not critical, so the circuit uses a simple bridge rectifier – BR1 – and smoothing capacitor – C1. The output from this network is around 14V when no load is connected, falling to 12V or so when the maximum current – about 500mA – is drawn.

An auxiliary output is provided on the terminal block so that you can tap this supply for other purposes if you wish.

When phototransistor A switches

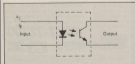


Figure 1: Opto-isolator

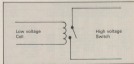
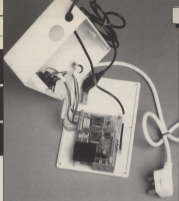


Figure 2: Mechanical relay



on its collector voltage falls towards 0V, causing TR3 to switch on. Current therefore flows through the coil of

Relay 1, and switch SW1 operates. The same happens with TR4. Relay 2 and SW2 when phototran-

sistor B is activated. Diodes D1 and D2 are included to suppress the voltage spikes which relay coils tend to generate.

Stage 3 is the 240V AC section containing the two relay switches, each with a capacitor across its contacts to suppress sparking and mains interference. You can see that if everything is wired up correctly there is no electrical path from the mains supply to the computer.

Even if stage 2 were to become live – for example after an insulation breakdown in the relays or transformer – the opto-isolator still provides protection.

A printed circuit board is used to hold all components as this greatly simplifies construction. If you have facilities to make your own, the foil pattern is shown in Figure IV with the drilling schedule in Figure V.

If you don't fancy making one yourself it's a lot easier to buy the commercially produced board from RH design – see details in the panel – especially since this comes drilled and tinned ready for use.

The basic design will suit most

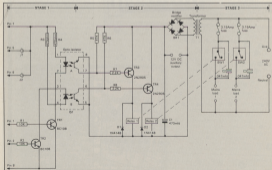


Figure III: Circuit of mains controller

domestic applications, and it can easily be adapted to handle specialised jobs. For example, you could use a different type of relay if won't fit on the PCB, but leads to the coil can be taken from points A and B - Relay 1, or C and D - Relay 2, which are marked on the board.

You could drive two or even three

relays simultaneously from each output should your application demand more current or a more complex switch configuration. It doesn't matter which way round the coils are connected, and the output transistors can each handle up to 500mA.

Mains currents in excess of 5

Amps should be kept off the PCB, so use a large screw terminal block or a junction box to connect the switch contacts of any off-board relays.

There is enough power available from the joystick parts to drive two extra channels, making four in all. Use two PCBs and connect joystick pins 3 and 4 in place of 1 and 2 on the second board.

The specified transformer will supply enough current for all four channels, so you can omit T1, BR1 and C1 from the second board, and just connect the 12V and 0V output terminals from one board to the other.

If you plan to drive several relays from each output ensure that their combined coils will not draw more than 500mA or you will need a more powerful off-board transformer. The bridge rectifier can handle 1.5 Amps at up to 35V.

■ Next month we'll cover the construction details, testing and programming. ■

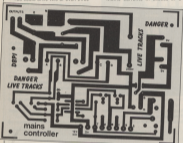


Figure 4: Full pattern for mains controller PCB

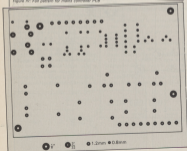


Figure 5: Drilling schedule for mains controller PCB

## PARTS LIST

Resistors: R1, R2 10k (brown/black/orange) Mafco order code M703; R3, R4 470ohm (yellow/violet/brown) M3308; R5, R6 67k (yellow/violet/orange) M67C; R7, R8 2.2k (red/violet/M3K2)

Capacitors: C1 470nfd 25V PC electrolytic FF765; C2, C3 0.047nfd 250V AC suppressor FF528C

Semiconductors: TR1, TR2 BC108 transistors Q8224; TR3, TR4 2N2006 transistors Q9177; D1, D2 1N4148 signal diodes Q4808; BR1 W005 bridge rectifier Q1275; BR2 dual optoisolator Y7825

Connectors: J1, J2 3.5mm PCB jack sockets F922C (optional); 1 Sp4 D1; IC socket F4527; 2 Euro facility chassis sockets PL42W (optional)

Miscellaneous: 2 12V flat relays AY20M; 1 PCB mains transformer 0V YK526; 2 Chassis fuse holders 20mm BR485; 2 3.15 anti-surge fuses 20mm R4 F18; 1 Joystick extension lead (eg Tandy 278-1678)

Printed circuit board - order code 08PT - available from R.H. Design, 137 Stonehill Avenue, Harrogate, North Yorks, Tel: 0423 506385. Price £3.95.

Other components available from Maglin Electronic Supplies Ltd, PO Box 3, Rayleigh, Essex S66 6LR. Tel: 0702 582911.

Approximate cost £18 plus case.

LET'S start with a success story. Darren Dodds from Tynes and Wear has the answer for Joy and Dave who were struggling with Quasimodo from Renault's little white book. They wanted to know how to get the second jewel back to its casing. Well here's how.

Come back after getting the jewel and climb down the ladder to the platform. From here you then jump onto the first rope in the series of three to the left, but you must not jump to the next rope until there is a gap in the bars.

The gap will appear after 10 bars and you can then proceed. Jump to the next rope and climb immediately to avoid being killed by the pesky bats. You can then jump across to the next rope when there is another gap.

Do not jump across to the last rope until a bat flies up behind you. Then jump to the next rope, go down and jump to the next ladder. You then do exactly the same for the next level.

So there you are, it was as simple as that. Darren signs himself as a newcomer to Atari Quest, and a very valuable one at that if I may say so.

Joy and Dave can now attempt to get the third jewel, while I try to find out how Darren can progress after getting the skeleton from the horse in English Software's Mediator. Perhaps Joy and Dave know the solution.

Two people are stuck on The Pay-Off from Signos/Atari in exactly the same place. They are both keeping each other company on top of the fire escape having disabled the alarm, yet they are unable to get through the window without disturbing the guard inside.

Moore, Purvis and Hillman are

# Going bats over Quasimodo

## By Brillig

desperate to solve this one for their sanity, and in Mr Purvis's case for the sake of his future marriage.

To get past this obstacle you must remember that a jewel thief requires sticky fingers, and this may lead you to a very professional method of stopping the glass from crashing into the bars and waking the doxy guard inside.

Of course you will still need to find a way of getting the glass from the window frame, so you may have to read more into this solution than is at first apparent.

Nigel Burton has come up with the answer to my question on Hitch Hikers Guide to the Galaxy, and the strange vocabulary identified by the Zorkbook kit.

Wimbu is the equivalent of screaming. However Woonbe is apparently the writer of a slimming guide book, Galaxis Woonbeers. As in all instances you should consult the source of infinite knowledge for further details. A large T-shirt for

your help is on its way Nigel, with my thanks.

It Myskow also contributed a neat little drawing of the Wimbu of Wootton County which although not the wisest, was a worthy entry. Mr Myskow is also struggling to get past Terrain in Morden's Quest from Melbourne House.

He assured me that he will persevere to solve this game, and I can tell him that he is closer to solving it than he thinks, in this game persistence will be rewarded time and time again.

For this clue, and a comprehensive map of the game, I am indebted to Peter Dean. The map is laid out very clearly and, as Peter comments, the password is the only obscure part of Morden's Quest.

From the map I see that you ought to be able to trace a route avoiding Tarax. That should give me enough time to find the answer to the question.

I also note from a previous letters column an enquiry about adventure writing programs for the Atari. The only known program is Adventure Writer from Codewriter Inc, who no longer trade.

Gilsoft, from who the program was licensed, has no plans to release The Quill for the Atari, so there will be no flood of amateur adventures as on other more fortunate machines.

The only adventure I know of that is so written for the Atari is The Quest of Abraham Cross. Andy Warwick advised me of its existence, and also of the Glitch of the Month, in Adventure Quest from Level 3.

At the glass door in the Black Tower you do not have to hold the Mist Stone, merely type IM and you will be there. A medium size T-shirt is on its way Andy, and thanks for the info.

Sorry I couldn't review the games I mentioned last month, but they never arrived! Better luck next time. ■







By  
STEWART  
BULLOUGH

OUR hero in this game is Frank, a greasy little fellow whose main aim in life is to collect and eat as much fruit as possible.

To this end you must guide him around eight different levels, each a little harder than the last, until he has devoured all of the fruit that lies scattered around.

Your time is limited though, and as you play you will see the timer at the bottom of the screen tick away relentlessly. You must find the fastest and safest route around each level in order to complete it before your time runs out.

Frank is moved left, right, up or down by moving the joystick in the appropriate direction. He can also jump by holding down the trigger button as you move the joystick. However he can only jump to something which is on the same level as he is.

If Frank can get right to the edge of a platform he can often jump that little bit further, and there are many shortcuts to be found by using this method.

Scoring is very straightforward — for each piece of fruit collected you get 10 points, and each time unit remaining upon completion of a level will gain you a bonus of 100 points.

After you've completed all eight levels you go back to level one at a higher difficulty setting.

There are four difficulty levels, and you can choose them from the main title screen by using the Select button. The higher the level, the less time you will get to finish each screen.

The machine code PMG routine was written by Anthony Hughes. This little routine is really good, and can store up to 48 different frames, so credit where credit's due for a great sub-routine.

After typing in the listing check it with the Get-It-Right! program — see Page 31 — and when it is all correct SAVE yourself a copy and simply type RUN.

The initialization will take just over a minute, so be patient and wait for the game to begin.

## FRANK THE FRUIT FIEND













LAST December we printed an innocent looking little program called Get-It-Right! Since that time the December issue has sold like hot cakes, until we now find that we're completely out of stock.

Since many of you have been writing in asking about Get-It-Right! over the past few months we have decided to reprint it for you, with a couple of extra features.

The purpose of this program is to provide you with an easy way to make certain that you have typed our listings in correctly, and to show you exactly where any errors are located.

Firstly sit down and type in the listing below. If you are using cassette rather than disc you should modify line 1010 as detailed in line 0999 of the listing.

Once you're quite sure that you have got it right, SAVE a copy on to your tape or disc. This will be your master Get-It-Right! program.

Now to check a program you have just typed in from an issue of *Atari User*, use the following procedure:

Firstly SAVE a copy of the program you've just typed in on to tape or disc. This is your secure copy, just in case anything goes wrong.

Now create a listed version of the program by typing:

```
LIST "C"
```

```
LIST "D:filename"
```

Note that the normal filename is

## By André Willey and Richard Vanner

use on disc is D:TEST, but you are free to choose your own.

Load your master Get-It-Right! tape or disc into memory with:

```
LOAD
```

or

```
LOAD "D:GIR.BAS"
```

Type RUN and the main menu will appear. Option 1 will list a checksum to the screen, option 2 to the printer and 3 will quit the program and return you to Basic.

Enter the name of the file which you listed your new program to. Pressing Return on its own will use the standard filename of C: for cassettes, and D:TEST for disc.

Check each of the numbers the computer prints out against the numbers in the Get-It-Right! table in the magazine. Make a note of any lines where the number differs, which will need to be corrected later.

