A Database Publication

Computing with the AMSTRAD

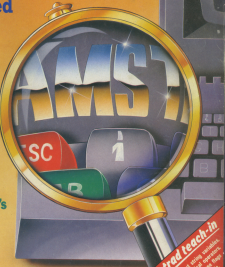
No. 9
September 1985
£1

The independent magazine for CPC464/664 users

Enhanced text: the vital routines

The ins and outs of mazes: hints for adventurers

We explore the Amstrad's role as a business micro

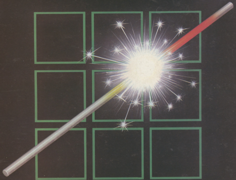


Amstrad teach-in

Find out about string variables, bit masks, logical operators, – and how to use flags in machine code.

GAMES Go coin-hunting on Planet Arg . . . or fight off the evil weevils

007



Computing with the AMSTRAD
The independent magazine for CPC 464 users
Enhanced text: the vital routines
The ins and outs of mazes: hints for adventurers
We explore the Amstrad's role as a business micro
GAMES Go coin-hunting on Planet Arg ...or fight off the evil weevils
Amstrad teach-in

Vol. 1 No. 9 September 1985

Managing Editor: **Derek Meakin**
Features Editor: **Pete Bibby**
The A Team: **Mike Bibby**
Alan McLachlan
Kevin Edwards
Roland Waddilove
Production Editor: **Peter Glover**
Layout Design: **Heather Sheldrick**
News editor: **Mike Cowley**
Advertisement Manager: **John Riding**
Advertising Sales: **Margaret Clarke**
Editor in Chief: **Peter Brameld**

Editorial: 061-456 8835
Administration: 061-456 8383
Advertising: 061-456 8500
Subscriptions: 061-480 0171
Telecom Gold Mailbox: 79:MAG001
Telex: 256871 MONREF G
Quoting Ref. 79:MAG001
Prestel Mailbox: 614568383

Published by:
Database Publications Ltd,
Europa House, 68 Chester Road,
Hazel Grove, Stockport SK7 5NY.

Subscription rates for
12 issues, post free:
£12 – UK & Eire (Sterling only)
£20 – Rest of world (surface)
£40 – Rest of world (airmail)

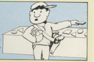
ABC Member of Audit
Bureau of Circulations

"Computing with the Amstrad" welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a 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.

While every care is taken, the publishers cannot be held legally responsible for any errors in articles, listings or advertisements.

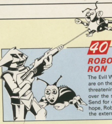
"Computing with the Amstrad" is an independent publication and neither Amstrad Consumer Electronics plc or Amsoft are responsible for any of the articles in this issue or for any of the opinions expressed.

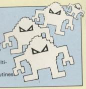
News trade distribution:
Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Tel: 0293 27053.

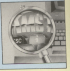


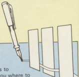
Dungeon Adv
Jet Set
FINAL FRONTIER
Level 9 C
SUPERCHESS FOR THE AMSTRAD CPC 464
DEEP THOUGHT
ARTIST AND SPRITE DESIGNER

Contents



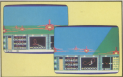






Now YOU can fly with the legendary Red Arrows – in the most challenging flight simulation ever!

It's the most exciting flight simulator ever written for a home computer – the product of many months of dedicated work by some of Britain's top programmers, enthusiastically aided by the talents of aircraft designers, engineers, mathematicians – and the Red Arrow pilots themselves.



FREE RED ARROW CONTEST

Everyone buying the program can enter an exciting competition. Winners will be given a VIP visit to the Red Arrows base at RAF Scrampton, the wartime home of the Dambusters, including two nights' accommodation at a luxury hotel. While at Scampton you will be invited to sit at the controls of a Hawk.

ORDER FORM

	Tape (£8.95)	5¼" Disc (£11.95)	3" Disc (£12.95)	3½" Disc (£12.95)
Amstrad	□	N/A	□	N/A
Atari	□	□	N/A	N/A
BBC B	□	□	□	N/A
Comm. 64	□	□	N/A	N/A
Electron	□	N/A	N/A	□
Spectrum	□	N/A	N/A	□

I wish to pay by:

□ Access/Mastercard/Eurocard No.

□ Barclaycard/Visa No.

□ Cheque/PO made payable to Database Publications Ltd.

Expiry date: / /

Name

Address

Signed

Send to: Database Software, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

(No stamp needed if posted in UK) *Please allow 28 days for delivery*

YOU CAN ALSO ORDER BY PHONE: 061-480 0171 *Don't forget to quote your credit card number and full address.* CA9

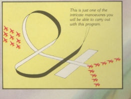
Every ounce of power contained in the micro, and its enhanced sound and graphics capabilities, is used to give the utmost realism to re-creating the most spectacular aeronautical displays ever seen in the skies of Britain.

You start by practising take offs and landings. Then, once you have won your wings, you fly in formation as part of the Red Arrows team. There's no margin for error as you fly a mere six to 10 feet from each other – at speeds of between 300 and 350 miles an hour!

But the real drama begins as you plunge into the death-defying manoeuvres that have been thrilling crowds at air shows for the last 21 years.

On the panel in front of you are all the instruments you need – plus a screen giving you an external view of the complete formation you are flying. Slip out of line for a second and the eagle-eyed Red Leader will be on the radio ordering you back into position.

The program comes with a detailed flight handbook that will soon give you the confidence to take YOUR place alongside the ace pilots of the Red Arrows, even if you've never flown before!



This is just one of the intricate manoeuvres you will be able to carry out with this program.

Put yourself in the pilot's seat of the most manoeuvrable fighter in the RAF!

RED ARROWS
A gripping, realistic computer simulation for the
Commodore
Spectrum
Amstrad
Electron
BBC Micro
Atari

ACCESS ORDERS NOW TAKEN

Haystack Peripherals CPC464 Software Haystack Peripherals

8 MIDGROVE, DELPH, OLDHAM OL3 5EJ Tel: 045 77 5229

	RRP	OUR PRICE
A'n'F		
Chuckie Egg	7.90	6.90
Chuckie Egg 2	8.95	7.95
ADDICTIVE		
Football Manager	7.95	6.95
ADVENTURE INTERNATIONAL		
Gremlins	8.95	7.95
ALLIGATA		
Blagger	8.95	7.95
Defend or Die	7.95	6.95
ACTIVISION		
Ghostbusters	10.99	9.99
AMSOFT		
American Football	9.95	8.95
Amsword (Advanced)	19.95	17.95
Amsword (Easy)	9.95	8.95
Centrecourt	8.95	7.95
Concise Basic Spec.	19.95	17.95
Concise Firmware Spec.	19.95	17.95
DFM Database*	14.95	12.95
Decision Maker	21.95	19.95
Devpac Ass/Disass	24.95	21.95
Entrepreneur	21.95	19.95
Fantastic Voyage	8.95	7.95
Guide to Basic I	19.95	17.95
Guide to Basic II	19.95	17.95
Hisoft DEVPAC*	34.95	29.95
Hisoft PASCAL	34.95	29.95
Home Accounts Manager*	14.95	12.95
Hurricane Higgins Pool	8.95	7.95
Hurricane Higgins Snooker	8.95	7.95
Investat*	29.95	26.95
Jet Boot Jack	8.95	7.95
Master Chess	8.95	7.95
Masterfile	24.95	22.95
Pitman Typing Tutor	9.25	8.25
Project Planner*	24.95	22.95
Scout Steps Out	8.95	7.95
Screen Designer	£14.95	12.95
Spreadsheet*	19.95	17.95
Starwatcher	17.45	15.45
Stock Aid*	29.95	26.95
Stockmarket*	8.95	7.95
Transact*	29.95	26.95
ANIROG		
Flightpath 737	6.95	5.95
Jump Jet	9.95	8.95
Moon Buggy*	7.95	6.95
ARIOLASOFT		
Archon	10.95	9.95
Hard Hat Mack*	8.95	7.95
One On One	8.95	7.95
ARTIC		
World Cup Football	7.95	6.95
ASK		
Number Painter	8.95	7.95
CCS		
Warzone	6.95	5.95
CDS		
Castle Blackstar	6.95	5.95
Steve Davis Snooker	7.95	6.95
C.P. SOFTWARE		
Backgammon	8.95	7.95
Bridge Player	9.95	8.95
Draughts	8.95	7.95
Pinball Wizard	8.95	7.95
Superchess	9.95	8.95
CRL		
Amstrad Artist	9.95	8.95
Handicap Golf	6.95	5.95
Rocky Horror Show	9.95	8.95
Test Match	6.95	5.95
CABLE		
Digger Barnes	7.95	6.95
CAMEL		
FlexiFrend	7.50	6.50
Grasp	8.50	7.50
CREATIVE SPARKS		
Dangermouse in Double Trouble	8.95	7.95
DATABASE		
Mini Office	5.95	5.25

	RRP	OUR PRICE
Red Arrows	8.95	7.95
DESIGN DESIGN		
Dark Star	7.95	6.95
Tank Busters	7.95	6.95
DIGITAL INTEGRATION		
Fighter Pilot	8.95	7.95
DK'Tronics		
Minder	9.95	8.95
DOMARK		
A View to a Kill	10.99	9.99
DURELL		
Combat Lynx	8.95	7.95
Death Pit	7.95	6.95
ELITE		
Frank Bruno's Boxing	8.95	7.95
GARGOYLE		
Dun Darach	9.95	8.95
GEMINI		
Cashbook* (inc. VAT file)	59.95	51.95
Database	19.95	17.95
Final Accounts*	59.95	51.95
Home Accounts	19.95	17.95
Report Generator	19.95	17.95
VAT File*	19.95	17.95
Combo Pack* (Cashbook + final Acc.)	89.95	79.95
GILSOFT		
The Quill	16.95	15.25
GLOBAL		
Operation Caretaker	9.95	8.95
HEWSON		
Zapp	14.95	13.45
HISOFT		
Font 464*	7.95	6.95
IMAGINE		
World Series Baseball*	8.95	7.95
INCENTIVE		
Confuzion	6.95	6.25
Millionaire	6.95	5.95
INTERCEPTOR		
Azimuth 3000	8.95	7.95
Forest at Worlds End	6.00	5.25
Heroes of Karn	6.00	5.25
Jewels of Babylon	6.00	5.25
Message from Andromeda	6.00	5.25
Warlord*	6.00	5.25
INTRIGUE		
Tomb of Kuslak	7.50	6.50
KNIGHTSOFT		
Animated Strip Poker	8.95	7.95
KUMA		
Artwork	9.95	8.95
Database	14.95	12.95
Easi VAT*	39.50	34.50
Forth	19.95	17.95
Fruity Frank	6.95	5.95
Galaxia	5.95	5.25
Gems of Stradus	7.95	6.95
Holdfast	5.95	5.25
Home Budget	19.95	17.95
Music Composer	9.95	8.95
Star Avenger	6.95	5.95
LEVEL 9		
Adventure Quest	9.95	8.45
Colossal Adventure	9.95	8.45
Dungeon Adventure	9.95	8.45
Emerald Isle	6.95	5.95
Lords of Time	9.95	8.45
Red Moon	6.95	5.95
Return to Eden	9.95	8.45
Snowball	9.95	8.45
LLAMASOFT		
Psychedelia	6.00	5.25
LOTHLORIEN (WARMASTER)		
Johnny Reb	6.95	5.95
Redcoats	6.95	5.95
Special Operations	6.95	5.95
MARTECH		
Brian Jack's Superstar Chall.	7.95	6.95
MELBOURNE HOUSE		
Hobbit	14.95	12.95

	RRP	OUR PRICE
Mordon's Quest	6.95	5.95
Way of the Exploding Fist*	9.95	8.95
MICROGEN		
Everyone's a Wally	9.95	8.95
Herberts Dummy Run*	9.95	8.95
Pyjamarama	8.95	7.95
MICROPOWER		
Ghouls	6.95	6.15
Killer Gorilla/Gauntlet	9.95	8.95
MIRRORSOFT		
Caesar's Travels*	7.95	6.95
First Steps/Mr. Men	8.95	7.95
Here + There/Mr. Men	7.95	6.95
MOSAIC		
Erik The Viking	9.95	8.95
MYRRDIN		
Flight Simulation	11.95	10.45
NEW GENERATION		
Machine Code Tutor*	14.95	13.45
OCEAN		
Daley Thompson's Decathlon	8.95	7.95
Hunchback II	8.95	7.95
Kong Strikes Back	8.95	7.95
P.S.S.		
Battle for Midway	9.95	8.95
The Covenant*	6.95	5.95
PICTURESQUE		
The Code Machine	19.95	17.95
REALTIME		
3D Starstrike	6.95	5.95
SHARDS		
Mystery/Java Star	7.95	6.95
SOFTWARE PROJECTS		
Blinkey	7.95	6.95
Jet Set Willy	8.95	7.95
Manic Miner	8.95	7.95
TASKSET		
Super Pipeline 2	8.95	7.95
TASMAN		
TASCOPY	9.90	8.90
TASPRINT	9.90	8.90
TASWORD	19.95	17.95
ULTIMATE		
Alien 8	9.95	8.95
Knightlore	9.95	8.95
U.S. GOLD		
Beach Head	9.95	8.95
Pole Position	9.95	8.95
VIRGIN		
Sorcery	8.95	7.95
VORTEX		
Android One	7.95	6.95
Android Two	7.95	6.95
WINTERSOFT		
Ring of Darkness	9.95	8.95

ADD ONS	
Amstrad Speech Synth	26.95
DK'Tronics Speech Synth	34.95
DK'Tronics Light Pen	24.95
Pro-Ace Joystick	11.95
Superpower Sideways ROM card	39.95
Superpower Disk User Utilities ROM	39.95
Superpower Mailing List ROM	39.95
Superpower Programmers Toolbox*	39.95
Superpower Ass. Disass. + Mach. Code*	39.95

C15 Computer Cassettes (Box of 10) **4.50**

PHONE FOR DETAILS OF OTHER SOFTWARE – WE CAN SUPPLY ALMOST ANY AMSTRAD TITLE AVAILABLE!

Some disc versions now available – ring for prices.

* Please ring to confirm availability

Overseas orders – Please add £1.00 per item.
Send sterling money orders

BEWARE OF CHEAP BOOTLEG SOFTWARE – ALL OUR TITLES ARE ORIGINALS!

Please Send:	Price
TOTAL	

ALL OUR PRICES INCLUDE VAT AND CARRIAGE

Cheques/PO's to:
HAYSTACK PERIPHERALS
8 Midgrove, Delph, Oldham OL3 5EJ
Tel: Saddleworth (04577) 5229

NAME ....

ADDRESS ....

....

Phone No. ....

Access No. ....

All ASCII codes can be printed from the Amstrad computer

Yet another unbeatable deal from Datastar Systems!

STAR SG10

Character Sets

96 Standard ASCII Characters
88 Standard International Characters
96 Italic Characters
88 Italic International Characters
96 NLQ Characters
88 NLQ International Characters

Character Matrix

Standard Characters: 9 dot × 11 dot
STAR Block Graphics: 6 dot × 6 dot
IBM Block Graphics: 12 dot × 11 dot
User Defined: 8 dot × 4 to 11 dot
Near Letter Quality: 17 dot × 11 dot

Character Fonts

Normal (10 CPI)
Elite (12 CPI)
Condensed (17 CPI)
NLQ (10 CPI)

Line Spacing $\frac{1}{6}$, $\frac{1}{8}$ in. or $\frac{7}{72}$ in. Standard

Dimensions (W × D × H)
392 × 315 × 145mm (15.4 × 12.4 × 5.7 in.)

Number of Columns
80, 96, 136, (40, 48, 68 Double Width)

Print Head 9 Wires (User Replaceable)

THE *INCREDIBLE* NEW STAR SG-10 NEAR LETTER QUALITY PRINTER.

features:—

- Epson and IBM compatible print codes
- Friction and tractor feeds.
- Externally Accessible DIP-Switches
- High-resolution bit-image graphics
- Downloadable character sets
- NLQ, italics, condensed, enlarged, super & subscript styles.
- High 120 CPS print speed, logic seeking.
- Hex dump facility.
- Macro-instructions.
- Touch-sensitive control panel.
- Standard parallel Centronics interface.
- 2K print buffer — expandable to 10K
- Commodore 64 version available shortly

Star SG-10 £259 + VAT	£297.85
Parallel cable for any Micro (max. price)	£20
2 Spare ink ribbons	£5
2000 sheets of continuous listing paper	£15
Next day doorstep delivery service	£10
	£347.85

Datastar's all in price £297.85!!!

Post your cheques to

Datastar Systems UK

Unicom House, 182 Royal College Street, London NW1 9NN. Telex 295931 UNICOM G
PERSONAL CALLERS WELCOME — We are situated by the junction of Camden Road, near the railway bridge
MONDAY-FRIDAY 9-6 SUNDAY 10-1 *EXPORT ENQUIRIES WELCOME*

Credit Card Hot Line 01-482 1711

AS MAIN DISTRIBUTORS FOR THE STAR RANGE, WE ARE ABLE TO OFFER EXCELLENT DEALER MARGINS

Vortex SOFTWARE

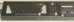
· PLEASE SEND ME ·

□ T.L.L.	AMSTRAD CPC 464	£7.95
□ ANDROID TWO	AMSTRAD CPC 464	£7.95
□ ANDROID ONE	AMSTRAD CPC 464	£7.95
TOTAL VALUE		£

NAME BLOCK LETTERS PLEASE

ADDRESS

ENCLOSE P/O CHEQUE PAYABLE TO VORTEX SOFTWARE (SALES) LTD.
SEND TO VORTEX SOFTWARE (SALES) LIMITED. VORTEX HOUSE,
24 KANSAS AVENUE, OFF SOUTH LANGWORTHY ROAD, SALFORD, M5 2GL.



Breakthrough in education

AMSTRAD computers have made the big breakthrough into the education scene and are now set to seriously challenge the supremacy of the BBC Micro in schools and colleges.

Northern Computers, educational distributors for Amstrad, report a massive upsurge in interest and demand since the latest round of troubles hit Acorn.

Virtually one new education authority a day is ordering either a CPC464 or 664 for evaluation and assessment.

Northern Computers sales boss Gareth Littler says the first ten education authorities and the first two universities have officially started to use Amstrads instead of BBC Micros.

More significantly, individual schools and colleges in more than 20 education authorities and 10 universities are already using Amstrads, he says.

Gareth Littler . . . reports massive upsurge in demand for Amstrads

"Comments from the purchasers have been very positive, especially with regard to educational pricing, which starts at £189 for a complete 64k system that includes a green screen monitor," says Littler.

The Amstrad had a slow start in education, he admits, but interest greatly increased when Acorn Computer shares were first suspended earlier this year.

There was a further round of enquiries from educationists when Acorn shares were suspended again recently.

Littler described the Amstrad versus Acorn situation as being like a boxing match where one opponent has jumped out of the ring.

He believes there are three key factors pointing toward future success for Amstrad machines in the teaching environment.

A language tutor program helps youngsters with experience of BBC Basic learn how to use Amstrad Basic, enabling computer studies teachers to use a mixture of Amstrad and BBC micros.

A free educational primer pack is being produced for secondary and higher levels.

The Amstrad network development is nearing completion and the first installation is due to start up this month at Bootle High School.

This system comprises eight colour Amstrads linked to a 10mbyte hard disc on one site, and ten colour Amstrads connected to a 10mbyte hard disc on another site.

The two school sites are more than half a mile apart and will be linked by a radio-telecommunications aerial on the roof of each building.

PIPPED AT THE POST

AMSTRAD has narrowly failed to win an award that would have stamped it as the major success story of British marketing during the last 12 months.

The Marketing Society shortlisted the achievements of the Amstrad sales and marketing team in the consumer category of its annual awards.

This was in recognition of the Amstrad's feat in capturing 16 per cent of the home computer market in only six months.

But although there was disappointment for marketing chief Malcolm Miller and his colleagues on the awards night, they still had the satisfaction of knowing that the Amstrad's performance had made a deep impression on the judges.

Japanese games for Amstrad

SEVERAL of Japan's top arcade games are to be released as Amstrad programs as a result of a deal between Konami UK and Ocean Software.

The first two titles to be converted for the Amstrad are Hypersports, a multi simulation, and Konami's Tennis. They will cost £8.95 each.

Other titles to follow over the coming months include Yie Ar Kung-Fu, Hyper Rally, Konami's Golf, and fun arcade games Mike and Comic Bakery.

Ocean will produce and publish the programs throughout Europe on the Imagine label.

● Clinching the deal in our picture are Kenji Hiraoka, Konami's UK MD, and David Ward of Ocean Software.

GOLDEN OLDIE

FILMS old and new provide the inspiration for the latest two games for the Amstrad from Datasoft, marketed in the UK by US Gold.

Steven Spielberg's latest blackbuster The Goonies becomes an action/strategy game with eight screens of increasing difficulty in a hunt for pirate treasure, based on the film's plot.

Zorro first appeared on the screen as a 1920s silent movie starring Douglas Fairbanks.

Datasoft has turned it into an adventure game with 15 screens of increasing difficulty.

TRAIN TO FLY WITH THE RED ARROWS

RED Arrows, the new flight simulator from Database Software for the CPC464 and CPC664, makes use of 3D graphics to soar to what one leading reviewer has described as "heights of realism never achieved before".

Written in cooperation with the world-famous pilots themselves, it is a faithful reconstruction of the intricate manoeuvres with which the RAF aerobatics team are currently holding hundreds of thousand of people spellbound at air displays throughout the UK.

Such is the expertise needed to "fly" the Red Arrows simulator that it boasts its own training program with four degrees of difficulty.

"When we first looked at the specifications for the program we realised it was so realistic you would almost need a pilot's skills to come to grips with it", says Derek meakin, head of Database, "so we decided to incorporate a full-scale training program".

Novice flyers receive Help messages as part of their training and the Delete key can pull them into formation if they go off course.

They can also learn to fly the Hawk in solo mode before joining the rest of the team.

The training program includes an auto-pilot facility to control either thrust or steering as the would-be Red Arrow perfects his aerial skills.

Field trials of the Red Arrows program have shown that youngsters are eager to accept the challenge of the simulation. Unperturbed by the level of difficulty involved, they quickly elevated the program to cult game status as they vied with one another to master it.

"What they seem to enjoy most about it is that here at last is a program which does not insult their intelligence", says Derek Meakin.

For those who can prove they have outstanding skills the simulator package offers a free high-score competition to win an all expenses paid weekend, during which they will visit the Red Arrows at their RAF Scampton base.

Priced at £8.95 on cassette, Red Arrows is also available on disc, price £12.95.

Part of the proceeds from the sale of the program are to go to Service charities nominated by members of the Red Arrows team.

HISOFT C FOR CPC

A LANGUAGE which it is claimed is set to become the standard on the next generation of computers is now available for the Amstrad.

Software house HiSoft is offering its own HiSoft C for the Amstrad, a full specification compiler which conforms to the standard definition of the language.

A number of features also make it useful as an interactive learning tool for the beginner, claims the company.

A complete 150-page guide to the C language is provided with the program, which costs £34.95 on tape, £39.95 on disc.

Robson joins in

LATEST of the famous to enter the computer game world is footballer Bryan Robson.

The Manchester United and England captain has put his name to a computer assisted board game for the Amstrad called Bryan Robson's Super League.

Like other board games it has dice, false money, counters and other accessories.

The aim is for eight teams to battle it out for the league title. They travel to away matches, managers buy and sell off players and if unsuccessful see their club go bankrupt.

But unlike a traditional board game, Super League is accompanied by a cassette program. It not only allots league positions but controls home and away fixtures and conditions under which they are played, such as the weather or a team's finances.

Two players can manage all eight teams, with computer assistance, and so compete for the league title. Price: £19.95.

ROM disc player for the Amstrad?

AMSTRAD officials are being non-committal about reports that they intend to bring out a compact disc ROM player to interface with their machines.

Most of the world's leading electronics companies are working on optical memory systems that will allow massive databases to be contained on 12cm laser-read discs.

A ROM player based on a Philips drive unit was shown at the recent Consumer Electronics Show and one report said Amstrad executive William Poel had seen the £450 device and was "very interested."

A source at Philips was quoted as saying his company had already been having discussions with Amstrad about the drive.

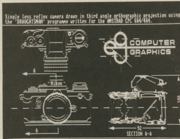
Drive units have been available to original equipment manufacturers in North America since early this year according to tape and disc manufacturer 3M.

Nigel Murphy, 3M UK product manager, says CD data storage is the technology that will substitute for all forms of recording media in 10 to 15 years time.

Roman rough stuff

THE latest adventure game for the Amstrad from Interceptor Micros casts the player in the role of a Celtic warlord in mortal combat with the champion of the Roman legions in the first century AD.

Written by David Banner and with graphics by Terry Greer, the game is set beyond time and space in the home of the Celtic and Roman gods. Price: £6.



Single lens reflex camera drawn in third angle orthographic projection using the 'DRAUGHTSMAN' programme written for the AMSTRAD CPC 664/464.

GRAPHICS PROGRAM SNAPS CAMERA

THIS´screen shot shows a single lens reflex camera drawn in third angle orthographic projection using a program just released for the CPC464 and 664.

Draughtsman costs £19.95 and comes from Leicestershire-based Computer Graphics. The company claims it will exploit the true graphic potential of the machines.

Clocking on

A MULTI-use three part cassette program has been released for the Amstrad by Bubble Bus Software which enables the computer to be used for display purposes and as a clock and calculator.

The first part of the program, Graf It, allows the user to create 3D pie, bar and line graphs in colour.

Clock It turns the Amstrad into an analogue or digital clock with stop, lap and alarm functions and the Calc It provides all the usual calculator functions.

Price: £8.95.

Celebrating

HEWSON Consultants celebrates its fifth birthday in the games software business with three releases for the Amstrad.

First out is the Amstrad conversion of the chart-topping adventure Dragontorc. It will cost £7.95.

Promised for October are an original Amstrad arcade adventure and Southern Belle, a steam engine simulator based on a King Arthur class locomotive on the London to Brighton run.

Books aid machine code beginners

PUBLISHER Melbourne House has brought out two new programming guides for Amstrad owners, plus a text adventure and an arcade game, and promises more new Amstrad titles shortly.

Amstrad Machine Language for the Absolute Beginner is for anyone frustrated by the limitations of Basic who wants to write faster, more powerful programs or subroutines.

The book offers complete instruction in Z80 machine language programming, including specific applications which can be demonstrated and used on the Amstrad, and costs £6.95.

The Z80 Reference Guide is a programming tool that gives a greater understanding of the chip that makes the Amstrad tick. The machine language programming guide costs £9.95.

Other Amstrad book titles to be released by Melbourne House include Amstrad Whole Memory Guide, Amstrad Readymade Machine Language Routines, Amstrad Games Book and Writing Adventure Games on the Amstrad.

Mordon's Quest is a new text adventure written by John Jones-Steele, author of Classic Adventure. The game contains more than 150 locations and 400 words.

Sophisticated text-compression routines allow for the development of complex and challenging problems, with highly-descriptive locations to set the scene. It costs £6.95.

The Way of the Exploding Fist is a new arcade game featuring martial arts, designed to simulate realistically all the traditional skills like blocking, ducking, jumping, kicking, feinting and somersaulting.

Set in a tournament situation playing against the computer or another user, the aim is to work through ten skill levels to become a master of karate. It costs £9.95.

Eight new ROMs on the way

NO fewer than eight helpful new ROMs for the Amstrad are being brought out over the next two months by Micro Power.

They fit the new £39.95 sideways ROM card Super Power which can hold seven ROMs and sits at the back of the machine.

Already available are a disc user's utility with editing facility and a mailing list ROM.

Out this month are a programmer's toolbox and an assembler/disassembler with turtle graphics routines and machine code monitor.

To follow are a word processor and a database, available around the end of August, and spreadsheet and graphics, out next month.

The first four ROMs are £39.95 each. Cost of the other four had not yet been decided.

Micro Power boss Bob Simpson says he will have signed up 500 computer dealers to act as Super Power advice centres by mid-autumn.

THE THINGS THAT STRINGS ARE MADE OF...

PETE BIBBY looks at string variables in Part 8 of our series for beginners

THIS month we're going to be taking a look at string variables and exploring some of the Basic commands used to create and manipulate them.

You'll remember that string variables are the ones that end in the dollar sign, $. They hold groups of letters, numbers, punctuation marks and spaces, all lumped together as one.

To be slightly formal, we can store the word CATS in the string variable *moggy$* using the following assignment statement:

```
LET moggy$="CATS"
```

After this, a quick

```
PRINT moggy$
```

will result in

```
CATS
```

appearing onscreen.

Of course, we don't need the LET, but we do need the inverted commas. These are the delimiters, the things that mark the beginning and the end of the string. Try entering:

```
moggy$=CATS
```

and see what you get.

You can have numbers making up the strings as:

```
figure$="1234"
PRINT figure$
```

and

```
number$="0567"
PRINT number$
```

shows. However, these are string variables and you can't do arithmetic with them. Try:

```
PRINT number$+figure$
```

or

```
PRINT figure$-number$
```

and you'll see what I mean. The type mismatch error message means just that. You've used the wrong type of variables.

Having said that you can't do maths with strings, there is a way you can manipulate them that looks uncannily like addition. This is where you join or "concatenate" two strings together. To show what I mean, set up two strings variables such as:

```
first$="first"
second$="second"
```

and then enter:

```
join$=first$+second$
```

The plus sign looks like we're doing an addition but how do you add first to second? What actually happens is that the two are joined together, as you'll see if you enter:

```
PRINT join$
```

and get:

```
firstsecond
```

in reply. This shows that the two strings have been strung together to make a third string.