Now reload the saved copy of your program. If the two lists of numbers matched completely you

may RUN your program in the sure knowledge that it is as we printed it.

If not, check the problem lines against the listing, and repeat the above process from the first step.

If you are working on a long program you may need to type:

```
[Control-1]
```

as the screen fills up to pause the display. Press it again to restart.

A few tips: If the error has occurred on a REM or a PRINT Line it may not be fatal, and your program may run OK.

If it occurs on a line with DATA, or machine code characters - those funny little graphics shapes - make sure that you have corrected the problem before you try to run the program.

All that remains for you to do now is type in the listing. For a bit of practice you can even check Get-It-Right! with its own checksum just for good measure. Unless of course you got it wrong... ■



```

00 0000 00000000 00000000 00000000
01 0000 00000000 00000000 00000000
02 0000 00000000 00000000 00000000
03 0000 00000000 00000000 00000000
04 0000 00000000 00000000 00000000
05 0000 00000000 00000000 00000000
06 0000 00000000 00000000 00000000
07 0000 00000000 00000000 00000000
08 0000 00000000 00000000 00000000
09 0000 00000000 00000000 00000000
10 0000 00000000 00000000 00000000
11 0000 00000000 00000000 00000000
12 0000 00000000 00000000 00000000
13 0000 00000000 00000000 00000000
14 0000 00000000 00000000 00000000
15 0000 00000000 00000000 00000000
16 0000 00000000 00000000 00000000
17 0000 00000000 00000000 00000000
18 0000 00000000 00000000 00000000
19 0000 00000000 00000000 00000000
20 0000 00000000 00000000 00000000
21 0000 00000000 00000000 00000000
22 0000 00000000 00000000 00000000
23 0000 00000000 00000000 00000000
24 0000 00000000 00000000 00000000
25 0000 00000000 00000000 00000000
26 0000 00000000 00000000 00000000
27 0000 00000000 00000000 00000000
28 0000 00000000 00000000 00000000
29 0000 00000000 00000000 00000000
30 0000 00000000 00000000 00000000
31 0000 00000000 00000000 00000000
32 0000 00000000 00000000 00000000
33 0000 00000000 00000000 00000000
34 0000 00000000 00000000 00000000
35 0000 00000000 00000000 00000000
36 0000 00000000 00000000 00000000
37 0000 00000000 00000000 00000000
38 0000 00000000 00000000 00000000
39 0000 00000000 00000000 00000000
40 0000 00000000 00000000 00000000
41 0000 00000000 00000000 00000000
42 0000 00000000 00000000 00000000
43 0000 00000000 00000000 00000000
44 0000 00000000 00000000 00000000
45 0000 00000000 00000000 00000000
46 0000 00000000 00000000 00000000
47 0000 00000000 00000000 00000000
48 0000 00000000 00000000 00000000
49 0000 00000000 00000000 00000000
50 0000 00000000 00000000 00000000
51 0000 00000000 00000000 00000000
52 0000 00000000 00000000 00000000
53 0000 00000000 00000000 00000000
54 0000 00000000 00000000 00000000
55 0000 00000000 00000000 00000000
56 0000 00000000 00000000 00000000
57 0000 00000000 00000000 00000000
58 0000 00000000 00000000 00000000
59 0000 00000000 00000000 00000000
60 0000 00000000 00000000 00000000
61 0000 00000000 00000000 00000000
62 0000 00000000 00000000 00000000
63 0000 00000000 00000000 00000000
64 0000 00000000 00000000 00000000
65 0000 00000000 00000000 00000000
66 0000 00000000 00000000 00000000
67 0000 00000000 00000000 00000000
68 0000 00000000 00000000 00000000
69 0000 00000000 00000000 00000000
70 0000 00000000 00000000 00000000
71 0000 00000000 00000000 00000000
72 0000 00000000 00000000 00000000
73 0000 00000000 00000000 00000000
74 0000 00000000 00000000 00000000
75 0000 00000000 00000000 00000000
76 0000 00000000 00000000 00000000
77 0000 00000000 00000000 00000000
78 0000 00000000 00000000 00000000
79 0000 00000000 00000000 00000000
80 0000 00000000 00000000 00000000
81 0000 00000000 00000000 00000000
82 0000 00000000 00000000 00000000
83 0000 00000000 00000000 00000000
84 0000 00000000 00000000 00000000
85 0000 00000000 00000000 00000000
86 0000 00000000 00000000 00000000
87 0000 00000000 00000000 00000000
88 0000 00000000 00000000 00000000
89 0000 00000000 00000000 00000000
90 0000 00000000 00000000 00000000
91 0000 00000000 00000000 00000000
92 0000 00000000 00000000 00000000
93 0000 00000000 00000000 00000000
94 0000 00000000 00000000 00000000
95 0000 00000000 00000000 00000000
96 0000 00000000 00000000 00000000
97 0000 00000000 00000000 00000000
98 0000 00000000 00000000 00000000
99 0000 00000000 00000000 00000000
100 0000 00000000 00000000 00000000

```

```

100 0000 00000000 00000000 00000000
101 0000 00000000 00000000 00000000
102 0000 00000000 00000000 00000000
103 0000 00000000 00000000 00000000
104 0000 00000000 00000000 00000000
105 0000 00000000 00000000 00000000
106 0000 00000000 00000000 00000000
107 0000 00000000 00000000 00000000
108 0000 00000000 00000000 00000000
109 0000 00000000 00000000 00000000
110 0000 00000000 00000000 00000000
111 0000 00000000 00000000 00000000
112 0000 00000000 00000000 00000000
113 0000 00000000 00000000 00000000
114 0000 00000000 00000000 00000000
115 0000 00000000 00000000 00000000
116 0000 00000000 00000000 00000000
117 0000 00000000 00000000 00000000
118 0000 00000000 00000000 00000000
119 0000 00000000 00000000 00000000
120 0000 00000000 00000000 00000000
121 0000 00000000 00000000 00000000
122 0000 00000000 00000000 00000000
123 0000 00000000 00000000 00000000
124 0000 00000000 00000000 00000000
125 0000 00000000 00000000 00000000
126 0000 00000000 00000000 00000000
127 0000 00000000 00000000 00000000
128 0000 00000000 00000000 00000000
129 0000 00000000 00000000 00000000
130 0000 00000000 00000000 00000000
131 0000 00000000 00000000 00000000
132 0000 00000000 00000000 00000000
133 0000 00000000 00000000 00000000
134 0000 00000000 00000000 00000000
135 0000 00000000 00000000 00000000
136 0000 00000000 00000000 00000000
137 0000 00000000 00000000 00000000
138 0000 00000000 00000000 00000000
139 0000 00000000 00000000 00000000
140 0000 00000000 00000000 00000000
141 0000 00000000 00000000 00000000
142 0000 00000000 00000000 00000000
143 0000 00000000 00000000 00000000
144 0000 00000000 00000000 00000000
145 0000 00000000 00000000 00000000
146 0000 00000000 00000000 00000000
147 0000 00000000 00000000 00000000
148 0000 00000000 00000000 00000000
149 0000 00000000 00000000 00000000
150 0000 00000000 00000000 00000000

```

```

150 00000000 0000
151 0000 00000000
152 0000 00000000 0000 0000
153 0000 00000000 0000 0000
154 0000 00000000 0000 0000
155 0000 00000000 0000 0000
156 0000 00000000 0000 0000
157 0000 00000000 0000 0000
158 0000 00000000 0000 0000
159 0000 00000000 0000 0000
160 0000 00000000 0000 0000
161 0000 00000000 0000 0000
162 0000 00000000 0000 0000
163 0000 00000000 0000 0000
164 0000 00000000 0000 0000
165 0000 00000000 0000 0000
166 0000 00000000 0000 0000
167 0000 00000000 0000 0000
168 0000 00000000 0000 0000
169 0000 00000000 0000 0000
170 0000 00000000 0000 0000
171 0000 00000000 0000 0000
172 0000 00000000 0000 0000
173 0000 00000000 0000 0000
174 0000 00000000 0000 0000
175 0000 00000000 0000 0000
176 0000 00000000 0000 0000
177 0000 00000000 0000 0000
178 0000 00000000 0000 0000
179 0000 00000000 0000 0000
180 0000 00000000 0000 0000
181 0000 00000000 0000 0000
182 0000 00000000 0000 0000
183 0000 00000000 0000 0000
184 0000 00000000 0000 0000
185 0000 00000000 0000 0000
186 0000 00000000 0000 0000
187 0000 00000000 0000 0000
188 0000 00000000 0000 0000
189 0000 00000000 0000 0000
190 0000 00000000 0000 0000
191 0000 00000000 0000 0000
192 0000 00000000 0000 0000
193 0000 00000000 0000 0000
194 0000 00000000 0000 0000
195 0000 00000000 0000 0000
196 0000 00000000 0000 0000
197 0000 00000000 0000 0000
198 0000 00000000 0000 0000
199 0000 00000000 0000 0000
200 0000 00000000 0000 0000

```





I HAVE had an Atari 1300E since last May and I am being plagued by a problem that has caused me considerable irritation over the last 17 months.

I wonder if anyone can suggest a method of stopping my machine from printing out signals from radio sets.

It occurs only during tape and disc loading operations and stops its signals from up to 20 miles away. I suspect the problem is somewhere in the computer TV cable but have no idea how to stop it.

I have to disagree with L. Wheelhouse's comments on *Flashback's* Chinese. It is definitely inferior to *Atari II*. The graphics are less detailed, less clear and it is mind-numbingly slow.

The loading screen is good for disc car compare with those of disc games and only makes loading time, if we agree that the faster mode is probably the best feature of the game.

However, *Flashback* must be congratulated for supporting the Atari II—and it's reasonable price too.

If you've bought the game with an enhancement *Flashback* on it. This is the only way to encourage companies to support the Atari.

Can you include memory requirements in reviews as I've missed against games that cost less than 48K? Also with a software boom on the way can you review more software? — **A.W.H. Crawford, Penzance.**

## Cartridge not needed

I AMN just purchased an Atari 800XL for my children and have fallen in love with it once! Mind you, I am so green you wouldn't believe it. But the point of my letter is that I am told that I need Atari's tape CAS 80057 to play certain tapes and to be able to save programs.

Are all the shops I have been into do not have it. Could you help please? — **R.W. Jackson, Sunny.**

The item you describe is actually nothing more than the

# CALLING ALL CARS - ON AN ATARI 130XE

Basic cartridge, as provided with the old 400 and 800 computers.

Your 800XL (and the 1300XL) has the Basic language built inside the casing, and so you don't need the cartridge at all.

When you are asked to "Plug in your Basic cartridge and switch on", all you need to do is simply switch on.

If any program or game requires you to remove the cartridge, just hold down the Option key on the keyboard as you switch on. This will disable the built-in Basic and act as though it wasn't there.

## Database on tape

I OWN an 800XL with a 101D monitor, but after a fruitless search of the shops in my area I have found it impossible to obtain a database/spreadsheet on cassette.

Although I am learning programming I am not good enough to write a program for my needs — I own hundreds of records and would like to be able to catalog them for easy access.