Try:

```
PRINT figure$+numbers
```

and see the result. The answer to this "sum" is greatly different from the answer to:

```
PRINT 1234 + 0567
```

The first answer is a concatenated string, the second a number. It's just the same as the difference between:

```
PRINT 2+2
```

and

```
PRINT "2"+"2"
```

Remember, where string variables are concerned, the plus sign means "join together", not "add".

String concatenation lies behind last time's Program IX, this month's Program I.

```
10 REM PROGRAM I
20 CLS
30 asterisk$="*"
40 FOR row=1 TO 10
50 LOCATE 11-row,row
60 PRINT asterisk$
70 asterisk$=asterisk$+"*"
80 NEXT row
```

Program I

Take a close look at line 70. This takes the string variable *asterisk$* and adds – or, rather, concatenates – an asterisk to it. The resulting string is stored back in *asterisk$,* now one character longer.

As the FOR . . . NEXT loop cycles, so *asterisk$* grows in length, resulting in the triangle of asterisks.

Let's now go back to our original string. If you've reset your micro, recreate the variable with:

```
moggy$="CATS"
```

Have you noticed that when we used:

```
PRINT moggy$
```

we got the answer CATS and not "CATS" as we might have expected? The quotation marks have disappeared. The point is that the inverted commas are there to mark the ends of the string, not to be part of the string itself.

What if we had wanted them to appear? Could we do it by putting the whole thing in inverted commas? Try it and see. Unless your Amstrad's very different from mine, I think that you'll find that:

```
moggy$=""CATS""
```

results in a syntax error message. Don't despair though, there is a way of doing it making use of Basic's CHR$ function. But before we can do this we have to learn about something called the Ascii code.

As you probably know, your Amstrad works by numbers. Everything it does, from flashing an angry syntax error message to attacking Earth with aliens in an arcade game is done by numbers. Even when it's dealing with words as in:

```
PRINT "CATS"
```

it does it by numbers. Every character has its own code number.

The code for A is 65, while a question mark is represented by the number 63. All the letters, numerals 0-9 and punctuation marks have their own code numbers listed in a table known as the Ascii code.

For what it's worth, Ascii – pronounced "askey" – stands for the American Standard Code for Information Interchange. Table I shows a brief summary of the more useful codes.

The full set of codes is shown in Appendix III of the User Instructions. It's not exactly good reading but browse through it sometime and get an idea of how it's laid out.

So, to recap, each character you see on the Amstrad's screen has a number that represents it. The capital letters have the Ascii codes 65 to 90. You can convert these codes to their characters using the Basic function CHR$ mentioned earlier. Try entering:

```
PRINT CHR$(65)
```

and you'll have a capital A on the screen. It'll probably come as no surprise to find that:

```
PRINT CHR$(66)
```

produces B or that:

```
PRINT CHR$(67)
```

gives C. Once you've grasped how the CHR$ function converts Ascii into alphabet you'll be able to follow such masterpieces as Program II.

```
10 REM PROGRAM II
20 PRINT CHR$(67);
30 PRINT CHR$(65);
40 PRINT CHR$(84);
50 PRINT CHR$(83)
```

Program II

I hope that you're feeling outraged by the indiscriminate use of PRINTs in the last program. We don't have to use a separate PRINT for each CHR$, we can string them all together as in:

```
PRINT CHR$(67)CHR$(65)CHR$(84)CHR$(83)
```

Now you see where the term string comes from . . .

So far, we've only used the Ascii codes in the range from 65 to 90. Program III uses a FOR . . . NEXT loop to show the characters whose codes go from 32 to 126.

Here we not only have capital letters, there are also punctuation marks, lower case letters, numbers and even a space – 32. All these are the things that strings are made of.

```
10 REM PROGRAM III
20 FOR ascii=32 TO 126
30 PRINT CHR$(ascii);" ";
40 NEXT ascii
```

Program III

So using CHR$ and the relevant Ascii code we can create any string. In fact, the Amstrad has a whole set of graphics characters available using Ascii codes. You'll see these if you change the figure at the end of line 20 to 255.

If you want to know more about these characters I refer you to *Computing with the Amstrad's* excellent series on graphics from Geoff Turner and Michael Noels.

However, for the moment, let's just look at the capital letters produced by Program IV:

```
10 REM PROGRAM IV
20 FOR capitals=65 TO 90
30 PRINT CHR$(capitals);" ";
40 NEXT capitals
```

Program IV

Each time round the FOR . . . NEXT loop, *capitals* increases in value, ranging from 65 to 90. The result is that the CHR$ of line 30 prints out

Code	Character	Code	Character	Code	Character	Code	Character	Code	Character	Code	Character
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		
48	0	64	@	80	P	96	`	112	p		

Table I: Ascii codes and their associated characters

the whole of the alphabet in turn using capital letters.

Program V does exactly the same thing but in a rather better way.

```
10 REM PROGRAM V
20 offset=64
30 FOR letter=1 TO 26
40 PRINT CHR$(offset+letter);" ";
50 NEXT letter
```

Program V

Here, after *offset* has been set to 64 in line 20, the loop control variable *letter* ranges from 1 to 26. Line 40 sees the current value of *letter* added to the value of *offset* to produce an Ascii code for the CHR$ to process.

This will range from 65, when *offset* is 1, to 90, when *offset* is 26 and so the upper case letters appear. But, if the result is the same as in Program III, why bother to rewrite it?

The answer is that I find a loop going from 1 to 26 producing the alphabet more intelligible than one going from 65 to 90 to the same end. Also, look how easy it is to produce lower case letters using the offset method.

```
10 REM PROGRAM VI
20 offset=96
30 FOR letter=1 TO 26
40 PRINT CHR$(offset+letter);" ";
50 NEXT letter
```

Program VI

Notice how little Program VI differs from Program V, yet look at the difference in output. Here, having *offset* as 96 ensures that the values CHR$ works on go from 97 to 122.

These are the Ascii codes for the lower case letters, hence the differing output. Can you modify the program to produce the numbers 0 to 9? The codes range from 48 to 57.

To save yourself the bind of looking up the Ascii code for each character, Amstrad Basic has a very useful function, the aptly named ASC. This takes a character and returns its Ascii code. So:

```
PRINT ASC("A")
```

returns 65 while:

```
PRINT ASC("a")
```

gives 97. You can use string variables inside the brackets as:

```
inside$="x"
PRINT ASC(inside$)
```

will show. Also ASC clearly differentiates between numbers and strings as shown by the differing results of:

```
PRINT ASC(7)
```

and

```
PRINT ASC("7")
```

Bear in mind that ASC only works on the first letter of a string. While it's perfectly allowable to have something like:

```
PRINT ASC("CAT")
```

you only get the code returned for the first letter. In other words,

```
PRINT ASC("XYZ")
```

gives exactly the same result as:

```
PRINT ASC("X")
```

the Y and Z being left out in the cold.

However ASC is a lot more than just a quick way of getting an Ascii code. It can be useful in mugtrapping – catching user errors – as Program VII shows.

```
10 REM PROGRAM VII
20 PRINT "Enter an uppercase letter "
;
30 INPUT entry$
40 IF ASC(entry$)<65 OR ASC(entry$)>9
0 THEN CLS:PRINT "I said an uppercase
letter":PRINT:GOTO 20
50 PRINT "Well done!"
```

Program VII

As you'll have found out if you've run it – and if you haven't, you should have – the program only accepts upper case letters. Line 40 checks the Ascii value of *entry$*. Only values in the range 65 to 90 produce the upper case alphabet, so if ASC(entry$) is below or above this value there's been an erroneous input. This is another way of saying someone's made a mistake or is trying to crash your program.

The GOTO then sends the program back to line 20 for another try. Only when the Ascii code of *entry$* is in the upper case range does the program get to the final message.

However, don't you think that Program VII might be a bit fierce? After all, someone might have put in p when they meant P. Rather than have the micro point out their error – possibly putting them off computers for life – why not have the Amstrad do it for them?

After all, it only takes an offset of 32 to allow for the 32 characters between an upper case letter and its lower case counterpart. Program VIII shows how it's done.

```
10 REM PROGRAM VIII
20 PRINT "Enter. a letter ";
30 INPUT entry$
40 ascii=ASC(entry$)
50 REM check if in letter range
60 IF ascii<65 OR ascii>122 OR (ascii
>90 AND ascii<97) THEN GOTO 20
70 REM if lower case subtract offset
80 IF ascii>90 THEN ascii=ascii-32
90 entry$=CHR$(ascii)
100 PRINT entry$
```

Program VIII

Here the Ascii value of *entry$* is held in *ascii*. Line 60 checks that *entry$* is either upper or lower case. If it isn't the mugtrap has the user trying again.

By the time the program gets to line 80, *entry$* must be one or the other. Here it's tested and, if it's lower case – a code greater than 90 – 32 is taken away to make it upper case. In effect, ASC is allowing your Amstrad to correct human errors.

However, that's just one way of solving the problem and in some Basics it's the only way you have. Locomotive Basic has two functions UPPER$ and LOWER$ that make the job a lot easier.

When applied to a string UPPER$ changes all the lower case letters to upper case ones. LOWER$, as you might guess, does exactly the opposite. Numbers and punctuation marks are left unchanged. After all, what is the capital form of 2? Try examples such as:

```
PRINT UPPER$("hijkl")
PRINT LOWER$("ABC12def")
PRINT UPPER$(pqr$£1KL")
```

and you'll soon get the grasp of them. You can use string variables inside the bracket$ if you want. Enter:

```
mixed$="aaCCffZZ990 N"
PRINT UPPER$(mixed$)
```

if you doubt me. Program IX shows UPPER$ being used in a more efficient version of Program VIII.

```
10 REM PROGRAM IX
20 PRINT "Enter a letter ";
30 INPUT entry$
40 ascii=ASC(entry$)
50 REM check if in letter range
60 IF ascii<65 OR ascii>122 OR (ascii
>90 AND ascii<97) THEN GOTO 20
70 PRINT UPPER$(entry$)
```

Program IX

As you can see, lines 70 to 100 of the old program have been replaced by one line using UPPER$. Can you alter line 70 so that only lower case letters are displayed?

Before we leave the Ascii code I want to deal briefly with the codes in the range 0 to 31. These codes are rather different from the other codes we've used so far.

All the codes in the range 32 to 126 produce output on the screen when used with CHR$( ). The codes from 0 to 31 don't display the character set but they do affect the micro. They're what are known as control codes, and that's what they do, they control the micro. Try:

```
PRINT CHR$(12)
```

and see, or rather, don't see what happens. As you'll have seen, or not, as the case may be, 12 is the control code for clearing the text screen. In effect it's the same as CLS.

Try:

```
PRINT CHR$(7)
```

and you'll hear what for tradition's sake is known as the bell.

Chapter 9 of the User Instructions gives all the control codes and their uses. Try them all and see if you can figure out what's happening.

I particularly like codes 8, 9, 10 and 11 which move the text cursor backwards, forwards, down and up one character space respectively. You can have a lot of fun with them. Try to explain what's happening with:

```
PRINT "CATS" CHR$(8)CHR$(32)
```

You can even incorporate them inside string variables by adding– or, rather, concatenating – them together just like normal strings. You can see what I mean by entering:

```
blank$="CATS"+CHR$(8)+CHR$(8)+CHR$(8)
+CHR$(8)+CHR$(32)+CHR$(32)+CHR$(32)
+CHR$(32)
```

After this, the string variable *moggy$* contains four characters, four control codes and four spaces. Now when you

```
PRINT blank$
```

you'll see nothing as the four backspaces overwrite CATS.

Don't worry too much if you don't grasp control codes straight away. Like everything else on the Amstrad, understanding comes with practice.

Just so long as you have the idea that numbers or Ascii codes can represent characters that's all you need to know for the time being.

Before we leave CHR$ entirely, do you remember our problem with "CATS"? Ascii codes come in handy here. Enter:

```
moggy$=CHR$(34)+"CATS"+CHR$(34)
```

and then:

```
PRINT moggy$
```

to get the sought-after

```
"CATS"
```

It should come as no surprise that the Ascii code for inverted commas is 34. So, a cunning use of CHR$ allows you to display characters in a way impossible from the keyboard. Similarly,

```
PRINT CHR$(123)
```

and

```
PRINT CHR$(125)
```

give the curly brackets not found on any key.

And finally, how long is a piece of string? That's not such a silly question as it might seem. As you'll find out in the next couple of months, we do cut our strings into pieces – it's known as string slicing – and it's important to know their length. Because of this Basic has the function LEN.

It's not hard to use. Suppose, for reasons I can't imagine, you wanted to find the length of the string ABC using your Amstrad. All you'd have to do is enter:

```
PRINT LEN("ABC")
```

and 3 is returned as ABC is three characters in length. It's hardly a shock, is it? More realistically, you might want to know the length of a string variable which could be changing all the time during the running of a program. Set up a string variable with:

```
yourchoice$="whatever"
```

and

```
PRINT LEN(yourchoice$)
```

will tell you the number of characters it contains.

As I said, LEN is fairly straightforward but there are a couple of special cases to watch out for. The length of a space is 1, not 0 as you might think. If you don't believe me, enter:

```
PRINT LEN(" ")
```

and see for yourself. Remember, spaces count as one character, so:

```
gap$="Hello Mum"
PRINT LEN(gap$)
```

gives the answer 9, not 8.

Another special case is that of the null string, the string that contains nothing. Set one up with:

```
null$=""
```

and find its length with:

```
PRINT LEN(null$)
```

It makes sense that the answer is 0. After all, it contains no characters.

While it may seem a bit daft having a string that contains nothing, it comes in very handy as the control condition of a WHILE . . . WEND loop when slicing strings. But more of that next time. For the moment I leave you with Program X.

```
10 REM PROGRAM X
20 entry$=""
30 WHILE LEN(entry$)<>4
40 PRINT "Enter a four letter word ";
50 INPUT entry$
60 WEND
70 PRINT entry$
```

Program X

This is just a mugtrap using LEN to ensure that words of the right length are entered.

Line 20 sets *entry$* to the null string. This isn't strictly necessary as the Amstrad assumes a string is the null string until told differently. However, it is good programming practice, making the listing more intelligible and, so, easier to debug.

Until next month I'll leave you to figure out how the rest of the program works and set this problem. The program is satisfied with 1234 but this isn't a word. Can you do anything about that?

AMSTRAD COMPUTER • USER SHOW

WIN A COMPLETE AMSTRAD SYSTEM When you book an advance ticket you enter the prize draw

Everything for the Amstrad Computer User

NOVOTEL EXHIBITION CENTRE
Saturday 5th and Sunday 6th October

The first show specifically for Amstrad computer users

The Amstrad Computer User Show at the Novotel will house the largest display of Amstrad products ever assembled under one roof. There'll be all the latest new machines, software, peripherals and services.