Therefore could you please inform me of any database/spreadsheet and how to get it? — **R. Goss, Richmond, W. Yorkshire.**

Unfortunately you're quite right — there isn't a desired database/spreadsheet available on cassette for the Atari.

Really, these applications need a disc drive to be effective, so cassette handling is by no means slow.

Your best bet would be to buy yourself a second-hand disc drive, and get a copy of *Mini Office II* for the Atari

when it is released shortly.

This features both a database and a spreadsheet, plus a word processor, mailmerge, graphics package and communications software all on one disc. You'll find it the cheapest way to buy these items.

## 600XL upgrade kit

I HAVE just news for 600XL owners who have been looking for a cheap upgrade kit.

A firm in the States has one for \$25 US — shipping included — that will make a 600XL perform exactly like the 800XL.

I recently installed one for a friend of mine in about 20 minutes, and the tape has 27,000 free bytes and can run any of the software I have for my 800XL.

The kit consists of two new 64K chips and three jumper wires, two of which require some simple soldering to install.

The instructions are simple and easy to follow. The firm is happy to ship over here as long as payment is in US dollars — even as an international money order.

The company is: Computer Service Land, 1033 Mt. Street Street, Falls Church, Va. 22045, USA. — **Bob Marlowe, USAFL.**

## XL-XE compared

I AMN thinking of buying either an Atari 800XL or 1300XE.

I am get an 800XL plus 705D disc drive for £730, but

the lowest price I have seen for a 1300E + 705D disc drive is about £280.

I rather like the 1300E's styling, but this isn't the best criterion for choosing a model. I know it has an extra 64K of paged memory, but I have read that this software exploits this.

Does the 1300E have any other different features to the 800XL, and if so are they supported by enhanced software? — **Marcus Green, Manchester.**

The 1300E is the same as the 800XL in most respects, and has all of the features provided by its smaller brother.

It does of course feature twice the memory capacity. This is implemented as four 128 banks of RAM, which can be switched in and out as required by the user.

Various software packages use this extra memory, including the DOS 2.0 command, SuperScript word processor, SuperFile database and Sync-Circ spreadsheet.

Also Basic XL from 688 will allow you to write programs of up to 64k, and provides a further 37% for data, strings, graphics and so on.

It also maintains complete compatibility with standard Atari Basic.

## "Missing" memory

WHenever I type PRINT P#(0) on my 800XL I get the answer: 37800.

I know that the address built into the computer uses game memory, but surely not 255.

It is possible that I could have destroyed a chunk of RAM by inadvertently writing on the computer's memory in the wrong sequence?

If so, surely I would not have been the first novice programmer to do so. — **Patrick E. George, Rotherham, South Yorkshire.**

Don't worry, you haven't damaged your computer — this figure is perfectly normal.

Just be thankful you aren't using a 130XE, as you'd have had 90% of owners, by your calculations, if that were the case.

Although the 8000, has 64k of memory the 6802 chip that it uses can only access a total of 64k - including all of the operating system in ROM. Basic itself, your programs and so on.

The CG is 10k long, and Basic is 8k. The screen is about 1k, and various special areas for input and output take up another 1k or so. Thus you have just "used" 28k.

The area under Basic and the CG is still there as RAM, but you can only get at it by turning one or the other off, and then sending your Basic program into save+reset+end.

If you really want to access this extra RAM you must have a small machine code program in command when you turn the CG off in order to stop the system from crashing.

## Ramdisc on 130XE

I WISH to use the ramdisc facility on my 130XE offered by DOS 2.5 when using MacDisk. Which also fits Atari? - **Tim Brian Holt, Camberley, Surrey.**

Unfortunately MacDisk comes on a protected disc with Atari DOS 2.0 provided.

It would be quite hard to reformat the disc to use DOS 2.5 as, without any detailed technical programming experience, you'd probably be best sticking with MacDisk as it is.

## Four line listings

I read in the manual for my 8000E that no listing should exceed three lines, but in your December issue the *Pyramid* game had three listings of four lines.

So we made each listing in two lines instead of four. The game worked all right but it is a bit funny on the writing.

In the January issue you

# ATARI USER Mailbag

We welcome letters from readers - about your experiences using the Atari models, about tips you would like to pass on to other users... and about what you would like to see in future issues.

The address to write to is:

**Mailbag Editor  
Atari User  
Europe House  
88 Chester Road  
Hazel Grove  
Stockport SK7 5RY**

have a listing in *Mr Pumpy* with four lines. My old the same and split the line in two. The game works quite well but I wonder if it is all right.

I have read the Atari Handbook and it says you can write listings of four lines or more but does not say exactly how.

It just says to play about with the C, S or Tab key. Could you explain how to write these lines? - **M. Meadows, Bicester, Oxford.**

The computer can only accept a maximum of three screen lines of input for any given program line and there is a limit to how to get around this.

However there's nothing to say that you can't try and pack as much into these three lines as possible.

Firstly, before you start typing in your program type:

**POKE #0**

This will set the margin to 0, so giving you back the two empty spaces at the start of each line.

If that still doesn't give you enough room to enter a given line use abbreviations. Thus if a line of a listing reads:

**IF @=0 THEN GOTO L1:IN  
RETURN L1: END "HELLO"**

try typing in:

**IF @=0 THEN GOTO L1:IN**

The computer will understand the two as meaning the same thing and when you type LIST you will see the full version, not the abbreviations.

Consult your manual for the

correct abbreviations for all of the Basic commands - most are the first few letters followed by a dot.

By using both of these methods there are no lines of program that you should ever have any problem with.

Don't forget someone had to enter them into the same type of computer in the first place, so they must go in.

## Better disc labels

YOU may have noticed that it is very difficult to see floppy disc labels and write-protect tabs separate from a box of floppies.

In searching for a suitable alternative I got a packet of self-adhesive coloured labels from a well known High Street stationer.

These labels seemed to be even more useful than standard write-protect tabs since they aren't double as a colour-coding system.

Usually I used the red labels to protect discs that I must not write on under any circumstances. However that proved my downfall.

The write-protect circuitry in the Atari disc drive uses infra red light to sense the presence or absence of a write-enable notch, and infra red goes straight through these labels.

I formatted two discs containing valuable programs before I realised what was happening. This type of write-

protection is worse than none at all since it implies a false sense of security.

The moral of the story is keep trying to find the correct labels, or if you must use coloured labels use any colour other than red - blue or green would be best - **Jolly Rogers, Wincersh, Berks.**

## Using checksum

COULD you tell me how to use your checksum. I have tried to follow your instructions in the March issue of Atari User, but have had no success.

I do not understand your statement, "Now Just Use It Right into the machine, taking note of the instructions contained in the ROM in the 8000".

I have typed in all your games since I got my 8000E in January 1985, and only two have worked. It is so disappointing to all for several hours typing in and then find nothing happening. - **David Brunton, East Lothian.**

\*\*\*

COULD you please repair your checksum as I have lost my copy of your corrected version. - **Jon, Dale, Manchester.**

\*\*\*

I BOUGHT your December 1985 issue of Atari User, and I also bought the tape for that month so I would use the checksum.

However when I tried to follow your instructions on how to load the checksum and then type RUN the program will not work. Please could you print the instructions more clearly? - **T. Macklin, Gwent, South Wales.**

That was in our command. In response to many similar letters pleading for us to reprint the checksum on the original issue is now out of print we have done so this month.

We've even added a couple of extra features. So now you don't need to write in with problems with our listings - you can all find out for yourself where your typing mistakes are.

## Silencing disc din

I HAVE an 80001, with disc drive and was plagued with a terrible disc-scratching sound the whole time I used the drive.

Not knowing any better I thought this was normal, but after talking with John Lawson at Computer Support I found out there is a simple way to silence the drive.

Just afterwards all wires to the drive, turn it upside down, under the screen and turn the drive back the right way up.

Then take off the lid, apply a very small drop of silicone-like fluid (runners - being very careful not to get any on or near the head) - and finally just put it off back together. — **G.J. Rowler, Hawthorn, Vic.**

## Paint dump program

I AM the proud owner of an Atari 8000X, disc drive, cassette recorder and 1024 printer.

The keyboard and disc drive came in a clear pack, which I received. The Home File Manager, The Revoff and Print.

When I added a printer to my system I chose the Atari 1024, this came with AtariWriter and Dump1024.

When I loaded up Dump1024 I discovered to my dismay that the program would only dump 63 screen picture files.

Rather than write my own dump program I decided to write a modifier program to run Print files into 63 screen picture files.

I worked on this in Mode 7 - which Print works in - it took two bits to define each pixel as there are four colours in Mode 7 and there are also four different combinations of two bits - 00, 01, 10 and 11.

Because of this it takes 40 bytes for each screen line in Mode 7.

Getting back to the problem, Dump1024 would only output pictures drawn in Mode 8 or 14. So to convert

Print files to Mode 14 files will need to do two calculations each 40 bytes.

Not quite so. Print does not use all of a Mode 7 screen, so extra bytes must be added to file and to make the total number of bytes up to 7550 (7500/40).

I have written the following self-explanatory program to demonstrate this problem. — **John Young, Bankbury, Oxon.**

```

1000  REM ***** PRINT FILES INTO MODE 14 *****
1010  REM *****
1020  REM *****
1030  REM *****
1040  REM *****
1050  REM *****
1060  REM *****
1070  REM *****
1080  REM *****
1090  REM *****
1100  REM *****
1110  REM *****
1120  REM *****
1130  REM *****
1140  REM *****
1150  REM *****
1160  REM *****
1170  REM *****
1180  REM *****
1190  REM *****
1200  REM *****
1210  REM *****
1220  REM *****
1230  REM *****
1240  REM *****
1250  REM *****
1260  REM *****
1270  REM *****
1280  REM *****
1290  REM *****
1300  REM *****
1310  REM *****
1320  REM *****
1330  REM *****
1340  REM *****
1350  REM *****
1360  REM *****
1370  REM *****
1380  REM *****
1390  REM *****
1400  REM *****
1410  REM *****
1420  REM *****
1430  REM *****
1440  REM *****
1450  REM *****
1460  REM *****
1470  REM *****
1480  REM *****
1490  REM *****
1500  REM *****
1510  REM *****
1520  REM *****
1530  REM *****
1540  REM *****
1550  REM *****
1560  REM *****
1570  REM *****
1580  REM *****
1590  REM *****
1600  REM *****
1610  REM *****
1620  REM *****
1630  REM *****
1640  REM *****
1650  REM *****
1660  REM *****
1670  REM *****
1680  REM *****
1690  REM *****
1700  REM *****
1710  REM *****
1720  REM *****
1730  REM *****
1740  REM *****
1750  REM *****
1760  REM *****
1770  REM *****
1780  REM *****
1790  REM *****
1800  REM *****
1810  REM *****
1820  REM *****
1830  REM *****
1840  REM *****
1850  REM *****
1860  REM *****
1870  REM *****
1880  REM *****
1890  REM *****
1900  REM *****
1910  REM *****
1920  REM *****
1930  REM *****
1940  REM *****
1950  REM *****
1960  REM *****
1970  REM *****
1980  REM *****
1990  REM *****
2000  REM *****

```

■ We're glad you liked the 1024 dump program John, and many thanks for the tip.

We'd also like to thank other readers who like the extra facility of printing Print files as well as AtariPrinter and Touch Tablet printers.

## The dump that doesn't

COULD you publish a 1024 screen dump because the one you printed did not work.

Also why have you not printed a review on the Diskos, I've heard that it's very good I would like to hear what you think about it before I buy. — **S. Street.**

■ Have you thoroughly checked your typing to make sure that all of the DATA statements are correct?

We can assure you that the program does work as printed - as you can see from John

Young's letter above (he has managed to get it working with no problems).

## On the Yukon trail

My reply to Robert and Steve Swinson's letter in the July 1985 Mailbag to find Yukon Yehon you must fill up the Hi-score table.

Once it is full, play the game once more so that your score is bigger than the last score on the Hi-score table.

Enter your name and the lowest score will be brought down off the table and your best score will go up. Then two full-screen will show the abandoned lowest score and Yukon Yehon will arrive on a jet, walk over to the abandoned score and blow it off the screen.

Yukon can be either a motor or gun-driven. I hope that interests your generation. — **Frank Driver, West Town, Peterborough.**

■ Matthew Barnoughs, of Ormskirk, also provided this solution. Our thanks to you both.

## Store data in ramdisc

I WOULD like to know if the Atari 130AE (a 64K or 128K computer as I am writing an adventure which is likely to take up more than 64K of memory).

The standard said that it got 131,002 bytes - 128K - of RAM like the following:

POKE 54017, 193+4\*  
address = 18\*Private

and then press RETURN, it came up with Ready so I then typed:

PRINT PEEK (0)

and it came up with 37902 bytes free which is not 128K of RAM.

Can you help me? — **Richard Powell, Woodhouse, Sheffield.**

■ The 130AE is indeed a 128K computer, but you must remember that Atari Basic only supports 37902 bytes of memory after the OS and graphics overheads have been

removed from the main 64K area. The four extra 16K banks can be accessed as you described.

Perhaps better still you could use DOS 2.0 to store your data in to the ramdisc, and read it back again as a file each time you need it. The ramdisc is very fast, so you shouldn't notice any appreciable delay.

## Flight simulator

ON your May 1985 Mailbag there was a letter from Alan Ford who was interested in flight simulators.

Has he had a look at Digital Integration's rather good Fighter Pilot?

Could you help me find some books on Atari. We are in my area about there, so where do I go? — **R. Hasleigh, Huddersfield.**

■ Thanks for the advice on the flight simulators, of which there are now quite a number, including Jump Jet and Flight Simulator II.

As to your question on books, if you can't find anything locally why not try one of the larger mail order companies?

Both Software Express and Silas West should be able to help with most types of Atari book, and also give you advice on which titles would best suit your needs.

This would be far better than ordering blind from W.H. Smith, for example.

## Monitor mismatch

I OWN an Atari 8000X, with a 7080 disc drive and 1024 printer.

A few weeks ago I was given an Amstrad CP1608 monitor, but I found that the 8 pin DIN wouldn't match with the 9 pin DIN socket of my Atari.

I have tried to find information on how to connect the monitor to my computer, but

today seems to show how  
Can you please help? — **R.  
Zavala, Lower Merion, Balti-  
more.**

■ The Amstrad uses a system called RGB to connect with its monitor, and the 8000 uses a completely different method known as composite video.

Unfortunately there is no easy way to get the Atari to talk to the Amstrad monitor. You will need to get a composite monitor rather than an RGB one.

## Cleaner graphics

HOW do the professional draw graphics for games so it does not leave a trail and doesn't flash?

Can you give me a short program on how to draw moving graphics, and can you also tell me how to combine two graphics routines? — **Markus House, Hockley, Essex.**

■ Although your question is simple enough the answer is very complicated indeed. To make a game look as professional as a purchased one you will need to write it in machine code.

That said, you will get some good hints and tips on how to get your games working a bit faster if you have been following our series on Player/Missile Graphics which started in the May issue.

We covered display lists — which allow you to combine different graphic modes on a single screen — in Mike Rowe's series of articles which ran from July to December 1985.

## Simpler scrolling

ANSWERING Mike's question in your May 1986 issue is Alan's rather than scrolling was somewhat longwinded and I'd like to suggest some simpler methods of handling this problem.

I think it's even mentioned in the inadequate Atari Users Manual that by holding down the Control key and pressing the P key the list will be halted,

HOW do I run the Cavern Escape program from the May 1986 Atari User? I keep getting Error at line 5542 and 5550. Could you tell me why this is so? — **T.H. Mann, Nottingham.**

\*\*\*

I AM having trouble with Mike Rowe's Cavern Escape.

First I have programmed the game and before I have got Error 8 at line 5245.

After 2½ hours of using the keyboard my patience was wearing a thin skin.

The game does run, but so far my son has only been able to find one key in Room 9.

Where have I gone wrong and do I have to retype the full program? — **A. Mackay, Hull.**

\*\*\*

I TYPED in Cavern Escape from your May issue, and then CSW4011 is on its objects.

However when I see the program the screen went blank as was stated, but if I don't press reset it would have stayed blank forever.

After having pressed reset and typing RUN all I get is Error 8 at line 5500.

I split the 5500 into four lines to find where the error

and repeating the action will subsequently restart it.

As far as display screen output, and will work with any form of output to the screen — even graphics.

When you've found the action you wish the Break key may be employed as normal to return control to the keyboard.

Another more elegant solution is to type in the following line as a direct command:

POKE \$22,255:GR. 0

This tells the Atari to reinitiate its display list employing the line scroll capability, and must be reinitiated after each reset.

Not only does it show the listing of programs in a much more readable speed, but it also looks very impressive. — **Tony Barber, Australia.**

# Trouble in that Cavern

was and it showed up a READ A.

As a newcomer to the Atari 8000X I am baffled by this and would be grateful if you could tell me where I have gone wrong.

I have checked my listing line for line and it is exactly as is printed in Atari User.

One more query I have is that after line 5000 there is a line starting:

5000 READ

Should this be 5000 or is it a misprint? — **Alan Symms, Southwold, S. Devon.**

\*\*\*

I AM having trouble getting Cavern Escape to work. I typed it in, corrected all errors using the checksum and sat down to enjoy the game.

The screen went blank after I typed in RUN as it's supposed to be the trouble is that's all that happened.

I have checked the programs checksum several times now, and I find there is an odd discrepancy in the listing — page 20 — the following line appears between lines 5000 and 5010:

5000 READ

I included this line in my

listing, but references in your checksum the line is checked in the same order it is listed in my checksum it appears in its correct numeric place. Please could you explain this?

Finally one small point is that in line 7790 the second pair of hash marks is not very clear. Could you confirm that it is in fact a second one? — **Nigel Edwards, Lighton Buzzard, Beds.**

■ The extra line 5000 crept in to the listing by mistake, and lots of people have pointed it out to us.

However it doesn't make any difference — the program will run correctly with or without it — phew!

The three sets of characters in line 1190 are two inverse hash symbols (#), then an inverse hash and an inverse dollar sign, and finally two more inverse hashes.

If you had used the checksum and got it wrong, the numbers would not have matched, so we can only assume that if you did use it that it might you must have typed in the listing correctly.

Anyway — the answer to all of your problems is — use the checksum. You don't have any excuse now, as the full instructions have been reprinted on page 33.

■ This tip is a useful one, but only if you have an XL or XE. The older 400 and 800 computers do not have this feature, which is why we printed Mr Hill's solution.

As you say, for XL and XE owners POKE \$22,255 is a very elegant answer to this problem.

## Antic chip playing up?

UNFORTUNATELY it is not well with my 8000X. The graphics otherwise on certain commercial programs appear incorrectly.

For instance, on AtariSoft's Pole Position the tyres on the racing car begin flashing

through software whenever other cars overtake.

On US Gold's Dromedary — an excellent about ten up game — the Defender-style radar screen occupies a part of the screen reserved for user data.

Also the scrolling sprite colours in the Hall of Fame do not work properly.

On English Software's Flightfighter there appear to be three paths that the first tunnel when there should only be one.

I know these errors are not caused by faulty software or head-alignment problems with my data recorder because I have tried using a friend's recorder to find the same errors, and they do not appear on his 8000X.

My theory is that there is a



More  
**£25**  
winners!

## ELECTRONIC TYPEWRITER

From G. D. MASSEY

Of all the Five-liners that we have received, this one is probably the most straightforward. It enables your Atari keyboard to act like an electronic typewriter using its normal printer.

The screen acts just like the old memo pad made on the 400 and 800 computers, except for the fact that control codes will go straight to the printer, rather than doing their normal function on the screen. Thus you can turn printer options on and off by sending the special codes. Some people may also wish to add a POKE 762,1 to remove the cursor.

The program was written in Class Five at Broadclyst Primary School - so Class Five, thanks for your Five-liner!

```

10 POKE 100,1000:PRINT "G.D.M."
20 WHILE 80,100,100
30 PRINT "G.D.M."
40 FOR I=0 TO 1:STEP 0.001:GOTO 50
50 FOR I=0.001 TO 0.000:STEP -.001

```

✓ *Get it right!*

10 1000 20 1000 30 1000  
40 1000 50 1000 60 1000

## TWO FOR THE PRICE OF ONE

From RON SMITH

SOONER or later the Atari programmer needs to solve two problems - disabling the Break key and disabling the Attract Mode.

Most solutions for the Break key problem suffer from the drawback of having to repeatedly execute code within the Basic program after each I/O operation or change of graphics mode.

Presented here is a once-and-for-all five liner that solves the problem for most games and applications and throws in an answer to the Attract Mode problem for good measure. These should form the first few lines of your program for earliest protection.

The program uses just 24 bytes of machine code, tucked away at the top of page 8, leaving the rest clear for data or other routines.

The Atari displays a television screen once every 80th of a second, but has time between screens (the Vertical Blank) to leave a Basic program in order to carry out its own housekeeping.

Two user-patchable locations are provided during this time where the programmer may point the computer to a short machine code subroutine of his own. This routine uses the first of those opportunities, and acts up an immediate mode (as opposed to the second, deferred mode). Vertical Blank Interrupt (VBI) which resets IRQEN, the Interrupt Request Enable byte at \$C7F4 (\$D3FE) and ATTRACT, the Attract Mode driver and flag at 77 (\$40).

The great advantage of this sledgehammer approach is that, once

started, the routine is executed automatically 50 times a second, at no cost to the Basic program. The approach is AI compatible, SPEAK vector notwithstanding.

The program is constructed as follows:

Line 10 is a belt-and-braces step, setting up IRQEN and its shadow at 16 (\$10) while the main routine is loading.

Line 20 sets up the immediate mode VBI and points the Operating System at our routine sitting at 1376, (\$D4F0).

Line 30 activates the VBI and our routine becomes operational within 1/50th second.

Line 40 contains the VBI initialization and pointer data.