The show will be two days of fun, entertainment, information and bargains. We'll be organising competitions, prize draws, advice centres and lots of bargains.
If you're an Amstrad owner, or just thinking about being one, you can't afford to miss the Amstrad Computer User Show.

Opening hours

10.00am to 6pm both days.

Admission charges

On the door: Adults £3 under Sixteens £2 but you can save £1 per ticket and the queues by using the coupon below. 10% discount for parties of 10 or more.
Buses: Frequent service from Central London.
Tubes: Hammersmith Broadway (Metropolitan, District and Circle Lines).
Car Parking: Several car parks in the immediate area.

Exhibitors

If you're interested in exhibiting, stands can cost as little as £99. Just write or ring Tim Collins, at the address below and we'll send you a full exhibitors pack.
Computer Marketplace (Exhibitions) Ltd., 20, Orange Street, London WC2H 7ED Tel: (01) 930 1612

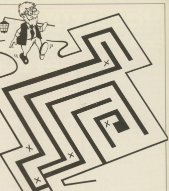
Computer Marketplace (Exhibitions) Ltd.,
20, Orange Street, London WC2H 7ED.

Please send_____Adult tickets at £2 and_____under sixteen tickets at £1 for the Amstrad User show.
I enclose cheque/P.O. to the value of £________ payable to Computer Marketplace (Exhibitions) Ltd.

Name______________________________
Address____________________________
____
______________________Postcode________

CA 9

There's more to a maze than meets the eye...



IT seems from the mail that I've received that a lot of you share my fascination with adventures. I would like to thank you all for writing in. Keep up the good work.

Most of the problems I've been asked to help with are from people who have found a maze that they can't get out of. Since mazes usually hide something worth having, and should therefore be mapped, I have decided to devote this and my next column to methods of solving them.

While I shan't be giving specific answers to the problems I've been sent, all of you who have written in will find something of use and should be able to solve more difficult mazes.

They generally fall into two categories, those needing movement to solve them and those that need thought. I shall look at those that need movement.

Most, if not all mazes have one thing in common – there's a way out. I know I'm stating the obvious, but my purpose in doing so is to make sure you realise that although it may seem impossible, you can get out if you understand the nature of it.

Remember that the programmer will have written it logically, and it is up to you to solve it in the same manner.

Let's look at some examples that illustrate the various types of maze you are likely to meet. By far the most common type are those that present this kind of room description: *"You are at a junction with exits north, south, east, west"*.

Generally to tackle these DROP an object, make a move and LOOK. If you can still see the object you have dropped then you obviously haven't moved. So get a piece of paper, the larger the better, draw a box and write inside it what object you have dropped.

Adventuring with Gandalf

If the move you just tried was NORTH, put a cross at the top of the box to show that you can't move in that direction. Then try a different direction. If the object is still there, then put another cross.

If it isn't there draw another box, DROP another object and then try another direction. Keep on doing this and eventually, by trial and error, you will be able to map out the whole maze. But if you're underground or in an unlit room, don't drop the lamp.

Incidentally, you don't have to make your map the way I have suggested. I make my maps like this because it's the way I feel happiest with. The best way to make a map is the way *you* feel happiest with.

The next most common type of maze is that in which the room description seems to be the same for every location you move to. Look closely at the following example and see if you can work out how many moves have been made. The location you are in is described thus:

"You are in a tangled, gloomy jungle with exits in all directions".
NORTH.

"You are in a tangled, gloomy jungle, with exits in all directions".
WEST.

"You are in a tangled gloomy jungle with exits in all directions".
WEST.

"You are in a tangled gloomy jungle with exits in all directions'.

Yes! You're right, two moves have actually been made. Look closely and you'll see that there are three different descriptions – check the commas! You have the description for your original location, then there is a change when you move NORTH and another when you first move WEST. The fourth description is the same as the third, so you haven't moved.

Competition time

MUCH to my shame, a number of problems have been raised this month that I can't answer, the main reason being that I haven't seen the adventures in question.

So to speed things up and, therefore, be able to give you more help, I have decided to run a small competition.

I want you to make maps and write out a full solution to the adventures you have solved. Whoever sends in the most or the best will get a copy of Peter Gerrard's book "Exploring Adventures on the Amstrad" and some software.

In the event of a tie the prize will go to the first person to send their solutions in, and so the competition will close on October 1.

Quite often with this type of maze any attempt to map it by dropping objects will result in this type of message:

"Your OBJECT disappears into the undergrowth and is irretrievably lost!"

The secret of solving this sort of maze is to make a map based on whether the room descriptions do, or do not, change. So keep going in one direction until the description remains the same.

Then try another direction until that remains the same, then another direction . . . and so on. Eventually you'll have found a way out or mapped the whole maze or both.

Quite often you'll come across a description that is totally unlike any of the others. Usually this means that there is an object of value or an exit nearby. Here it often pays to stop mapping and try single moves in each direction just to see if there's anything nearby.

Another fairly common type of maze, that can literally have you at your witt's end, is the kind where for every move you try to make you end up in the place you've just started from. Typically you get this type of comment:

"You have become completely disorientated in the warren of tunnels surrounding you and are back where you started from. There are exits north, south, east and west".

Mazes like these look very difficult, and indeed they are, but they can still be solved with a little patience. The person who has programmed the maze knows that adventurers like their problems to be hard but not impossible, so you will be able to get out reasonably easily.

There are two methods, one of which should work for you. Either a reasonable number of set moves are needed for you to get out, say six, or you only need make one move in the right direction. Often in the latter type, the move is subject to a random response. Think of it as the computer saying to itself:

"Well, they've picked the right direction, now I'll toss a coin to see if I'll let them out!"

If you're not aware that this kind of maze exists you can spend a lot of time wandering around trying to get out. Believe you me, I know. I once spent weeks in such a maze before I realised what was going on.

So try making about 10 moves in each direction. If this doesn't work then try likely combinations of moves such as NORTH, WEST, SOUTH, EAST, and so on.

Remember to use the save-game facility and make sure you have a game saved at the point at which you enter the maze. Then, if and when you die, you will be able to re-enter it. The bad news is that you have a lot of keying-in to do, the good news is that you will, eventually, get out. I promise.

Most other types of maze require you to think carefully about where you've been, what you've got and what you can see. We'll have a look at some of them next time.

Problems solved

CARLA Fowler can't open the musical door in Fantasia Diamond. *Give the conductor his property and drop the score. I don't think you need to get the picture (anyone know different?) Forget the river for the time being.*

Peter and Lesley Knowles are also having trouble here. *The black window in the dingy cellar is in need of a good clean.*

Finally, Angela White has some questions about Colossal Adventure. *Try being a paperboy if you want some reward from the gazette. Say a magic word at Y2. FEE FIE FOE FOO has a rather special effect on the eggs when said. Try it and then think under what circumstances this may be of use.*

Help wanted

CAN anybody help John Buckland who is having trouble with Forest at World's End? He wants to know how to enter the witch's hovel to get the key and whether the centaur is a red herring or not.

Rita Smith can't get past the cannibal's village in Jewels of Babylon, and Glenis Pickering, who wrote a very nice letter, can't work out how to kill the crocodile.

K. Nash has worked out how to do it but hasn't said, he's stuck at the vertical slab of rock – serves him right for not telling! Angela White wants to know whether the sailor is dead or not and what to do with him and also what to do with the octopus.

She's also having difficulty with Heroes of Karn. She wants to know the purpose of the pirate, bat, spider, phoenix and the vampire.

S.T. Hodgson wants to know how to collect the bowl and leave the room without dying in Message from Andromeda.

David Meddick wants to know how to get the sundial in Adventure Quest. Now as far as I can remember, there was no problem in taking it. Does anybody know better?

MIRRORSOFT

SEPTEMBER SIZZLERS

BOULDER DASH

One of the great classic games of all time is now available to even more of you – Amstrad and Atari owners everywhere can now experience the ultimate underground journey.

Search through the 16 caves, each with 5 levels of difficulty, and collect as many jewels as quickly as possible. Don't forget to block the amoebas, transform the butterflies, or outmanoeuvre the fireflies – or you'll never get out alive!

Available September 5th

Amstrad CPC464 and Atari 48K versions back-to-back on one tape at £9.95.

ASHKERON!

'Ashkeron is something for everyone: a complex, challenging, witty game for the adventurer, and its got pretty pics for the zap'em crowd. A excellent game – buy it!' – *Home Computing Weekly*

'It is highly professional, sophisticated software ... in its walls lies a challenge everyone will find interesting and rewarding' – *Crash Micro*

'This game gets my royal assent' – *Your Spectrum*

'A charming graphic adventure' – *MicroScope*

Available now for
Amstrad £6.95
Spectrum 48K £5.95

With built-in competition!

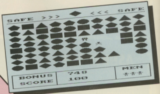
MIRRORSOFT

Maxwell House, 74 Worship Street, London EC2A 2EN
Trade orders to: Purnell Book Centre, Paulton, Bristol BS18 5LQ

The 8th Personal Computer World Show

Have you ever got yourself vaporised?

Collect the coins in GEOFF TURNER's Drogna game – but mind the Meanie or you'll end up with less than you bargained for . . .



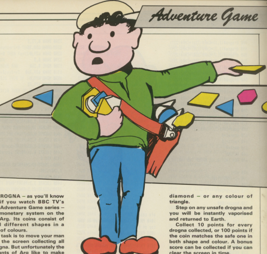
MAIN VARIABLES

h%	High score.
men%	Men left.
counter%	Drogna left.
drogna$(5,2)	Symbols for drogna.
drog%(10,8)	Shape at each position.
col%(10,8)	Colour at each position.
man$(2)	Symbols for man.
meanie$(2)	Symbols for Meanie.
score%	Score.
fast%	Speed of Meanie.
m%,n%	Position of Meanie.
x%,y%	Position of man.
bonus	Bonus score.
total%	Total score.

PROGRAM STRUCTURE

190	Initialises main variables.
670	Resets variables for new game.
820	Titles and instructions.
1240	Sets up screen display.
1510	Main game loop.
1830	Moves man.
1990	Changes safe drogna.
2120	Checks drogna status.
2190	Updates bonus.
2260	Updates score.
2320	Lose one life.
2410	Draws squares for vaporisation.
2680	Displays drogna.
2780	Game over.
2900	Adds score and bonus.
3010	Moves Meanie.

```
10 REM******************************
20 REM*      THE DROGNA GAME       *
30 REM*       Geoff  Turner        *
40 REM(c)Computing with the Amstrad
50 REM******************************
60 GOSUB 820
70 MODE 0
80 INK 2,2:INK 7,0:INK 9,13
90 INK 5,18:INK 12,0
100 BORDER 0:PAPER 7
110 CLS
120 h%=0
130 GOSUB 190:REM INITIALISE GROUP ON
E VARIABLES
140 GOSUB 670:REM INITIALISE GROUP TW
O VARIABLES
150 GOSUB 1240:REM SET UP SCREEN
```



DROGNA – as you'll know if you watch BBC TV's Adventure Game series – is the monetary system on the planet Arg. Its coins consist of several different shapes in a variety of colours.

Your task is to move your man around the screen collecting all the drogna. But unfortunately the inhabitants of Arg like to make things difficult for Earth people, and they only allow safe drogna to be collected.

The safe drogna is changed at random intervals, and is displayed at the top of the screen. It is only possible to collect coins of the safe shape or colour. For example, if a green triangle is displayed, it's safe to collect any green shape – such as a green diamond – or any colour of triangle.

Step on any unsafe drogna and you will be instantly vaporised and returned to Earth.

Collect 10 points for every drogna collected, or 100 points if the coin matches the safe one in both shape and colour. A bonus score can be collected if you can clear the screen in time.

By the way, watch out for the Meanie who moves across the screen from the left. If you collide with him, it's instant vaporisation, so don't hang around the left-hand edge of the screen.

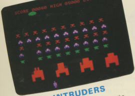
```
160 GOSUB 1510:REM PLAY GAME
170 IF men%=0 THEN GOSUB 2780:CLEAR:G
OTO 130
180 IF counter%=0 THEN GOSUB 2900:GOT
O 140
190 REM INITIALISE GROUP ONE VARIABLE
S
200 DIM drogna$(5,2),drog%(10,10),col
%(10,10),man$(2),meanie$(2)
210 SYMBOL AFTER 226
220 SYMBOL 226,0,127,127,127,127,127,
127,127
230 SYMBOL 227,0,254,254,254,254,254,
254,254
240 SYMBOL 228,127,127,127,127,127,12
7,127,0
250 SYMBOL 229,254,254,254,254,254,25
4,254,0
260 SYMBOL 230,0,1,3,7,15,31,63,127
270 SYMBOL 231,0,128,192,224,240,248,
252,254
280 SYMBOL 232,127,63,31,15,7,3,1,0
290 SYMBOL 233,254,252,248,240,224,19
2,128,0
300 SYMBOL 234,127,63,63,31,31,15,15,
0
310 SYMBOL 235,254,252,252,248,248,24
0,240,0
320 SYMBOL 236,0,3,15,31,63,63,127,12
7
330 SYMBOL 237,0,192,240,248,252,252,
254,254
340 SYMBOL 238,127,127,63,63,31,15,3,
0
350 SYMBOL 239,254,254,252,252,248,24
0,192,0
360 SYMBOL 240,0,1,1,3,3,7,7,15
370 SYMBOL 241,0,128,128,192,192,224,
224,240
380 SYMBOL 242,15,31,31,63,63,127,127
,0
390 SYMBOL 243,240,248,248,252,252,25
4,254,0
400 SYMBOL 244,0,0,0,0,3,3,1,15
410 SYMBOL 245,0,0,0,0,128,128,0,224
420 SYMBOL 246,1,2,4,8,0,0,0,0
430 SYMBOL 247,0,128,64,32,0,0,0,0
440 SYMBOL 249,0,0,0,0,0,7,5,15
450 SYMBOL 250,0,0,0,0,0,224,160,240
460 SYMBOL 251,6,7,2,2,6,0,0,0
470 SYMBOL 252,96,224,64,64,96,0,0,0
```

```
480 drogna$(1,1)=CHR$(226)+CHR$(227)
490 drogna$(1,2)=CHR$(228)+CHR$(229)
500 drogna$(2,1)=CHR$(230)+CHR$(231)
510 drogna$(2,2)=CHR$(232)+CHR$(233)
520 drogna$(3,1)=CHR$(230)+CHR$(231)
530 drogna$(3,2)=CHR$(234)+CHR$(235)
540 drogna$(4,1)=CHR$(236)+CHR$(237)
550 drogna$(4,2)=CHR$(238)+CHR$(239)
560 drogna$(5,1)=CHR$(240)+CHR$(241)
570 drogna$(5,2)=CHR$(242)+CHR$(243)
580 man$(1)=CHR$(244)+CHR$(245)
590 man$(2)=CHR$(246)+CHR$(247)
600 meanie$(1)=CHR$(249)+CHR$(250)
610 meanie$(2)=CHR$(251)+CHR$(252)
620 gap$=CHR$(32)+CHR$(32)
630 score%=0
640 fast%=10
650 men%=4
660 RETURN
670 REM INITIALISE GROUP TWO VARIABLE
S
680 a%=1
690 n%=(RND(1)*6)+
700 mean%=0
710 counter%=79
720 finish%=0
730 bonus=1000
740 total%=0
750 FOR i%=1 TO 10
760 FOR j%=1 TO 10
770 drog%(i%,j%)=INT(RND(1)*5)+1
780 col%(i%,j%)=INT(RND(1)*5)+1
790 NEXT
800 NEXT
810 RETURN
820 REM TITLES
830 PAPER 0:PEN 1
840 MODE 0
850 LOCATE 3,7
860 PRINT"THE DROGNA GAME"
870 LOCATE 9,13
880 PRINT"by"
890 LOCATE 4,15
900 PRINT"Geoff  Turner"
910 LOCATE 5,24
920 PRINT"PRESS A KEY"
930 WHILE INKEY$="":WEND
940 CLS
950 MODE 1
960 PEN 3
970 PRINT"************ INSTRUCTIONS *
*************";
980 PEN 1
990 PRINT
1000 PRINT"Collect all the DROGNA by
moving your"
1010 PRINT"man around the screen usin
g Z,X,] and \ keys"
1020 PRINT
1030 PRINT"Only collect DROGNA of a s
```

```
afe colour OR"
1040 PRINT"shape as indicated at the
top of the"
1050 PRINT"screen"
1060 PRINT
1070 PRINT"Step on an unsafe DROGNA a
nd you will"
1080 PRINT"be vaporised"
1090 PRINT
1100 PRINT"Score 10 points per DROGNA
 collected"
1110 PRINT"or 100 points for one whic
h matches"
1120 PRINT"the safe DROGNA in shape A
ND colour"
1130 PRINT
1140 PRINT"A bonus score is given for
 every screen"
1150 PRINT"cleared"
1160 PRINT
1170 PRINT"Avoid the MEANIE which mov
es across the"
1180 PRINT"screen or you will be vapo
rised"
1190 PRINT:PRINT
1200 PEN 3
1210 PRINT"************** PRESS A KEY
 *************";
1220 WHILE INKEY$="":WEND
1230 RETURN
1240 REM SET UP SCREEN
1250 MOVE 0,0
1260 DRAW 0,72,2
1270 DRAW 639,72
1280 DRAW 639,0
1290 DRAW 0,0
1300 MOVE 0,360
1310 DRAW 639,360
1320 MOVE 479,0
1330 DRAW 479,72
1340 MOVE 204,0
1350 DRAW 204,72
1360 GOSUB 2680
1370 PEN 1
1380 LOCATE 2,22:PRINT"BONUS"
1390 LOCATE 2,24:PRINT"SCORE"
1400 LOCATE 17,22:PRINT"MEN"
1410 LOCATE 8,22:PRINT SPC(7)
1420 LOCATE 8,24:PRINT SPC(7)
1430 LOCATE 8,22:PRINT bonus
1440 LOCATE 8,24:PRINT score%
1450 PEN 4
1460 LOCATE 17,24:PRINT STRING$(men%-
1,CHR$(248))
1470 PEN 14
1480 LOCATE 1,2:PRINT"SAFE >>>"
1490 LOCATE 13,2:PRINT"<<< SAFE"
1500 RETURN
1510 REM PLAY GAME
1520 colour%=INT(RND(1)*5)+1
1530 shape%=INT(RND(1)*5)+1
1540 PEN colour%
1550 LOCATE 10,1
1560 PRINT drogna$(shape%,1);
1570 LOCATE 10,2
1580 PRINT drogna$(shape%,2);
1590 PEN 4
1600 x%=10
1610 y%=8
1620 drog%(x%,y%)=0
1630 col%(x%,y%)=0
1640 PEN 4
1650 LOCATE x%*2-1,y%*2+2:PRINT man$(
1);
1660 LOCATE x%*2-1,y%*2+3:PRINT man$(
2);
1670 PEN 9
1680 PRINT CHR$(22);CHR$(1);
1690 LOCATE a%*2-1,n%*2+2
1700 PRINT meanie$(1)
1710 LOCATE a%*2-1,n%*2+3
1720 PRINT meanie$(2)
1730 PRINT CHR$(22);CHR$(0);
1740 WHILE men%>0 AND counter%>0
1750 mean%=mean%+1:IF mean%=fast% THE
N GOSUB 3010:mean%=0
1760 GOSUB 1830:REM MOVE MAN
1770 REM  delay=5:GOSUB 1840
1780 GOSUB 2190:REM UPDATE BONUS
1790 GOSUB 1990:REM CHANGE SAFE DROGN
A
1800 WEND
```

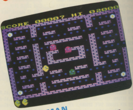
```
1810 IF counter%=0 THEN fast%=fast%-2
:IF fast%=0 THEN fast%=2
1820 RETURN
1830 REM MOVE MAN
1840 xx%=x%:yy%=y%
1850 IF x%>1 THEN IF INKEY(71)=0 THEN
 x%=x%-1
1860 IF x%<10 THEN IF INKEY(63)=0 THE
N x%=x%+1
1870 IF y%>1 THEN IF INKEY(19)=0 THEN
 y%=y%-1
1880 IF y%<8 THEN IF INKEY(22)=0 THEN
 y%=y%+1
1890 IF xx%=x% AND yy%=y% THEN RETURN
1900 LOCATE xx%*2-1,yy%*2+2:PRINT gap
$;
1910 LOCATE xx%*2-1,yy%*2+3:PRINT gap
$;
1920 PEN 4
1930 LOCATE x%*2-1,y%*2+2:PRINT man$(
1);
1940 LOCATE x%*2-1,y%*2+3:PRINT man$(
2);
1950 IF m%=x% AND n%=y% THEN GOSUB 23
20:m%=m%+1:IF m%=10 THEN m%=1
1960 SOUND 2,500,5,7
1970 GOSUB 2120:REM CHECK DROGNA
1980 RETURN
1990 REM CHANGE SAFE DROGNA
2000 count%=count%+1
2010 IF count%<15+ RND(1)*50 THEN RET
URN
2020 count%=0
2030 SOUND 1,200,20,7:SOUND 1,400,10,
5
2040 FOR DELAY=0 TO 100:NEXT
2050 colour%=INT(RND(1)*5)+1
2060 shape%=INT(RND(1)*5)+1
2070 PEN colour%
2080 LOCATE 10,1:PRINT drogna$(shape%
,1)
2090 LOCATE 10,2:PRINT drogna$(shape%
,2)
2100 PEN 7
2110 RETURN
2120 REM CHECK DROGNA STATUS
2130 IF col%(x%,y%)=colour% OR  drog%
(x%,y%)=shape% THEN GOSUB 2260:GOTO 2
170
2140 IF col%(x%,y%)=0 THEN RETURN
2150 GOSUB 2320
2160 counter%=counter%-1
2170 col%(x%,y%)=0:drog%(x%,y%)=0
2180 RETURN
2190 REM UPDATE BONUS
2200 PEN 1
2210 bonus=bonus-1
2220 IF bonus<0 THEN bonus=0
2230 LOCATE 8,22:PRINT SPC(5)
2240 LOCATE 8,22:PRINT bonus
2250 RETURN
2260 REM UPDATE SCORE
2270 PEN 1
2280 IF col%(x%,y%)=colour% AND drog%
(x%,y%)=shape% THEN score%=score%+100
 ELSE score%=score%+10
2290 LOCATE 8,24:PRINT score%
2300 counter%=counter%-1
2310 RETURN
2320 REM LOSE ONE LIFE
2330 PEN 4
2340 men%=men%-1
2350 GOSUB 2410
2360 IF men%=0 THEN RETURN
2370 PEN 4
2380 LOCATE 17,24:PRINT SPC(3);
2390 LOCATE 17,24:PRINT STRING$(men%-
1,CHR$(248))
2400 RETURN
2410 REM DRAW SQUARES
2420 ORIGIN 0,0,0,639,80,350
2430 a%=(x%-1)*64+16
2440 b%=377-(y%*32)
2450 c%=32:d%=8
2460 WHILE c%<750
2470 MOVE a%,b%
2480 DRAW a%+c%,b%,RND(1)*15
2490 DRAW a%+c%,b%-d%
2500 DRAW a%,b%-d%
2510 DRAW a%,b%
2520 a%=a%-8
2530 b%=b%+2
2540 c%=c%+16
2550 d%=d%+4
2560 SOUND 1,10,10,7,0,0,10
2570 WEND
2580 datum=TIME
2590 WHILE TIME-datum<600
2600 WEND
2610 CLG 7
2620 ORIGIN 0,0,0,0,639,399
2630 IF men%<>0 THEN GOSUB 2680
2640 PEN 4
2650 IF men%<>0 THEN LOCATE x%*2-1,y%
*2+2:PRINT man$(1);
2660 IF men%<>0 THEN LOCATE x%*2-1,y%
*2+3:PRINT man$(2);
2670 RETURN
2680 REM DISPLAY DROGNA
2690 FOR i%=1 TO 10
2700 FOR j%=1 TO 8
2710 PEN col%(i%,j%)
2720 LOCATE (i%*2)-1,(j%*2)+2:PRINT d
rogna$(drog%(i%,j%),1);
2730 LOCATE (i%*2)-1,(j%*2)+3:PRINT d
rogna$(drog%(i%,j%),2);
2740 NEXT
2750 NEXT
2760 RETURN
2770 RETURN
2780 REM GAME OVER
2790 IF score%>h% THEN h%=score%
2800 CLG 7
2810 PEN 6
2820 LOCATE 6,8:PRINT "GAME OVER"
2830 LOCATE 2,10:PRINT "YOUR SCORE ";
SCORE%
2840 LOCATE 2,12:PRINT "HIGH SCORE ";
h%
2850 LOCATE 2,14:PRINT "PLAY AGAIN (y
/n) ?"
2860 i$=INKEY$
2870 IF i$="y" OR i$="Y" THEN RETURN
2880 IF i$="n" OR i$="N" THEN CLS:MOD
E 1:END
2890 GOTO 2860
2900 REM ADD BONUS TO SCORE
2910 IF bonus=0 THEN RETURN
2920 WHILE bonus>0
2930 score%=score%+1
2940 bonus=bonus-1
2950 LOCATE 8,24:PRINT score%
2960 LOCATE 8,22:PRINT SPC(4)
2970 LOCATE 8,22:PRINT bonus
2980 SOUND 1,bonus,5,7
2990 WEND
3000 RETURN
3010 REM MOVE MEANIE
3020 PRINT CHR$(22);CHR$(1);
3030 mm%=m%:nn%=n%
3040 m%=m%+1:IF m%=11 THEN m%=1:n%=y%
3050 PEN col%(mm%,nn%)
3060 IF col%(mm%,nn%)=0 THEN PEN 7
3070 LOCATE mm%*2-1,nn%*2+2
3080 PRINT meanie$(1)
3090 LOCATE mm%*2-1,nn%*2+3
3100 PRINT meanie$(2)
3110 PEN 9
3120 LOCATE m%*2-1,n%*2+2
3130 PRINT meanie$(1)
3140 LOCATE m%*2-1,n%*2+3
3150 PRINT meanie$(2)
3160 PRINT CHR$(22);CHR$(0);
3170 IF m%=x% AND n%=y% THEN GOSUB 23
20:m%=m%+1:IF m%=11 THEN m%=1
3180 GOSUB 2770
3190 SOUND 4,50,10,5
3200 RETURN
```

4 GREAT GAMES FOR THE PRICE OF ONE!


SCORE 00060 HIGH 05000 LIVES 3

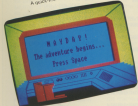
ALIEN INTRUDERS

With only your laser for protection, destroy the waves of aliens who threaten to engulf you. A quick-fire version of an all-time great!


SCORE 00007 HI 02000

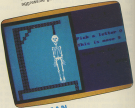
SNAPMAN

Steer your man around the maze, gobbling up energy pellets while keeping away from the aggressive ghosts who are out to get you.


MAYDAY!
The adventure begins...
Press Space

MAYDAY

Guide the sole survivor of a stricken spaceship through the wreckage of his craft. Recover vital medical supplies . . . or a planet is doomed!


Pick a letter
This is move 5

DEADMAN

The time honoured game of suspense – but with some novel endings. Guess the hidden letters or something nasty could happen to you.

RE FOR THE AMSTRAD NOW PRESENT -

UTOPIA £29.95

BASIC UTILITIES ROM
(Prod Code EU)

40K USER RAM

Beebug's 'TOOLKIT' is the standard utilities ROM for the BBC Micro and has sold thousands of copies. Now the author has written an Amstrad version. Available only on ROM the program contains numerous Basic Programming AIDS including search/replace within Basic program, listing basic variables, moving basic lines, load, save, verify, type, dump, format, copy and much more.
ALL INCL PRICE £29.95

AND GREAT NEWS FOR MACHINE-CODE ENTHUSIASTS:

MAXAM

IN CARTRIDGE **NOW ONLY** **£49.95 (incl VAT, p&p)**
NOW ALSO AVAILABLE IN **ROM ALONE** FOR ONLY £39.95
ALL VERSIONS NOW CONTAIN **FULL** SPECIFICATION
DISC £26.95 TAPE £19.95

PRESS COMMENT

"Innovative device this article finished on the ARNOR editor...... well worth the money!" - **AMSTRAD COMPUTER USER**

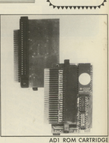
"Assemblers......look no further ARNOR is the best I have seen"
- COMPUTING W.T. AMSTRAD

"Absolute magic! - ARNOR must be the market leaders"
- POPULAR COMPUTING WEEKLY

"Quite Special......................difficult to match"
- COMPUTING TODAY

"ARNOR are to be congratulated on a superb job.....definitely the best"
- HOME COMPUTING WEEKLY

"A product no serious AMSTRAD user can afford to be without"
- YOUR COMPUTER



AD1 ROM CARTRIDGE

IN GOOD COMPUTER STORES EVERYWHERE - OR DIRECT.......

PLEASE SEND ME
(PRODUCT CODE) (PRICE)
__________ __________
__________ __________
__________ __________

Total £__________

I enclose Cheque/PO for £__________
(PAYABLE ARNOR LTD)

OR
Please debit my Access/Visa

Card No □□□□□□□□□□□□□□□□

NAME: __________ SIGNATURE: __________
ADDRESS __________

CWA9

ALL TRADE/CREDIT CARD SALES, ENQUIRIES ETC 01-688-6223
SEND TO ARNOR Ltd Dept.□, THE STUDIO, LEDBURY PLACE, CROYDON, SURREY CRO 1ET

pride utilities LTD.