Line 50 contains the instructions which set IRQEN to 112 and ATTRACT to zero, plus the jump that takes us back into the operating system's tasks, before rejoining the Basic program.

```

10 POKE 16,127:POKE 1676,120
20 FOR I=16 TO 170:POKE I,0:GOTO 40
30 GOTO 50
40 POKE 112,1:POKE 112,0:POKE 5,70:GOTO 50
50 POKE 1376,120:POKE 1376,1376:GOTO 1376

```

✓ *Get it right!*

10 1200 20 1200 30 1200  
40 1200 50 1200 60 1200

**Original,  
useful -  
or just  
plain fun!**

If you'd like to have a go at writing some Five-liners of your own - and we pay £25 for every one we print - then we'd like to hear from you.

Try to make them original, useful or entertaining. After we'd received the first couple of dozen programs to add your name on the screen, or play a musical scale, we began to get impatient for something more stimulating. Let's see you use your imagination! Also, please remember that

you're far more likely to get one published if you send us a neat listing, with good explanatory notes and a copy of the program on tape or disc. Our staff haven't got the time to decipher little scraps of assembler code in illegible listings. If we did, we'd never actually complete the magazine.

Send your Five-liners to us at: Five-liners, Atari User, Europe House, 48 Chester Road, Hazel Grove, Stockport SK7 5NF.



**THIS program uses graphics mode 8 to produce an 80 column text display which can be combined with mode 8 graphics to produce professional looking displays in your own programs.**

Before running this program, **SAVE** a copy to disc or tape. This is very important because the program will rewrite itself then delete a large part of itself. This is to save you a lot of hard work trying to type the control codes in the program correctly.

The program will read the data statements and then write new program lines containing the string definitions for the machine code routine and the character set data. These strings are almost unreadable, containing approximately 400 Escape and Control key characters.

After writing these lines, the program will delete all the data statements and most of the program elements involved in creating these strings, only leaving three lines not needed for the program. These are lines 1235, 1240 and 1250, and they should be deleted. They cannot be deleted by the program because they are required by the delete routine! If you do not **SAVE** a copy before running, and you have made any typing errors, you will have to start all over again.

When you first run this program the screen will go black and the data statements will be counted down on it. When the counter reaches zero the screen will return to the normal graphics 0 display and you will see the strings being printed out and the cursor moves down over the screen text.

This is followed by a few screens of line numbers, which is the line deletion process taking place. When complete the program will return you to direct mode BASIC with the familiar **READY** prompt. At this point the three lines mentioned earlier should be deleted.

Now **SAVE** the new program under a different filename to the original. It is this version of the program that you will use to produce your 80 column displays.

Some of you may be wondering how your Atari can start writing new parts of the program and deleting lines all by itself. This is because your Atari is braver than the average computer.

Actually it is very easy to persuade

# 80 column text plus graphics mode 8 equals the pro look

By R. and A. LINES

your Atari to do the work for you - this can be achieved with one simple **POKE**. Using the **POKE** puts the computer in to 'BASIC Key mode'. Poking 842 with 13 puts the computer into screen input/output mode. Poking 842 with 12 returns the computer to keyboard input mode.

This means that anything on the screen and below the cursor when this poke is performed will be treated as a direct mode command. Anything you require to be done is printed on the screen and the cursor is positioned above the commands. Then you **POKE** 842,13:STOP.

You must also have a **CONT** statement after the last command

you want executed, which will continue program execution at the next line after the **STOP** command. This next command should be **POKE** 842,12. For an example of how to do this see line numbers 820 to 900.

The machine code routine is called by the **USR** function:

```
Z=USR(ADR(POKE),  
ADR(CHAR),X,Y,ADR(CHR))
```

The first parameter is the address of the string that contains the 80 column program and the second is the address of the string containing all the data for the redefinition of the character set.

The third parameter is the X coordinate of the starting position of

*W* enter to show you how this routine works we have printed three demonstration programs. To see them you should first create your main program by following the instructions in the text.

After the working session has been created, delete lines 1235, 1240 and 1250 and **SAVE** it to disc or tape. This is now your master program.

For each of the three demo programs type **NEW** and then type in the lines. You must now **LIST** - not **SAVE** - a copy to your tape or

disk, and you may then check it if you desire.

To run them, first **LOAD** the master program, then **ENTER** the first demo to change the two files. You could **SAVE** a copy of the merged program at this stage if you so desire. After you have **RUN** it you may **ENTER** the next demo, and after that the third.

Demo 1 shows an 80 column table of computer facts. Demo 2 shows what an adventure game might look like in 80 columns, and Demo 3 shows graphics and 80 column text on a chart.



the line of text to be printed. It can have a value between 1 and 80, but it should be remembered that text can only be printed on the same line—that is, it will not continue on to the next line.

The fourth parameter is the Y coordinate of the starting position of the line of text to be printed. It can have a value between 1 and 185.

The fifth parameter is the address of the string that contains the line of text to be printed, in this case CHR.

The USR function is provided as line 300 in the program, although of course you may place a USR call wherever you desire one. The one in line 300 is set up to be used as a substitute with the demonstrations provided.

Each line of text must be terminated by an @ because the machine code routine uses this character to determine when to finish printing.

The normal keyboard characters can be obtained, and in addition a pound sign—£—can be obtained by using Control-follow.

Normal 80 column characters are

drawn on an 8 x 8 matrix, for instance the letter A is drawn as follows:

Character	Binary	Decimal
.....	00000000	0
..@..@..	00010000	16
..@..@..	00110000	32
..@..@..	01010000	48
..@..@..	01110000	64
..@..@..	10010000	80
..@..@..	10110000	96
..@..@..	11010000	112
..@..@..	11110000	128
.....	00000000	0

As we are using 80 characters per line each character can only be half so wide, that is it is a 4 x 8 matrix,

therefore the character data is stored in pairs of characters. For example, B and C are stored as shown below.

Character	Binary	Decimal
..B..C..	00001000	8
..B..C..	00101000	16
..B..C..	01001000	24
..B..C..	01101000	32
..B..C..	10001000	40
..B..C..	10101000	48
..B..C..	11001000	56
..B..C..	11101000	64
.....	00000000	0

**VARIABLES**

- CHR# Character set data.
- CHR# Text to be printed.
- PRN# Machine code routine.
- NUM# Counter for data statements.
- FOR# Row counter.
- LE# Row variable.
- C# Column variable for USR function.
- R# Row variable.
- F# Row variable.
- A# Offset for line delays routine (C#-184).

#### Main program

```

100 GOSUB 1000000,1000000,1000000,1000000
110 PRINT#0 "*****"
120 PRINT#0 "A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
@,
[ in row 10 "
130 PRINT#0 "FOR I=1 TO 26
140 FOR J=1 TO 26:PRINT#0 "CHR$(I+J-1)";
150 NEXT J
160 NEXT I
170 PRINT#0 "FOR I=1 TO 26:PRINT#0 "
CHR$(I);
FOR J=1 TO 26:PRINT#0 " ";
NEXT J
180 GOTO 1000000
190 GOTO 1000000
200 GOTO 1000000
210 GOTO 1000000
220 GOTO 1000000
230 GOTO 1000000
240 GOTO 1000000
250 GOTO 1000000
260 GOTO 1000000
270 GOTO 1000000
280 GOTO 1000000
290 GOTO 1000000
300 GOTO 1000000
310 GOTO 1000000
320 GOTO 1000000
330 GOTO 1000000
340 GOTO 1000000
350 GOTO 1000000
360 GOTO 1000000
370 GOTO 1000000
380 GOTO 1000000
390 GOTO 1000000
400 GOTO 1000000
410 GOTO 1000000
420 GOTO 1000000
430 GOTO 1000000
440 GOTO 1000000
450 GOTO 1000000
460 GOTO 1000000
470 GOTO 1000000
480 GOTO 1000000
490 GOTO 1000000
500 GOTO 1000000
510 GOTO 1000000
520 GOTO 1000000
530 GOTO 1000000
540 GOTO 1000000
550 GOTO 1000000
560 GOTO 1000000
570 GOTO 1000000
580 GOTO 1000000
590 GOTO 1000000
600 GOTO 1000000
610 GOTO 1000000
620 GOTO 1000000
630 GOTO 1000000
640 GOTO 1000000
650 GOTO 1000000
660 GOTO 1000000
670 GOTO 1000000
680 GOTO 1000000
690 GOTO 1000000
700 GOTO 1000000
710 GOTO 1000000
720 GOTO 1000000
730 GOTO 1000000
740 GOTO 1000000
750 GOTO 1000000
760 GOTO 1000000
770 GOTO 1000000
780 GOTO 1000000
790 GOTO 1000000
800 GOTO 1000000
810 GOTO 1000000
820 GOTO 1000000
830 GOTO 1000000
840 GOTO 1000000
850 GOTO 1000000
860 GOTO 1000000
870 GOTO 1000000
880 GOTO 1000000
890 GOTO 1000000
900 GOTO 1000000
910 GOTO 1000000
920 GOTO 1000000
930 GOTO 1000000
940 GOTO 1000000
950 GOTO 1000000
960 GOTO 1000000
970 GOTO 1000000
980 GOTO 1000000
990 GOTO 1000000
1000 GOTO 1000000

```

```

100 GOTO 100,100,100,100,100,100,100,100
101 GOTO 100,100,100,100,100,100,100,100
102 GOTO 100,100,100,100,100,100,100,100
103 GOTO 100,100,100,100,100,100,100,100
104 GOTO 100,100,100,100,100,100,100,100
105 GOTO 100,100,100,100,100,100,100,100
106 GOTO 100,100,100,100,100,100,100,100
107 GOTO 100,100,100,100,100,100,100,100
108 GOTO 100,100,100,100,100,100,100,100
109 GOTO 100,100,100,100,100,100,100,100
110 GOTO 100,100,100,100,100,100,100,100
111 GOTO 100,100,100,100,100,100,100,100
112 GOTO 100,100,100,100,100,100,100,100
113 GOTO 100,100,100,100,100,100,100,100
114 GOTO 100,100,100,100,100,100,100,100
115 GOTO 100,100,100,100,100,100,100,100
116 GOTO 100,100,100,100,100,100,100,100
117 GOTO 100,100,100,100,100,100,100,100
118 GOTO 100,100,100,100,100,100,100,100
119 GOTO 100,100,100,100,100,100,100,100
120 GOTO 100,100,100,100,100,100,100,100
121 GOTO 100,100,100,100,100,100,100,100
122 GOTO 100,100,100,100,100,100,100,100
123 GOTO 100,100,100,100,100,100,100,100
124 GOTO 100,100,100,100,100,100,100,100
125 GOTO 100,100,100,100,100,100,100,100
126 GOTO 100,100,100,100,100,100,100,100
127 GOTO 100,100,100,100,100,100,100,100
128 GOTO 100,100,100,100,100,100,100,100
129 GOTO 100,100,100,100,100,100,100,100
130 GOTO 100,100,100,100,100,100,100,100
131 GOTO 100,100,100,100,100,100,100,100
132 GOTO 100,100,100,100,100,100,100,100
133 GOTO 100,100,100,100,100,100,100,100
134 GOTO 100,100,100,100,100,100,100,100
135 GOTO 100,100,100,100,100,100,100,100
136 GOTO 100,100,100,100,100,100,100,100
137 GOTO 100,100,100,100,100,100,100,100
138 GOTO 100,100,100,100,100,100,100,100
139 GOTO 100,100,100,100,100,100,100,100
140 GOTO 100,100,100,100,100,100,100,100
141 GOTO 100,100,100,100,100,100,100,100
142 GOTO 100,100,100,100,100,100,100,100
143 GOTO 100,100,100,100,100,100,100,100
144 GOTO 100,100,100,100,100,100,100,100
145 GOTO 100,100,100,100,100,100,100,100
146 GOTO 100,100,100,100,100,100,100,100
147 GOTO 100,100,100,100,100,100,100,100
148 GOTO 100,100,100,100,100,100,100,100
149 GOTO 100,100,100,100,100,100,100,100
150 GOTO 100,100,100,100,100,100,100,100
151 GOTO 100,100,100,100,100,100,100,100
152 GOTO 100,100,100,100,100,100,100,100
153 GOTO 100,100,100,100,100,100,100,100
154 GOTO 100,100,100,100,100,100,100,100
155 GOTO 100,100,100,100,100,100,100,100
156 GOTO 100,100,100,100,100,100,100,100
157 GOTO 100,100,100,100,100,100,100,100
158 GOTO 100,100,100,100,100,100,100,100
159 GOTO 100,100,100,100,100,100,100,100
160 GOTO 100,100,100,100,100,100,100,100
161 GOTO 100,100,100,100,100,100,100,100
162 GOTO 100,100,100,100,100,100,100,100
163 GOTO 100,100,100,100,100,100,100,100
164 GOTO 100,100,100,100,100,100,100,100
165 GOTO 100,100,100,100,100,100,100,100
166 GOTO 100,100,100,100,100,100,100,100
167 GOTO 100,100,100,100,100,100,100,100
168 GOTO 100,100,100,100,100,100,100,100
169 GOTO 100,100,100,100,100,100,100,100
170 GOTO 100,100,100,100,100,100,100,100
171 GOTO 100,100,100,100,100,100,100,100
172 GOTO 100,100,100,100,100,100,100,100
173 GOTO 100,100,100,100,100,100,100,100
174 GOTO 100,100,100,100,100,100,100,100
175 GOTO 100,100,100,100,100,100,100,100
176 GOTO 100,100,100,100,100,100,100,100
177 GOTO 100,100,100,100,100,100,100,100
178 GOTO 100,100,100,100,100,100,100,100
179 GOTO 100,100,100,100,100,100,100,100
180 GOTO 100,100,100,100,100,100,100,100
181 GOTO 100,100,100,100,100,100,100,100
182 GOTO 100,100,100,100,100,100,100,100
183 GOTO 100,100,100,100,100,100,100,100
184 GOTO 100,100,100,100,100,100,100,100
185 GOTO 100,100,100,100,100,100,100,100
186 GOTO 100,100,100,100,100,100,100,100
187 GOTO 100,100,100,100,100,100,100,100
188 GOTO 100,100,100,100,100,100,100,100
189 GOTO 100,100,100,100,100,100,100,100
190 GOTO 100,100,100,100,100,100,100,100
191 GOTO 100,100,100,100,100,100,100,100
192 GOTO 100,100,100,100,100,100,100,100
193 GOTO 100,100,100,100,100,100,100,100
194 GOTO 100,100,100,100,100,100,100,100
195 GOTO 100,100,100,100,100,100,100,100
196 GOTO 100,100,100,100,100,100,100,100
197 GOTO 100,100,100,100,100,100,100,100
198 GOTO 100,100,100,100,100,100,100,100
199 GOTO 100,100,100,100,100,100,100,100
200 GOTO 100,100,100,100,100,100,100,100

```

```

201 GOTO 100,100,100,100,100,100,100,100
202 GOTO 100,100,100,100,100,100,100,100
203 GOTO 100,100,100,100,100,100,100,100
204 GOTO 100,100,100,100,100,100,100,100
205 GOTO 100,100,100,100,100,100,100,100
206 GOTO 100,100,100,100,100,100,100,100
207 GOTO 100,100,100,100,100,100,100,100
208 GOTO 100,100,100,100,100,100,100,100
209 GOTO 100,100,100,100,100,100,100,100
210 GOTO 100,100,100,100,100,100,100,100
211 GOTO 100,100,100,100,100,100,100,100
212 GOTO 100,100,100,100,100,100,100,100
213 GOTO 100,100,100,100,100,100,100,100
214 GOTO 100,100,100,100,100,100,100,100
215 GOTO 100,100,100,100,100,100,100,100
216 GOTO 100,100,100,100,100,100,100,100
217 GOTO 100,100,100,100,100,100,100,100
218 GOTO 100,100,100,100,100,100,100,100
219 GOTO 100,100,100,100,100,100,100,100
220 GOTO 100,100,100,100,100,100,100,100
221 GOTO 100,100,100,100,100,100,100,100
222 GOTO 100,100,100,100,100,100,100,100
223 GOTO 100,100,100,100,100,100,100,100
224 GOTO 100,100,100,100,100,100,100,100
225 GOTO 100,100,100,100,100,100,100,100
226 GOTO 100,100,100,100,100,100,100,100
227 GOTO 100,100,100,100,100,100,100,100
228 GOTO 100,100,100,100,100,100,100,100
229 GOTO 100,100,100,100,100,100,100,100
230 GOTO 100,100,100,100,100,100,100,100
231 GOTO 100,100,100,100,100,100,100,100
232 GOTO 100,100,100,100,100,100,100,100
233 GOTO 100,100,100,100,100,100,100,100
234 GOTO 100,100,100,100,100,100,100,100
235 GOTO 100,100,100,100,100,100,100,100
236 GOTO 100,100,100,100,100,100,100,100
237 GOTO 100,100,100,100,100,100,100,100
238 GOTO 100,100,100,100,100,100,100,100
239 GOTO 100,100,100,100,100,100,100,100
240 GOTO 100,100,100,100,100,100,100,100
241 GOTO 100,100,100,100,100,100,100,100
242 GOTO 100,100,100,100,100,100,100,100
243 GOTO 100,100,100,100,100,100,100,100
244 GOTO 100,100,100,100,100,100,100,100
245 GOTO 100,100,100,100,100,100,100,100
246 GOTO 100,100,100,100,100,100,100,100
247 GOTO 100,100,100,100,100,100,100,100
248 GOTO 100,100,100,100,100,100,100,100
249 GOTO 100,100,100,100,100,100,100,100
250 GOTO 100,100,100,100,100,100,100,100
251 GOTO 100,100,100,100,100,100,100,100
252 GOTO 100,100,100,100,100,100,100,100
253 GOTO 100,100,100,100,100,100,100,100
254 GOTO 100,100,100,100,100,100,100,100
255 GOTO 100,100,100,100,100,100,100,100
256 GOTO 100,100,100,100,100,100,100,100
257 GOTO 100,100,100,100,100,100,100,100
258 GOTO 100,100,100,100,100,100,100,100
259 GOTO 100,100,100,100,100,100,100,100
260 GOTO 100,100,100,100,100,100,100,100
261 GOTO 100,100,100,100,100,100,100,100
262 GOTO 100,100,100,100,100,100,100,100
263 GOTO 100,100,100,100,100,100,100,100
264 GOTO 100,100,100,100,100,100,100,100
265 GOTO 100,100,100,100,100,100,100,100
266 GOTO 100,100,100,100,100,100,100,100
267 GOTO 100,100,100,100,100,100,100,100
268 GOTO 100,100,100,100,100,100,100,100
269 GOTO 100,100,100,100,100,100,100,100
270 GOTO 100,100,100,100,100,100,100,100
271 GOTO 100,100,100,100,100,100,100,100
272 GOTO 100,100,100,100,100,100,100,100
273 GOTO 100,100,100,100,100,100,100,100
274 GOTO 100,100,100,100,100,100,100,100
275 GOTO 100,100,100,100,100,100,100,100
276 GOTO 100,100,100,100,100,100,100,100
277 GOTO 100,100,100,100,100,100,100,100
278 GOTO 100,100,100,100,100,100,100,100
279 GOTO 100,100,100,100,100,100,100,100
280 GOTO 100,100,100,100,100,100,100,100
281 GOTO 100,100,100,100,100,100,100,100
282 GOTO 100,100,100,100,100,100,100,100
283 GOTO 100,100,100,100,100,100,100,100
284 GOTO 100,100,100,100,100,100,100,100
285 GOTO 100,100,100,100,100,100,100,100
286 GOTO 100,100,100,100,100,100,100,100
287 GOTO 100,100,100,100,100,100,100,100
288 GOTO 100,100,100,100,100,100,100,100
289 GOTO 100,100,100,100,100,100,100,100
290 GOTO 100,100,100,100,100,100,100,100
291 GOTO 100,100,100,100,100,100,100,100
292 GOTO 100,100,100,100,100,100,100,100
293 GOTO 100,100,100,100,100,100,100,100
294 GOTO 100,100,100,100,100,100,100,100
295 GOTO 100,100,100,100,100,100,100,100
296 GOTO 100,100,100,100,100,100,100,100
297 GOTO 100,100,100,100,100,100,100,100
298 GOTO 100,100,100,100,100,100,100,100
299 GOTO 100,100,100,100,100,100,100,100
300 GOTO 100,100,100,100,100,100,100,100

```



# ✓ Get it right!

100	1000	10000	100000
100	1000	10000	100000
101	1010	10100	101000
102	1020	10200	102000
103	1030	10300	103000
104	1040	10400	104000
105	1050	10500	105000
106	1060	10600	106000
107	1070	10700	107000
108	1080	10800	108000
109	1090	10900	109000
110	1100	11000	110000
111	1110	11100	111000
112	1120	11200	112000
113	1130	11300	113000
114	1140	11400	114000
115	1150	11500	115000
116	1160	11600	116000
117	1170	11700	117000
118	1180	11800	118000
119	1190	11900	119000
120	1200	12000	120000
121	1210	12100	121000
122	1220	12200	122000
123	1230	12300	123000
124	1240	12400	124000
125	1250	12500	125000
126	1260	12600	126000
127	1270	12700	127000
128	1280	12800	128000
129	1290	12900	129000
130	1300	13000	130000
131	1310	13100	131000
132	1320	13200	132000
133	1330	13300	133000
134	1340	13400	134000
135	1350	13500	135000
136	1360	13600	136000
137	1370	13700	137000
138	1380	13800	138000
139	1390	13900	139000
140	1400	14000	140000
141	1410	14100	141000
142	1420	14200	142000
143	1430	14300	143000
144	1440	14400	144000
145	1450	14500	145000
146	1460	14600	146000
147	1470	14700	147000
148	1480	14800	148000
149	1490	14900	149000
150	1500	15000	150000
151	1510	15100	151000
152	1520	15200	152000
153	1530	15300	153000
154	1540	15400	154000
155	1550	15500	155000
156	1560	15600	156000
157	1570	15700	157000
158	1580	15800	158000
159	1590	15900	159000
160	1600	16000	160000
161	1610	16100	161000
162	1620	16200	162000
163	1630	16300	163000
164	1640	16400	164000
165	1650	16500	165000
166	1660	16600	166000
167	1670	16700	167000
168	1680	16800	168000
169	1690	16900	169000
170	1700	17000	170000
171	1710	17100	171000
172	1720	17200	172000
173	1730	17300	173000
174	1740	17400	174000
175	1750	17500	175000
176	1760	17600	176000
177	1770	17700	177000
178	1780	17800	178000
179	1790	17900	179000
180	1800	18000	180000
181	1810	18100	181000
182	1820	18200	182000
183	1830	18300	183000
184	1840	18400	184000
185	1850	18500	185000
186	1860	18600	186000
187	1870	18700	187000
188	1880	18800	188000
189	1890	18900	189000
190	1900	19000	190000
191	1910	19100	191000
192	1920	19200	192000
193	1930	19300	193000
194	1940	19400	194000
195	1950	19500	195000
196	1960	19600	196000
197	1970	19700	197000
198	1980	19800	198000
199	1990	19900	199000
200	2000	20000	200000

Main program

The checksum table for the main program printed here is for the original listing. Once you *Print* the program it will modify itself, and hence give a different table of values. Make sure that you *checksum* it, removing any typing errors, before you *RUN* it, otherwise you may have to type it all over again.