SYSTEM X — 32 new extended Basic commands.

Yes, at last you can upgrade your 464 or 664 with this massive range of commands whilst using approximately 3K of memory. Take some of the hard work out of your programming, use them in your programs and take pleasure in computing with your much enhanced Amstrad — even gives your computer a Circle and Fill command. **NEW COMMANDS:**

I FRAME	I RSCREEN	I LSCREEN	I USCREEN
I DSCREEN	I CIRCLE	I FILL	I INVIS
I VIS	I GETKEY	I CURON	I CUROFF
I R	I RESET	I FLUSH	I PROTEC
I DEPRO	I CAPON	I CAPOFF	I GPEN
I HELP	I GETCHAR	I PRON	I PROFF
I RPEEK	I DPEEK	I DPOKE	I INVERSE
I GOVER	I ZIP	I MOTOR	I SHIFT

Cassette £9.95 inc. Disc £13.94 inc. (464 and 664)

TRANSMAT
Easy Tape to Disc Transfer

Offers easy transfer of **tape based software** onto **disc.** For Amstrad DDI-1 disc drive system. Nothing available on the market can match TRANSMAT'S capabilities and versatility. Beware of poor copies. Offers: Auto or non auto modes, removes protection from BASIC, comprehensive disc header reader, adds relocator if necessary, easy erase or rename programs. ***"This is a very easy to use utility"*** – *Personal Computing Today, July 1985.*

Cassette £7.95 inc. Disc £11.95 inc. (464 and 664)

TOMCAT
The Headerless Backup Copier

Achieve the ultimate in **tape to tape backup copying** with this no nonsense program which handles practically everything, even headerless, usually in one go. If you require to load and list protected software etc. then you require SYCLONE, the versatile tape to tape utility. If all you need are backup copies of your software which load in up to 4 times the norm, then you must have TOMCAT. There is probably nothing on the market that can anywhere match TOMCAT'S performance and easy operation. ***"Second best thing to disc drive".***

Cassette £6.95 inc. Disc £10.95 inc. (464 and 664)

AMSTRAD 464/664

QUALITY UTILITIES FROM OUR FAST, RELIABLE AND FRIENDLY MAIL-ORDER SERVICE. ALL ENQUIRIES WITH S.A.E. ANSWERED. TRADE AND OVERSEAS ORDERS WELCOME.

RSX SYCLONE 2
Tape to Tape Utility

A RSX Tape to Tape utility which adds brand new commands, offering many facilities, for your CPC 464. Allows you to backup your valuable software to load in up to one quarter of the time. Choice of seven saving speeds, 1000 to 4000 baud. SYCLONE not required to load back high speed copies. Comprehensive header reader. Commands available from BASIC, load and list protected BASIC programs. List your WELCOME tape. Disc compatible. ***"Not only a reliable backup copier, but also a very useful tape to tape tool kit".***

Cassette £6.95 inc. Disc £10.95 inc. (464 only)

RSX PRINTER PAC-1
Printer Utility for DMP1 and EPSON Compatible Printers

A resident system extension which adds SIX new commands to your CPC 464/664. A two program pack which caters for the **AMSTRAD DMP1 printer** (AMDUMP) and **EPSON compatible printers** (EPDUMP). Screen dump in all modes. Two sizes with EPDUMP. Text dump in all modes. Specify background inks. Abbreviated codes to printer. Three new type styles for the DMP1. Can be used with AMSWORD. ***"A very useful utility, frequently used".***

Cassette £5.95 inc. Disc £9.95 inc. (464 and 664 — 664 version on disc only)

SCRIPTOR
Definable Character Fonts for the DMP1

A must for **DMP-1 printer** owners. Offers six definable character fonts. Allows your DMP-1 to perform — don't upgrade. Offers: true lower case descenders, joined writing, futuristic, bold and heavy, italics, character definer program, Amsword compatibility. Professionally finish your letters. ***"Tired of the standard type-style on the DMP-1 Printer? Then Scriptor is for you."*** *- PCT July, 1985*

Cassette £6.95 inc. Disc £10.95 inc. (464 and 664)

ZEDIS II
Comprehensive but friendly Editor and Disassembler

A comprehensive yet friendly machine code editor and disassembler. Indispensible to the novice and expert. Offers full Z80 disassembly, continuous menu display, break point insertion and register inspection, Hex/character input, register preload before execution, fast Hex/character search, output to printer. Full instructions to disassemble ROMs. ***"Very easy to use".***

Cassette £6.95 inc. Disc £10.95 inc. (464 and 664)

SPECIAL OFFER
WORTH £3.95

Buy two or more titles and receive, completely free, a real time Digital Alarm Clock program on cassette.

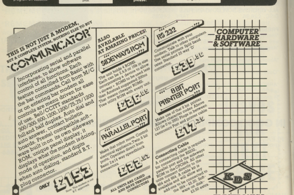
PRIDE PROMISE

We are so confident in our professionally manufactured and packaged software that we dare ask you to show us a cheaper comparable commercial program that can match the high standards of any one of our products and we will refund the difference in price.

CUSTOMER ENQUIRIES
TEL: 0582 411686
between 9.00-10.00 am please.

PRIDE UTILITIES LTD.
(Dept. CWA)
7 Chalton Heights
Chalton, Luton,
Beds. LU4 9UF
England

ALL PRICES INCLUDE VAT AND P & P
For Europe add £1.00 per title
For Australasia add £1.75 per title
For Rest of World add £1.50 per title



THIS IS NOT JUST A MODEM, BUT A COMPLETE SYSTEM, NOTHING ELSE TO BUY

'COMMUNICATOR'

Incorporating serial and parallel interfaces, to allow software control of all functions. Each feature controlled from Basic with the bar commands. Call from M/C or on entering bar modem all controls are menu driven for ease of use. Bell/CCITT standards 300/300, 600, 1200, 1200/75, 75/1200 full and half duplex. Auto dial and auto answer, contact bulletin boards. Prestel compatible, software built in on its own sideways ROM, unique panel display, it displays what the modem is doing, mode of operation, and digits when auto dialing, standard B.T. plug connector.

ONLY £153

Note: This modem is not B.T. approved.

ALSO AVAILABLE AT AMAZING PRICES!

SIDEWAYS ROM

The unit holds 4 ROMS. Each can be 2,4,8 or 16k in size incorporating a device to allow + slower + ROMS to be used less than Amstrad suggested 200ns That means cheaper ROMS. Free utility ROM with every unit.

£26.95

PARALLEL PORT

Make that robot move. Control electrical appliance. Twin 8 bit ports. Operates direct from Basic 2x14 way speedblock connectors.

£22.95

ALL UNITS ARE CASED AND HAVE THROUGH CONNECTORS

RS 232

Communicate with your modem. Talk to other computers. Use serial printers Split baud rates Standard 25 way 'D' connector.

£39.95

8 BIT PRINTER PORT

Make use of that 8 bit printer Allows character codes Above 127 (ie 0 to 255) Plugs in between centronics Port and printer cable

£17.95

Connecting Cable

A connecting cable is required for connecting the RS232, parallel port and the sideways ROM to the CPC664. All other units connect directly to the 464 and 664. A cable kit is available and will allow any add-ons to be connected to the 664 – £5.20. Software on disk-£3.50 extra.

COMPUTER HARDWARE & SOFTWARE

KDS ELECTRONICS

15 HILL STREET, HUNSTANTON, NORFOLK PE36 5BS. TEL:(04853)2076

Please add VAT to all prices.

Willie makes whoopee

HAVING converted the very successful Manic Miner to run on the Amstrad, Software Projects' equally successful **Jet Set Willy** was not far behind.

Once out of the mine, Miner Willy wasted no time at all in spending his new found wealth. Having bought a mansion and yacht he decides to throw a party.

It is a wild success but the housekeeper is rather upset about the aftermath. Willy is given an ultimatum – no sleep until all the debris has been cleared.

When you live in a 100 room mansion that is some headache.

In Manic Miner each individual screen had to be completed before starting the next one. Jet Set Willy has no such constraints and you can wander from screen to screen at will.

However you will only be awarded points for collecting the objects from the party.

When you load for the first time you will see that you are provided with eight Willies.

Wow, I thought, now here is a game that will last all night. I couldn't have been more mistaken.

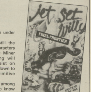
Should you manoeuvre Willy into a position that means certain death, he dies (obviously!) He is then reincarnated in the same position and unless you move away quickly he will die again, and again, and you'll soon be minus several Willies.

In one particular instance I lost all eight Willies in under three seconds.

The graphics are still the same old Spectrum characters as used in the Manic Miner conversion. How long will software houses insist on bringing the Amstrad down to the level of more primitive machines?

Green screen users among you will be pleased to know that a black and white option is available which provides better contrast.

On several occasions I entered the bedroom only to be shown the way out by the irate housekeeper. What I can't understand is that if Willy can afford a 100 bedroom mansion why can't he get himself some decent staff?



James Riddell

It's fun in the dungeon

DUNGEON Adventure is the final and best part of Level 9's Middle Earth trilogy.

The plot carries on where Adventure Quest left off, but your role changes. Now you play the greedy adventurer who, on hearing of the fall of Agaliarept, decides to attempt to loot the evil lord's desmenes.

About a mile from the dark tower you are hit by a sleep spell. You awake on a sandbank beneath a bridge. All your possessions have been stolen and the light is fading.

The first problem you are going to notice is that you only have a limited number of moves before it gets dark. Once this happens you only have a couple of moves before evil ghouls rise up and claim your soul.

The secret is to search the area thoroughly and then plan out your moves so that you can collect everything and reach the other side of the bridge before it gets dark.

You should ARM yourself with six objects before tackling the tree on the island. It might be a juggler but there is a limit to the number of things it can handle!

The first of your six objects is easily found and very noisy if used properly.

You will find you need to use it twice, once for a valuable treasure and again for something that, on reflection, should ensure you stay awake.

The masked man will leave you dice, coins and a wand. The tree might at last fall, you think, but heed the dryad's plea.

Now your problem is to find a way of carrying all your illgotten loot. Then you remember the packing case that was beside you when you awoke.

So it's back to the river and once you get IN, your hands are no longer full. That strange machine in the corner is used to resurrect you when you die. But you have to have your soul strengthened first.

A windswept hill provides the answer if you are patient and Lady Luck is on your side.

Now you are ready to cross the bridge and brave the evil lord's dungeons. However, they are underground and you have no light.

A quick trip back to the river provides the answer but by this time you will probably have been killed a dozen times. You should now know what to do and a careful plan will see you accomplish everything just as the sun sets.

The first thing you see in the caves would appear to solve all your desires. But beware, all that glisters . . .

The round room will offer you a choice of directions. Moving in a roughly clockwise direction will solve some immediate problems 'but I hope you save the game before getting too curious.

It is here that the first of the renowned Level 9 puns appears – the cube of metal and the hay.

And as for the octopus, well, corny just isn't in it.

I'll leave you to fend for yourself at this point. Assuming I haven't missed anything, you should find that you have mapped just over 20 locations. Only 200 more to go.

A couple of tips – from the round room you are going to have to do a lot of mapping, so use a fresh sheet of paper.

And remember what the

staff is made of. Keep the wand handy. It is a boring thing and likely to send someone to sleep.

In many ways Dungeon Adventure is a reviewer's nightmare. I've spent two weeks on it and I'm only halfway through.

One thing that came as a surprise was that GET isn't recognised. However TAKE is, and can be abbreviated to T.

I have yet to encounter an adventure with as much scope, complexity and atmosphere as Dungeon Adventure. It is the best I have ever played.

Paul Gardner

I'm chess bored...

I THINK Deep Thought **Superchess** from CP Software must be a conversion from an old Z80 machine code program.

It plays a dull, solid game, but lacks a number of almost-standard trimmings such as a clock, underpromotion, detection of repetition and solving two and three move mate problems.

Actually, anyone can test a chess program – first try the King, Queen versus King ending.

The program manages this win if set to Tournament speed (four minutes per move) but fails at the King, Rook versus King ending – not too surprising as I have yet to see any micro program manage that one.

The program has a simple book of King and Queen Pawn openings and likes the Ruy Lopez thus:

Deep Thought	D. Alliss
1. P-K4	P-K4
2. N-KB3	N-QB3
3. B-N5	

but force a Giuoco Piano and set the Blackburne trap thus:

3. B-B4	N-Q5?

I have never seen a program refuse this offer of a pawn unless protected by an opening book so no surprise when:

4. N*P	Q-N4!

Really dumb programs (and people) play N*P forking Queen and Rook but BQ*NP threatens mate in one and White is in deep trouble. Deep Thought played:

5. B*P+	K-K2
6. P-KB4	Q*NP
7. R-B1	Q*P+
8. K-B2	Q*P+ or N*P

A slightly better program would lose the Knight immediately at move 6 (by castling) – this gives better development with only one pawn down.

An interesting feature is that the program can "Recommend" a move, but it does this too quickly – five seconds – to be recommended. The program can also be set to one of seven playing-speeds ranging from eight seconds per move (Level 0) to 24 hours (Level 6).

These levels are one fewer than the number of lookahead plies (moves by White or Black).

On average each side has roughly 30 moves and yet the time for each level only rises by about six so the program cannot be considering all possible moves unless it has perfect move ordering (in which case why bother to look ahead?).

The program uses depth first alpha-beta with killer heuristic (techniques almost 20 years old) with an extra ply for checks and end-games.

This means it was looking 5-6 plies ahead in the simple end-game tests.

This is not enough for the King, Rook ending which requires either some chess knowledge of King opposition or the 45 ply look-ahead of the super mainframes which can discover or invent almost all chess knowledge without the hindrance of "intelligence".

In conclusion: Probably a good program three or four years ago, but now one expects either a much stronger

Keep an eye on your money

MONEY Manager (Connect Systems) is quite the best and most flexible personal accounts system I've seen – but don't expect to buy it today and wear it tonight.

For the best use, you'll want to alter the sample headings to suit yourself, which needs a thorough understanding of how the program works, plus some logical thought on headings. This took me two days, and I think I'm fairly bright, accounts-wise.

In simple terms, it has three interlocking sections:

- **Accounts:** Something you put money into or pay money out from, like a bank, or building society or your wallet (cash), with up to nine headings available.
- **Classes:** Where you get your money from and what you spend it on, like salary or rent. These headings are coded as a letter plus up to nine numbers. That is, choose h for household and h1 as food, h2 as milk . . . or s1 – his salary and s2 – hers.
- **Entries:** Everything goes in here – cheques drawn, monies received, standing orders, what you spent for lunch . . . On each entry you are prompted for an account number and a classcode, and payments are entered as a minus figure. So, in entering your rates cheque, you give the bank heading reference and the rates code (they're all on screen) and the program does the necessary.

When one entry affects two accounts, like a cheque or a standing order to your savings account, you make a normal entry for the cheque, using the bank reference and transfer classcode and a minus figure. Then you duplicate the entry, using the cash account reference, the transfer classcode and a positive figure.

The "transfer" balance should always be nil because the duplicate entries cancel out, but the phone rings or the baby cries and you miss one. It's easy to see which and entering the correction out of order is no problem.

The manual says load the demo data "if you want to", but until you've created your own data it's the only entrance. And the "not accounts minded" need a simple explanation of how the program works and a little more on transfers. Otherwise explanations are explicit.

Changing headings or making, amending and deleting entries are the basic tools. The "extras" display or print out full details of any account or classcode or combination of both or all, as monthly or annual statements, and bar or pie charts.

And you can sort entries into any order or search for a name or amount with ease. A whole year's details are held with 100 monthly entries available.

This was well worth the initial time and effort and, where sources of income are few, could double as an inexpensive business system.

Dorene Cox

game for the money or more "options" such as problem solving, an indication of what it is "thinking" about while you wait, the option to force it to move after a certain time rather than just restart, and, almost essential, clocks for genuine tournament play.

Not recommended.

Dee Alliss

Here's a bright spark

ALONG with many other people, I puzzled for several weeks over the full-colour teaser adverts that preceded the release of **Confuzion,** by Incentive Software.

It is often a sad fact of life that those software companies with the worst software have the best adverts. Fortunately this was not the case with Confuzion.

The action takes place in a 64-storey industrial plant which is involved in the production of deadly confuzion bombs.

Each floor of the factory is constructed from a series of sliding panels plus one empty section.

The player manipulates these sliding sections in the same way that a child plays with a sliding block puzzle.

Each of the panels contains lengths of fuse wire of differing shapes. Wandering around this fuse wire is a brightly glowing spark.

Your task is to guide the spark to a bomb by sliding the panels in such a way that they create one continuous length of fuse wire leading to the bomb.

All of this frantic action is carried out in a race against time as the fuse at the top of the screen burns away.

The starting screen consists of a 4 x 3 grid containing one bomb and one spark. As the game progresses you move on to a 4 x 4 and then a 5 x 4 grid with multiple bombs.

Added hazards are water droplets from the factory's sprinkler system, found wandering around the fuses.

Concentrate too closely on guiding the spark and you could inadvertently transfer the water droplet on to the same path.

It is usually at these times that panic sets in and I frantically shift every block in sight, normally to no avail.

Confuzion makes full use of the Amstrad's wide range of colours to create some very pleasing visual displays.

Incentive Software have paid great attention to detail in the program. For example, all of the bombs rotate constantly with incredible realism.

Sorcery may still be the number one Amstrad game of 1985, but I can assure you that Confuzion will not be far behind.

James Riddell

Sink that U-boat!

MY Royal Navy days were spent aboard warships afloat and "stone frigates" ashore, so I don't profess to be an authority on submarine warfare.

Nevertheless, I am impressed by the apparent authenticity of **Hunter Killer.**

Protek have stuck to their torpedo tubes and resisted any urge to destroy the mood by adding fantasy embellishments.

You don't zap the aliens with lasers in this game, but stalk and sink them with a well-placed, long-running torpedo.

Your mission as the commander of a WW2 British submarine, on patrol off the Heligoland Bight, is to stop German U-boats reaching the Atlantic.

Plotting and steering interception courses to bring your vessel within range, you then line it up with a stable trim to fire towards, or on a collision course with, the surfaced enemy sub.

There are mined areas in the vicinity, plus shallows and a tidal flow to consider, along with your speed and bearing and that of the target.

Ballast tanks and hydroplanes control your depth and trim and battery charge is critical.

I found it best to close with the U-boats by using diesel motors to travel on the surface and charge up my batteries.

This increased the time I could then spend submerged – but also the depth-charge danger from patrolling aircraft.

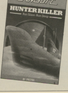
When submerging you must remember to switch over to electric motors and double check the indicator, as slightly prolonged button-pressure reverses the setting.

The three graphic screens are excellent in colour or mono and represent the control room, chartroom and a periscope view, which can be rotated through 360 degrees and used while on the surface.

If submerged, it can only be raised at the proper depth settings and sometimes bobs up and down like a yo-yo.

On the surface it is supplemented by radar, which ceases to function at depths below 10 feet, when the asdic takes over, the echo of its "ping" speeding up as the range decreases.

The scopes for these, plus numerous other instruments and gauges, are well depicted and soon become part of your shipboard routine.



There are four levels of difficulty and the comprehensive instructions are only let down by the lack of degree markings on the skimpy compass rose.

However, that is nit-picking, for on return from patrol, my Amstrad will enter harbour proudly flying the Jolly Roger as the traditional signal for a successful "kill!"

Ray Miller-Lawson

Learn assault craft...

IT'S time to get your tin hats out and lead the assault on the enemy territory in **Beach-Head,** from U.S. Gold.

This is one of the first conversions of their legendary Commodore 64 software, and it's great . . .

As the game begins, you are presented with a map of the area.

By moving the cursor which represents your forces, you can select one of two opening scenarios.

In the first you must guide your fleet through a narrow channel which is mined and is constantly criss-crossed by enemy torpedoes.

You score for every ship successfully negotiating the channel, but you inevitably lose a few ships.

The second option is to skip this section and proceed with a full complement of 10 ships to the anti-aircraft game.

This part of the game is great fun – enemy aircraft dive at your ship with guns blazing.

As they approach, they increase in size and detail.

You now take on the role of gunner, moving the gun up, down, left and right, pumping out shells at wave after wave of fighters.

Speed and accuracy are essential here, as with each hit

from the enemy your damage points increase.

Each time the total reaches 20 damage points you lose another ship.

The enemy planes defeated, their fleet now comes into play.

Once again you are a gunner, but this time the battle is ship to ship.

My first attempts at lobbing shells at silhouettes of enemy vessels were disastrous.

I then spotted a small notice on the display informing me of how long or short my last shot was – aren't computers wonderful?

Having survived this melée, the surviving ships head for the beach and land their cargo of tanks – the number of tanks landed being proportional to the number of surviving ships.

The tanks must be guided through a tortuous maze of tank traps and gun emplacements.

Finally they reach their target – a huge gun mounted on top of a hill. The hill is protected by numerous guns which must be eliminated before the large gun is destroyed.

The graphics are excellent, and the fact that your performance on each level determines your fire-power on the next is a great idea.

Let's hope we see many more U.S. Gold conversions in the future. **James Riddell**

Time to plan

Project Planner is a business package from Amsoft. It allows you, as the manual quotes, to "take control of time on any task".

If you have a need to estimate how long a certain job will take from day one to completion, when parts of the job rely on the completion of others, this package may be the thing you are looking for.

There are two cassettes and a manual provided in the package.

Considering there are programs on the two cassettes which would fill three or four Amstrads, it seems good value for money.

Project Planner uses critical path analysis and the first cassette is dedicated to teaching this concept.

The computer is used in conjunction with the manual, and within two hours you should understand the subject and be able to use the second cassette – a successful mix of book and computer teaching.

When you have mastered the first tape the second cassette, Applications program, allows you to input all the information about the various tasks that make up a project.

It then calculates all the sequences and networks and prints out the results, reports and bar graphs.

If you are in business, involved in planning and have a printer and preferably a disc drive – some of the programs take eight minutes to load on cassette – this may be the program you are looking for.

The documentation is clear and the programs work well in a business-type manner.

Lynne Savidge

This is a good draw, but...

AT first glance, CRL's **Artist and Sprite Designer** appears good value.

The main program is a graphics utility which enables you to draw and plot all manner of symbols and shapes which can be combined into sophisticated drawings.

Side two of the cassette contains a sprite designer program, together with a conversion program to assemble the data into machine code enabling the sprites to be called from within Basic programs.

The Artist program was, I found, quite easy to use, but due to the large number of keys to be used repeated reference to the instruction manual was necessary.

Unfortunately I found the manual to be a little sparse and unclear in places, which rather let the software down somewhat.

However, with a little perseverance and practice, it eventually became quite easy to use the program.

A pair of cursors are displayed on the screen, which are used as origin points for producing various shapes.

It is possible to draw lines, plot points or simply move the cursors around the screen.

The program allows a choice of three speeds for cursor movement, which is useful for producing fine or coarse detail.

A range of shapes can be plotted. Circles, ellipses, boxes and so on.

It is also possible to fill shapes either in a single colour or with a two-colour stipple effect.

There are numerous other facilities which, with a little practice, can be used to produce some interesting artwork.

I liked the delete feature, which is useful for clearing mistakes.

Included in the art program is a user defined graphic creator which allows UDGs to easily be defined. They can then be printed on to the drawing created with the art program.

As an added bonus, there are three demonstration screens on the cassette, which gives some idea of what can be achieved.

Sprite Designer, on side two, allows easy creation of multicoloured sprites which can be saved to tape and recalled for use in Basic programs.

Unfortunately I had a couple of problems with this program. Firstly, while trying to load in my sprites, a read error occurred. I could find no way out of the error situation other than resetting the computer and reloading the program.

When attempting to write a short Basic program to call the sprites, I used the Auto line number function, only to find that I couldn't break out of it because the Escape key appeared to have been disabled, necessitating a further reload.

As I said earlier, a good-value package – but unfortunately let down by a couple of design problems and a less-than-adequate set of instructions.

Geoff Turner

ACCESS & VISA ORDERS: 0227 266289

CUT OUT & KEEP THIS AD!

AMSTRAD USERS!

Having difficulty getting software for your CPC464 or 664? Then let us help you! **The Software Supplies Company** is a specialist trading division of a long-established mail-order software supplier with a worldwide customer base of thousands.

UK Postage (Europe) FREE for pre paid recreational software, and all other orders over over £20 (UK only). Otherwise 50p (£+) for orders up to £20. 75p (£1.50) for orders £30. £1 (£2) up to £50 and £1.50 (£3.00) over £50. Airmail/Freight elsewhere at cost. Heavy items marked with a dagger use the European rate.

PAYMENT: Please make cheques/P.O.s/M.O.s payable to **THE SOFTWARE SUPPLIES Co.** Or use you Access, Mastercharge or Visa cards.

NB: cheques or credit orders are not processed until goods ready for dispatch. Please read footnote:

- **Business Educational and Recreational Software for ALL home and business computers –**
- **Keen prices — excellent choice**
- **Send for lists (state machine type) or, if you want something quickly, phone your order through now!**

APPLICATIONS

Amscalc(†)	Amsoft	19.95
Amscalc Xtra(†)	Amsoft	34.95
Amsword(†)	Amsoft	d23.95
Cashbook Accounting(†)	Gemini	d65.95 59.95
Database	Gemini	d25.95 19.95
Database	Kuma	14.95
Decision Maker(*)(†)	Amsoft	d29.95 24.95
DFM Database + Labels	Dialog	14.95
Easi Amsword Word Processor	Amsoft	9.95
Easi VAT Accounts	Kuma	39.95
Entrepreneur(*)(†)	Amsoft	d29.95 29.95
Flexifriend (Home Budgeting)	Camel Micros	7.50
Home Budget	Kuma	19.95
Home Accounts Manager	Dialog	14.95
Home Accounts	Gemini	d25.95 19.95
Invostat	Dialog	29.95
Mastercalc 464(†)	Campbell	d29.95 24.95
Masterfile 464(†)	Campbell	d29.95 24.95
Masterfile Extension	Campbell	10.00
Micropen(†)	Amsoft	d49.00
MicroScript(†)	Amsoft	d49.00
MicroSpread(†)	Amsoft	d49.00
Mini Office	Database	5.95
Money Manager	Connect Systems	14.95
Project Planner(*)(†)	Amsoft	d29.95 24.95
Report Generator	Gemini	d25.95 19.95
Stock Aid	Dialog	29.95
Tasprint 464	Tasman	9.90
Tasword 464	Tasman	24.96
Transact	Dialog	29.95
VAT File(*)	Gemini	25.95 19.95
Camsoft Business Suites	Cambrian	d39.00
Nominal Ledger(†)	Cambrian	d33.00
Purchase Ledger(†)	Cambrian	d33.00
Stock Control/Invoicing/ Sales Ledger(†)		d99.00
TasCopy 464	Tasman	9.90

UTILITIES/LANGUAGES

Artist + Sprite Designer	CRL	9.95
Artwork	Kuma	9.95
Assembly Language Course(†)	Honeyfold	12.50
Azimuth 2000 (head alignment)	Interceptor Micros	8.95
BASIC Programming Course(†)	Honeyfold	10.50
Concise BASIC Spec(†)	Amsoft	19.95
Concise Firmware(†)	Amsoft	19.95
CPC464 Meddler (copier)	Nemesis	4.50
Devpac Ass/Disassembler	HiSoft	d29.95
Font 464	HiSoft	7.95
FORTH	Kuma	17.95
Graphics Designer	Amsoft	11.95
Guide to Amstrad BASIC 1(†)	Amsoft	19.95
Guide to Amstrad BASIC 2(†)	Amsoft	19.95
Guide to LOGO(†)	Amsoft	9.95
Introduction Manual(†)	Amsoft	9.95
Introduction to Sound 1	Amsoft	11.95
LOGO Turtle Graphics	Kuma	19.95
Machine Code Tutor	New Generation	14.95
Music Composer	Kuma	9.95
Operation Caretaker	Global Software	9.95
Pascal	HiSoft	d18.95 34.95
Screen Designer(†)	DJL Software	11.95
Speech Synthesiser	DK'Tronics	39.95
The Quill	Gilsoft	16.95
Zen Assembler	Kumana	19.95
ZAPP Z80 Assembler	Hewson	14.95
DevPac 80 (CP/M)(†)	Hi-Soft	d39.95
Maxam Arnor		r59.90 d26.90
		13.50
Pascal 80 (CP/M)	Hi-Soft	d39.95
Super Power Rom Card		r39.95
Super Power Mail + Club		r39.95
Disc Utilities		r39.95
Prog Tool Box		r39.95
Discassmon		r39.95

GAMES

3D Invaders	8.95
3D Star Strike	6.95
3D Time Trek	7.95
5-A-Side Soccer	8.95
Admiral Graf Spee	8.95
Adventure Quest	9.95
Alien Break-In	8.95
Alien 8	9.95
American Football	9.95
Amsgolf	d12.95 8.95
Android One	7.95
Android Two(*)	7.95
Animated Strip Poker(*)	8.95
Archon(*)	10.95
Atom Smasher	8.95
8 Jacks Superstar Challenge(*)	7.95
Backgammon	8.95
Battle for Midway	9.95
Blagger	7.95
Bridge Player -2	9.95
Bridge	8.95
Bridge-It	8.95
Caesars Travels	8.95
Castle Blackstar(*)	6.95
Centre Court	d12.95 8.95
Chopper Squad	6.00
Chuckie Egg	7.95
Classic Adventure	8.95
Classic Racing	8.95
Climate	8.95
Codename Mat	d12.95 8.95
Colossal Adventure	9.95
Combat Lynx	11.95
Crazy Golf	8.95
Cubit	8.95
Daley Thompsons Decathlon	8.95
Dark Star	7.95
Death Pit	7.95
Defend or Die	7.95
Detective (Cluedo)	8.95
Digger Barnes	8.95
Dragons Gold	7.95
Draughts	8.95
Dungeon Adventure	9.95
Electro Freddy	d12.95 8.95
Emerald Isle	8.95
Erbert	5.95
Eric the Viking	9.95
Everyone's a Wally(*)	9.95
Fantasia Diamond	7.95
Fantastic Voyage	8.95
Fighter Pilot	8.95
Flight Path 737	6.95
Football Manager	7.95
Forest at World's End	5.95
Fruity Frank	8.95
Galactic Plague	8.95
Galaxia	8.95
Gems of Stradus	8.95