1000	10000	100000	1000000
1000	10000	100000	1000000
1001	10010	100100	1001000
1002	10020	100200	1002000
1003	10030	100300	1003000
1004	10040	100400	1004000
1005	10050	100500	1005000
1006	10060	100600	1006000
1007	10070	100700	1007000
1008	10080	100800	1008000
1009	10090	100900	1009000
1010	10100	101000	1010000
1011	10110	101100	1011000
1012	10120	101200	1012000
1013	10130	101300	1013000
1014	10140	101400	1014000
1015	10150	101500	1015000
1016	10160	101600	1016000
1017	10170	101700	1017000
1018	10180	101800	1018000
1019	10190	101900	1019000
1020	10200	102000	1020000
1021	10210	102100	1021000
1022	10220	102200	1022000
1023	10230	102300	1023000
1024	10240	102400	1024000
1025	10250	102500	1025000
1026	10260	102600	1026000
1027	10270	102700	1027000
1028	10280	102800	1028000
1029	10290	102900	1029000
1030	10300	103000	1030000
1031	10310	103100	1031000
1032	10320	103200	1032000
1033	10330	103300	1033000
1034	10340	103400	1034000
1035	10350	103500	1035000
1036	10360	103600	1036000
1037	10370	103700	1037000
1038	10380	103800	1038000
1039	10390	103900	1039000
1040	10400	104000	1040000
1041	10410	104100	1041000
1042	10420	104200	1042000
1043	10430	104300	1043000
1044	10440	104400	1044000
1045	10450	104500	1045000
1046	10460	104600	1046000
1047	10470	104700	1047000
1048	10480	104800	1048000
1049	10490	104900	1049000
1050	10500	105000	1050000
1051	10510	105100	1051000
1052	10520	105200	1052000
1053	10530	105300	1053000
1054	10540	105400	1054000
1055	10550	105500	1055000
1056	10560	105600	1056000
1057	10570	105700	1057000
1058	10580	105800	1058000
1059	10590	105900	1059000
1060	10600	106000	1060000
1061	10610	106100	1061000
1062	10620	106200	1062000
1063	10630	106300	1063000
1064	10640	106400	1064000
1065	10650	106500	1065000
1066	10660	106600	1066000
1067	10670	106700	1067000
1068	10680	106800	1068000
1069	10690	106900	1069000
1070	10700	107000	1070000
1071	10710	107100	1071000
1072	10720	107200	1072000
1073	10730	107300	1073000
1074	10740	107400	1074000
1075	10750	107500	1075000
1076	10760	107600	1076000
1077	10770	107700	1077000
1078	10780	107800	1078000
1079	10790	107900	1079000
1080	10800	108000	1080000
1081	10810	108100	1081000
1082	10820	108200	1082000
1083	10830	108300	1083000
1084	10840	108400	1084000
1085	10850	108500	1085000
1086	10860	108600	1086000
1087	10870	108700	1087000
1088	10880	108800	1088000
1089	10890	108900	1089000
1090	10900	109000	1090000
1091	10910	109100	1091000
1092	10920	109200	1092000
1093	10930	109300	1093000
1094	10940	109400	1094000
1095	10950	109500	1095000
1096	10960	109600	1096000
1097	10970	109700	1097000
1098	10980	109800	1098000
1099	10990	109900	1099000
1100	11000	110000	1100000

Demo 1: Computer Pass

1000	10000	100000	1000000
1000	10000	100000	1000000
1001	10010	100100	1001000
1002	10020	100200	1002000
1003	10030	100300	1003000
1004	10040	100400	1004000
1005	10050	100500	1005000
1006	10060	100600	1006000
1007	10070	100700	1007000
1008	10080	100800	1008000
1009	10090	100900	1009000
1010	10100	101000	1010000
1011	10110	101100	1011000
1012	10120	101200	1012000
1013	10130	101300	1013000
1014	10140	101400	1014000
1015	10150	101500	1015000
1016	10160	101600	1016000
1017	10170	101700	1017000
1018	10180	101800	1018000
1019	10190	101900	1019000
1020	10200	102000	1020000
1021	10210	102100	1021000
1022	10220	102200	1022000
1023	10230	102300	1023000
1024	10240	102400	1024000
1025	10250	102500	1025000
1026	10260	102600	1026000
1027	10270	102700	1027000
1028	10280	102800	1028000
1029	10290	102900	1029000
1030	10300	103000	1030000
1031	10310	103100	1031000
1032	10320	103200	1032000
1033	10330	103300	1033000
1034	10340	103400	1034000
1035	10350	103500	1035000
1036	10360	103600	1036000
1037	10370	103700	1037000
1038	10380	103800	1038000
1039	10390	103900	1039000
1040	10400	104000	1040000
1041	10410	104100	1041000
1042	10420	104200	1042000
1043	10430	104300	1043000
1044	10440	104400	1044000
1045	10450	104500	1045000
1046	10460	104600	1046000
1047	10470	104700	1047000
1048	10480	104800	1048000
1049	10490	104900	1049000
1050	10500	105000	1050000
1051	10510	105100	1051000
1052	10520	105200	1052000
1053	10530	105300	1053000
1054	10540	105400	1054000
1055	10550	105500	1055000
1056	10560	105600	1056000
1057	10570	105700	1057000
1058	1058		

## These back issues are still available

**June issue:** Analysis of the 13800, Submarine, Advertising (Random numbers), Software reviews, Frog Jump, Monopoly, Security, Atari Insights - 50 (see Department Graphics, special 12 page feature on Commodore.com)

**July issue:** Chessmaster, Battle Plan, SOX 3.5, IT Commandments, Advertising, Display List Tutorial, Software reviews, Power Functions, Treasure Hunt, Keyboard Sounds, Monopoly, Insights - 50 (see, Department and Graphics)

**August issue:** Analysis of 52001, program generation, evaluation, Print Samples, Accounting, Trouble, Tables programs, first look at Logo, Radar 1990, the 2.0 upgrade offer, Display List Tutorial, Monopoly, Software reviews, Insights - 50 (see, Department and Graphics)

**September issue:** Sample resolution, the 52001, Mode 8, word, always ready, New Month, Date, Make, Display List Tutorial, 68000 addressing modes, for processing with Logo, Software reviews, Insights - 50 (see, Department and Graphics)

**October issue:** Database, Camera, graphics programs, Updates for 5000, 5000 assembler, 5000 Ram drive utility, Mac/Atar connecting through, Pentium, Software reviews, 58000 operating environment, Shipping, Insights - 50 (see, Department and Graphics)

**November issue:** Database programs, Business operation utility, IT graphics examples, IT software for, Day, Weather game, Display List tutorial, Advertising, Monopoly, Software reviews, Insights - 50 (see, Department and Graphics)

**March issue:** Machine code generator, 3, Knight's tour program, Complete 19, 1, Superchip module, Chat & review for early prices, Book reviews, Advertising, Software reviews, PLUS Atari ST User & Special section, Jeff Minter and Commodore, Main Pt. 1, Making the most of the ST

**April issue:** Sound synthesizer, Computer Pt. 2, Using modes 10 and 15, Monopoly strategy game, also displaying printing utility, Advertising, Software reviews, PLUS Atari ST User (review of 700k Module 2, more on the Midrange, Making the most of the ST, and all of the latest news)

**May issue:** Sam, Tutorial, Database, Camera, Escape maze game, Computer Pt. 3, Paper Mario Graphics Pt. 1, Atari's Spelling Programs, Advertising, Software reviews, PLUS Atari ST User, Atari Monopoly, The Pass and Dale (Julian's review), ST Graphics Pt. 1, DisplayList/Output, Picture conversion, programs, Making the most of your ST, and the latest 5.00 and 5.01 news from the States

**June issue:** Camera, Escape maze editor, Part 1 of Great Lines of Code - build a tight network, First look of Atari's Graphics, the Lotus No. 1, Paper Mario Graphics Pt. 2, Advertising, Software reviews, American Soccer, PLUS Atari ST User, Conclusion of ST Art series, Sprint 2.0 maze game in Beta, Two Young upgrades, Software reviews, Making the most of the ST, and all the latest ST news

**July issue:** Space Maze maze game, Camera history opened packets, Atari in Poland, the 10 Best list, Paper Mario Graphics Part 3, Five Lines No. 2, Advertising, Game & Display packages review, American Soccer, Shipping Issues, PLUS Atari ST User, 500 graphics in ST Beta, Three C packages reviewed, Making the most of the ST, Reviews 5.07 News

£1.25  
each



# Keying in long programs too much of a chore?

Then give your fingers a rest by sending for our monthly disc or cassette containing all the programs from each issue of Atari User. See order form opposite.



£3.95



£5.95

**JUNE: Frog Jump:** Guide the frog to his home in this version of the arcade classic. **LOBBING RAM PROGRAM:** Use the extra RAM of memory to good effect. **Submarine:** Steer the submarine. **Wrecks-Monster:** Drive pictures with a joystick. **Random Numbers:** Get random numbers from machine code. **Fifty Fifteen:** Can you keep the Fifty Fifteen happy in their cave?

**JULY: Bank Run:** Fatten the deserted city and find utility. **Disassemble:** Find out what's going on deep inside your Atari. **Tesserae Hunt:** Can logical thinking fix the tesserae. **Forecast Generator:** Keep generating passwords all you find out you like. **Ketchenko:** Convert your minis into an organ. **Quadrangle:** Can you sort out the maze of ropes in the belly?

**AUGUST: Assembler:** Make machine code programming easier. **Print! Generator:** Save money with this fun machine simulation. **Hexcode:** Complex patterns made easy. **Protocols:** Protect your programs from spying eyes. **Display List:** Demonstration program. **Master 1000:** Funniest test adventure. **Tech Talker:** Demonstration program.

**SEPTEMBER: Place March:** Help Hester the Web search the maze mazes! **World Data Maker:** Convert your machine code routines to DATA statements. **Display List:** Demonstration program. **Screen Change:** Dump your Mode II screen to a 1024 pixel. **Water:** Solve the Bubbles problem.

**OCTOBER: Postman:** That? **Post Memory Dump:** Essential dictionary in hex and Ascii. **Display List:** Demonstration program. **Strip Edge:** Action game for one or two players. **Computer Games:** Make your own mini computer. **Assembler Update:** Improvements for RAM assembler. **Ram Disc:** Make the most of the 100K's extra memory.

**NOVEMBER: Guy Fawkes:** Help Guy escape from the guards. **Caravan:** Teach your Atari to

be a paralympic. **Display List:** Demonstration program. **Wildcard Operations:** Utility to provide logical functions. **Circle:** Draw and fill a circle. **Plus/Minus:** Utility of the month. **Compline:** machine code pinball game.

**DECEMBER: Commodore:** More version of the famous TV game. **Get It Right!:** Atari User's own chess run program. **Discs:** Use of hardware on your Atari. **Lib Utility:** Makes listing easier. **Display List:** Demonstration program. **Plus:** Feature of the month. **Jan's Program:** machine code entertainment.

**BASIC COMPILER:** The entire set of Basic Compiler programs from the March to June issues on one tape/disc. Write easy routines for your programs that will run at machine code speed. Contains the Compiler, Library and many example programs. (N.B. Requires an assembler.)

**JANUARY 1986: Balls:** Help Mr. Hump rescue Littleville. **Get It Right!:** Atari User's own chessboard program. **Alan Atkins:** The game to accompany the machine code editor. **Lister:** Make listing programs easy. **Data:** Test the mine or another player of the strategy game. **Drango:** 1020 protoboarder system. **Plus:** Feature of the month. **Scramble:** Fighter machine code game.

**FEBRUARY: Monopolizer:** Programs to manipulate Monopoly's account. **Check:** Interesting about check utility. **Flashes:** Flashing colour utility. **Bottom:** Simple to play, hard to master. **Bridge:** The thinking person's card game. **Plus:**

Feature of the month. **Demon's Lair:** An adventure game you may never escape from.

**MARCH: Horse Play:** Knight's tour program. **Basic Compiler Program:** to accompany the new series. **Alan Atkins:** First part of monthly listing. **Plus/Minus:** Feature of the month. **Wrecker in the Cave:** An app you keep your head and help Wrecker find his?

**APRIL: Synthesizer:** Activate the hidden depths of the Atari sound chip. **Disc Index:** Keep track of disc files and how space with the index printing program. **Hexcode:** Make the most of Modes II and III. **Hexcode:** Can you beat the game that learns from its mistakes?

**MAY: Caravan Europe:** Can you help build Caravan escape from the labyrinth with King Monopoly's gold? **Play! Hexcode:** Program to accompany the start of the series. **Spelling:** Automate those weekly school spelling tests.

**JUNE: Place Compiler:** Create hundreds of new mazes for last month's Caravan Escape game. **Play! Hexcode Developer:** Create your own DATA shapes with the Play! Hexcode Editor. **Hex Lines I:** Simple Disc editor/printer - build it into your own program.

**JULY: Space Maze:** Manoeuvre your space ship through the labyrinthian space caverns. **Play! Hexcode Example:** Program to display your Play! Hexcode graphics. **Jackpot:** The original video slot machine simulation plus your own multi-line port interface. **Hex Lines II:** Create 800,000-type displays on your Atari. **Random:** Connect those bugs in the XL's Ram. **Hex Lines:**

**AUGUST: Frank the Frog:** Avoid Action in Basic and machine code. **80 Columns:** Display Use 80 column displays on your 5 bit Atari. **Get It Right!** By popular demand, an updated version of our chessboard program. **Play! Hexcode Programs:** Three programs to move your RAMs vertically. **Hex Lines:** Moved our utilities to hex lines.

TO ORDER USE THE SPECIAL FORM ON PAGE 49

## SAVE up to 50% on insuring your Atari, monitor, disc drives and printers

Atari User has negotiated special terms with one of Britain's biggest insurance companies to cover your Atari system (including all peripherals) against theft, breakdown or accidental damage.

Your system is covered while it is in your own home, while it is temporarily elsewhere, and even when it is in transit – but only within the UK.

Examples of this special price:

System worth	Premium
£200	From £10
£500	From £16
£1,000	From £35

Differences apply to rural, urban and metropolitan areas.

A scheme is also available for businesses which provides additional cover. Details on application.

*No insurance package has ever before offered you breakdown cover.*

**FOR SUBSCRIBERS ONLY!** The offers on this page are restricted to subscribers to Atari User. See the order form on the right.



You can save  
up to £10 with  
this exclusive  
reader offer



For Atari ST users we can offer FOUR 3½" single sided discs for the unbeatable price of **£5.99!**

**Your Atari  
needs protecting**

Now  
for the  
800XL and  
5T range

Protect your 520ST, 1040ST, 130XE or 800XL with our luxury dust cover made of soft, pliable, clear and water-resistant vinyl, bound with strong cotton and decorated with the magazine's logo.



only  
**£3.95**

**Keep your  
collection complete**

Bound in chocolate brown PVC and bearing the Atari User logo, this handsome binder will hold a year's supply of the magazine firmly secured in place with metal rods.



only  
**£3.95**

**Here's a really  
unbeatable offer  
for all Atari users!**

This top-quality T-shirt woven in an attractive shade of grey with the Atari logo in red is a genuine American 'Fruit of the Looms' product, made from 50% cotton/50% polyester.

The T-shirts are on sale at £9.25. Please order on our official order form.

Made in three sizes:

Small - 36"-36"  
Medium - 36"-38"  
Large - 38"-40"

All prices include postage packing and VAT. Valid to August 31, 1986. Orders within the UK only.

(/) £ P

**Annual subscription** UK & IRE £12 (including order) 2000  
Europe £15 2000  
Overseas (airmail) £28 2000  
Continental only 1000

**Reprints** UK & IRE £12 (including order) 2000  
Europe £15 2000  
Overseas (airmail) £28 2000

**Exclusive offers for subscribers**  
Discount offer (with information) 2000  
4.5% disc 01.99 2000  
Add £2 for Europe/Overseas  
\*These offers are available to all subscribers.

**Back issues**  
£1.25 UK & IRE £2.25 Europe £3.25 Overseas Annual  
1985 May 2000  
Jun 2000  
Jul 2000  
1986 Mar 2000  
Apr 2000  
May 2000  
Jun 2000  
Jul 2000

**Monthly Cassette/Disc**

Cassette	Disc	Price	Type	Disc
£1.25 UK	May	2017	Audiocassette	
£2.25 Europe/Overseas	Jun	2018		
	Jul	2019		
	Aug	2020		
	Sep	2021		
	Oct	2022/2023		
	Nov	2023/2024		
	Dec	2024/2025		
	1986 Jan	2025/2026		
	Feb	2026/2027		
	Mar	2027/2028		
	Apr	2028/2029		
	May	2029/2030		
	Jun	2030/2031		
	Jul	2031/2032		
	Aug	2032/2033		

**Basic Computer Cassette/Disc**  
Cassette £2.75 UK £4.25 Europe/Overseas Year 2000  
3 1/2" Disc £2.95 UK £4.45 Europe/Overseas Year 2000

**Communications Books**  
OF only The New Handbook of Computers £2.95 2075  
Software £2.95 2075  
Book books £2.95 2075

**T-Shirt**  
£4.25 UK Small 2000  
£5.25 Europe Medium 2000  
£5.75 Overseas Annual Large 2000

**Dust Cover**  
£3.95 UK 1000 2000  
£4.95 Europe/Overseas 520ST range 2000  
1040ST range 2000

**Binder**  
£3.95 UK £5.95 Europe £9.95 Overseas Air Mail 2000

Payment please indicate method ( / )  
 Home/Restaurant/Convenience Store/Cash/Visa TOTAL 2000

Charge PO made payable to Datacube Publications Ltd. Credit card copy sent

Name \_\_\_\_\_/Signed

Address \_\_\_\_\_

Post code \_\_\_\_\_ Tel \_\_\_\_\_

Send to: Atari User, FREEPOST, Europa House, 80 Chester Road, Hound Green, Southampton SO2 0YF.

Please allow 4/5 weeks for delivery.

Order at any time of the day or night.

Order by Post Telephone Order 041-629-7931  
Buy '86 from 41-629-6282

Discover why 041-629-7931  
Phone Open

Atari's finest in your name, address and credit card number







## **ATARI DB-CALC**

*for use with the  
520 ST, 520 ST PLUS, 260 ST & 1040 ST.*

*Voted best database programme  
(with calculating possibilities)  
in Holland and Belgium.*

*Power without Price*

Why spend hundreds of pounds organising your home or office life when you can do it for **£49.95**.

So simple to use no technical knowledge is needed.

An excellent easy to use database with built in calculating facilities specially designed to take full advantage of the Gem Environment and other elegant features built into your ST computer.

- Compatible with the mouse.
- Suitable for customised reports, mailing labels, memberships, sales statistics, stocklists, office and personal records, financial statistics, etc.
- Designed for half megabyte and one megabyte versions - if you upgrade you don't need to create new files as you can expand your existing ones and there is no need to buy more expensive software packages.
- Ready made files for:- Addresses, break even point analysis, cash book, labels, quantity discounts, print lists, statistics
- Unlimited number of fields per record.

- Superfast sort and an ever faster search on a field.
- Selects on any combination of fields.
- Built in text editor which integrates with mailmerge.
- Allows all printer control functions.



- Built in calculator to help you work out difficult mathematical formulae.
- Comprehensive manual which can easily be understood.

For further information contact:-

# **Robtek**



# HITTING THE STREETS

with two games from Databyte



## AZTEC

The first really new idea in real-time adventure! Hi Res action-  
animation on-screen every moment. Dozens of "living"  
"breathing" monsters, Aztec guards, trapdoors, death-reveal Real  
Life movement - walking, jumping, crawling, hanging, fighting.  
There's never been anything as novel, as exciting!

Cassette 9.95 Disk 14.95.

## POLAR PIERRE

POLARPIERRE is a one or two player arcade game where two can  
play at the SAME TIME. Guide Pierre or his cousin Jacques  
through an icy obstacle course, racing through ski jumps, mystery  
chalets, lifts, electro-generators, shooting snowballs, springboards,  
strainers, and thunder and lightning.

POLARPIERRE also includes a construction set so you can design  
and play your own screens.

Cassette 9.95 Disk 14.95.

# DATABYTE

Available from all good computer stores, or post free from  
DATABYTE, 15 Wolsey Mews, Kentish Town, London NW5. Telephone 01-482 1755

Aztec and Polar Pierre are trademarks of Databyte Ltd. © 1984, 1985, 1988. Distributed by Databyte under license from  
Databyte Inc. Distribution arranged by International Computer Group.

For All ATARI Computers: 408-890/512K with minimum 49K RAM.