Ghostbusters	10.99
Ghouls	7.95
Gilligans Gold	6.90
Grand Prix Driver	8.95
Hard Hat Mack(*)	10.95
Hareraiser Prelude	8.95
Hareraiser Finale	8.95
Harrier Attack	d12.95 8.95
Haunted Hedges	8.95
Heathrow ATC	7.95
Herbert's Dummy Run(*)	9.95
Heroes of Karn	5.95
Hold Fast	8.95
House of Usher	6.95
Hunchback	d12.95 6.95
Hunchback II	8.95
Hunter Killer	d12.95 8.95
Interdictor Pilot	17.95
Jammin'	d12.95 8.95
Jet Set Willy	7.95
Jetboot Jack	8.95
Jewels of Babylon	5.95
Johnny Reb	6.95
Jump Jet	9.95
Killer Gorilla Gauntlet	9.95
Knight Lore	9.95
Kong Strikes Back	8.95
Laser Warp	8.95
Lords of Time 9	9.95
Magic Sword (+ book)	8.95
Manic Miner	7.95
Master Chess	d12.95 9.95
Match Fishing(*)	9.95
Message Andromeda	5.95
Mighty Magus	6.95
Millionaire	6.95
Minder	9.95
Moon Buggy	7.95
Mr Wong's Laundry	8.95
Mutant Monty	d12.95 8.95
Oh Mummy	8.95
One On One(*)	8.95
Pinball Wizard	9.95
Punchy	d12.95 8.95
Pyjamarama	d12.95 8.95
QuackJack	d12.95 8.95
Redcoats	6.95
Return to Eden	9.95
Ring of Darkness	8.95
Roland Ahoy	d12.95 8.95
Roland Goes Digging	d12.95 8.95
Roland Square Bashing	d12.95 8.95
Roland in the Caves	d12.95 8.95
Roland in Space	d12.95 8.95
Roland in Time	d12.95 8.95
Roland on the Ropes	d12.95 8.95
Roland on the Run	d12.95 8.95
Shadow Fire(*)	9.95
Shadow of the Bear	8.95

Sherlock Holmes(*)	14.95
Sir Lancelot	6.95
Snooker	d12.95 8.95
Snowball	9.95
Software Star	7.95
Sorcery	8.95
Space Hawks	d12.95 8.95
Spanner Man	d12.95 8.95
Special Operations	6.95
Splatt	d12.95 8.95
Star Avenger	8.95
Star Commando	8.95
Star Clash	8.95
Steve Davis Snooker	7.95
Stockmarket	d12.95 8.95
Sultan's Maze	8.95
Super Pipeline II(*)	8.95
SuperChess 3.5	9.95
Survivor	7.95
Tank Busters	7.95
Technician Ted	7.95
The Hobbit	14.95
Tornado Low Level (TLL)(*)	7.95
Tripods	11.50
World Cup Football	7.95
Xanagrams	8.95
Zodiac	6.95
Zorakk The Conqueror	6.95

STOP PRESS

Raid	9.95
Beachhead	9.95
Home Reviews	8.95
Chess	d12.95
ACE(*)	9.95
A. Higgins Pool	d12.95 8.95
A. Higgins Snooker	d12.95 8.95
Frank-n-Stein	8.95
Fruit Machine	8.95
The Covenant	6.95
The Prize	8.95
Scout Steps Out	8.95
Gate Crasher	8.95
Gremlins	9.95
F. Bruno's Boxing	8.95
Exploding Fist(*)	9.95
Wizzards Lair	6.99
Rocky Horror Show(*)	8.95
Execution(*)	6.95
A View to Kill(*)	10.99
Red Moon	9.95
Astro Attack	8.95
Test Match	9.95
Psychedelia	6.00
Dummy Run(*)	9.95
Pawsl(*)	6.95
Blinky(*)	7.95
Confusion	6.95
Maidens Quest	6.95
Fire Ant	7.95

EDUCATIONAL

Animal Vegetable Mineral – Database of user defined objects.	8.95
First Steps with Mr Men.	8.95
Happy Letters. 3+. Letter recognition and matching.	8.95
Happy Numbers. 3+. Number recognition and counting.	8.95
Happy Writing. 3 to 6. Writing letters, numbers and words.	8.95
Map Reading. 7 to 15. Compas directions and coordinates.	8.95
Osprey. 9 to 14. Simulated bird conservation.	9.95
Time Man One. 4 to 9. Telling the time.	8.95
Time Man Two. 4 to 10. Twenty four hour clock.	8.95
Word Hang. 5 to 11. Spelling hangman.	8.95
Word Wise. 7 to 15. Geography database.	8.95
Star Watcher.	d23.95 19.95
Identikit.	7.95
The Key Factor(*)	9.95
The Maths Examiner (14+)	9.95
The Maths Invaders (4 to 12)	7.95
Number Painter (5+)	8.95
Physics Examiner (14+)	9.95

The list above doesn't include **everything** for the CPCs – there's an ever increasing list of disc-based software, and latest show introductions, for instance. So if you are hunting for an old classic, or something scorchingly new, or simply obscure . . . do try us! Note that a single asterisk (*) indicates new or imminent release.

Discount Club

Join the Software Supplies Discount Club. Annual fee is £10 which entitles you to 15% discount off normal RRP on all disc and tape based software – Regardless of machine type – any home or business computer.

Think about what your, friends, clubmates and colleagues can save! and no problems if you change or upgrade your machine. And if your not sure about joining, buying any six software titles at normal RRP over any period of time, counting now, ensures full membership for the year following.

Membership ensures regular mailings of new products, info and discs. Please specify machine type(s).

the software supplies company

p.o. box 19, whitstable, kent ct5 1tj

TRANSFORM LTD. Amstrad

COMPUTER AND MONITOR

CPC 464 Computer. Cassette with Green Monitor	249.00
CPC 464 Computer Cassette with Colour Monitor	359.00
CPC 664 Computer. Disc with Green Monitor	339.00
CPC 664 Computer. Disc with Colour Monitor	449.00

DISC DRIVE

3" Disc Drive with CP/M and Logo.	199.00
3" Disc Drive without Controller.	159.00

WORD PROCESSING

TASWORD 464

Wordprocessing program complete with full screen prompts and tutor. This program will allow you to produce letters, forms etc. fully compatible with all printers. Features include page numbering, search and find, search and replace, block copy and block move. The tutor will give full training on the use of the above functions.

SUPPLIED ON CASSETTE	£19.95
SUPPLIED ON DISC	£23.95

DATABASE

MASTERFILE 464

This database has set the standard for micro computers. With 100% machine code it will allow 240 characters per field. 40 fields per record and 34,000 characters per file. This will enable you to keep a record of 500 full names and addresses. Features include full search facilities, sort in ascending or descending order, column totals, 1 or 22 records per screen.

SUPPLIED ON CASSETTE	£24.95
SUPPLIED ON DISC	£29.95

SPREADSHEET

MASTERCALC

Spread sheet program from the same software house as Masterfile. Features includé 3000 cells, 8 byte floating precision, 40-80 column display, direct totals and subtotals, 3 bar histograms, cursor direction keys, add or delete columns or rows and high resolution screen copies on Epson compatible printers.

SUPPLIED ON CASSETTE	£24.95
SUPPLIED ON DISC	£29.95

Note cassette versions can be used on disc.

PRINTERS · LEADS · DISCS

DOT MATRIX PRINTERS

BROTHER M1009	£187.45p
EPSON LX80	£199.00p
EPSON FX80	£372.60p
NATIONAL PANASONIC KX-P1091	£287.50p
TAXAN KAGA KP810	£322.00p

DAISYWHEEL PRINTERS

SMITH CORONA TP1	£217.35p
BROTHER HR15	£401.86p

ACCESSORIES

CENTRONICS LEAD	£9.95p
3" DISCS	£4.95p
MODULATOR	£29.95p

All prices include VAT. Please add £5.75p for delivery of printer or computer. If you are ordering a printer and computer delivery is free.

TRANSFORM LTD (Dept. SU) 01-658 6350
24, West Oak, Beckenham, Kent BR3 2EZ

VISA

AMSTRAD EDUCATION SCHEME

(FOR BONA FIDE EDUCATION AND TRAINING ESTABLISHMENTS ONLY)

- £200 educational software and utility pack on disc supplied FREE with each disc drive system.
- Substantial discounts and software for schools.
- 1 year FREE hardware service contract.
- Business and training software for higher education.
- Amstrad, BBCBASIC(Z80) language pack.
- Details of the Amstrad network and hard disc system now available.

CPC464 Colour Computer

64K RAM, 32K ROM
standard high speed Basic,
27 colours, 80 column text,
3 channel sound, printer interface,
built-in data recorder.
System supplied with £100
software pack on cassette, also
with green or colour monitor.

SYSTEM
PRICES
START
AT
£189

CPC664 Colour Computer

As the CPC464, but built-in
unit is a 180K disc drive.
Includes CP/M 2.2, Dr LOGO
and AMSDOS. System supplied with
£200 educational software and utility
pack on disc, also with green
or colour monitor.

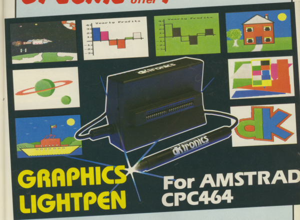
SYSTEM
PRICES
START
AT
£279

THE AMSTRAD CPC 664 COMPUTER SYSTEM

FURTHER INFORMATION FOR SCHOOLS, COLLEGES, POLYTECHNICS, UNIVERSITIES AND TRAINING SCHEMES CONTACT THE AMSTRAD EDUCATIONAL DISTRIBUTORS:

NORTHERN COMPUTERS LTD.
Churchfield Road, Frodsham, Cheshire WA6 6RD. Tel: (0928) 35700 (10 lines)

SPECIAL reader offer!



-and it's only £24.95!

IT'S the most imaginative add-on yet developed for the Amstrad – a highly sophisticated light pen that allows you to use the machine's superb graphics to the full.

And the first models to come off the production line are being offered exclusively to readers of Computing with the Amstrad!

We're making them available to readers at an unbeatable £24.95 – and that includes packing and despatching to your door by recorded delivery.

The complete kit consists of the light pen, Amstrad interface and a graphics package that will soon have you designing your own screen masterpieces.

The interface, which plugs into the expansion port on the back of the Amstrad, is compatable with both the CPC464 and CPC664 and the disc drive unit.

The light pen can be used with your own Basic or machine code programs quite easily. Several example programs in the manual demonstate its use.

The powerful graphics package supplied with the light pen will allow you to create colourful pictures which can be saved to tape or disc. It's entirely menu driven using colourful icons, so even the youngest child can use it.

The superb software allows you to draw in the sort of fine detail that is not normally possible with a light pen. By magnifying a section of the screen or using a combination of light pen and cursor keys single pixels can be plotted.

There's a choice of 10 colours, four pen sizes or airbrush. Rubber banding is also catered for and text can be written horizontally or vertically.

All in all, it makes it the most versatile package ever offered at the price.

Send for it NOW – use the official order form on Page 77.

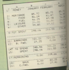
QUICK TO LEAR

THAT'S...

FINALIST BRITISH MICRO COMPUTING AWARDS 1985

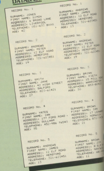
MINI OFFICE

SPREADSHEET



```
   ___A___   ___B___ ___C___ __D__
1:  MONEY    JANUARY FEBRUARY MARCH

2: MORTGAGE    85.72   85.72
3:     FOOD    46.24   41.47
3:     FUEL    46.25   47.78
4:             20.00   20.00
5:  LEISURE    99.85   17.12
6:    OTHER

8:TOT SPENT   298.06  211.55

11: EARNINGS  321.21  321.21
12:  B. FWD.   27.25    0.00

14: TO SPEND  348.46  321.21
15:    SPENT  298.06  211.55

17:REMAINING    0.00  109.66

19:     SAVE    0.00   82.25
20:   C.FWD.    0.00   27.41
```

DATABASE



```
RECORD No. 1                        RECORD No. 1

SURNAME: JONES                      SURNAME: ANDREWS
FIRST NAME: SIMON                   FIRST NAME: JAMES
ADDRESS1: 6 BROAD LANE              ADDRESS1: 12 ELF ROAD
ADDRESS2: LIVERPOOL                 ADDRESS2: HEREFORD
TELEPHONE: 051-633 8000             TELEPHONE: 321-623451
AGE: 42                             AGE: 13

RECORD No. 2                        RECORD No. 2

SURNAME: ANDREWS                    SURNAME: ANDREWS
FIRST NAME: PETER                   FIRST NAME: PETER
ADDRESS1: 12 ELF ROAD               ADDRESS1: 12 ELF ROAD
ADDRESS2: HEREFORD                  ADDRESS2: HEREFORD
TELEPHONE: 321-623451               TELEPHONE: 321-623451
AGE: 19                             AGE: 19

RECORD No. 3                        RECORD No. 3

SURNAME: SMITH                      SURNAME: BRINN
FIRST NAME: JANE                    FIRST NAME: KIETH
ADDRESS1: 42 HIGH STREET            ADDRESS1: 15 MILL
ADDRESS2: SALFORD                   ADDRESS2: WARRINGTON
TELEPHONE: 823-61421                TELEPHONE: 053-8092
AGE: 27                             AGE: 30

RECORD No. 4                        RECORD No. 4

SURNAME: YATES                      SURNAME: BROWN
FIRST NAME: IAN                     FIRST NAME: IAN
ADDRESS1: 177 FORD ROAD             ADDRESS1: 17 LEA
ADDRESS2: GULLHAM                   ADDRESS2: NORWICH
TELEPHONE: 452-986 76543            TELEPHONE: 831-
AGE: 35                             AGE: 21

RECORD No. 5                        RECORD No. 5

SURNAME: ANDREWS                    SURNAME: BROWN
FIRST NAME: JAMES                   FIRST NAME: JIM
ADDRESS1: 12 ELF ROAD               ADDRESS1: 8 ELM
ADDRESS2: HEREFORD                  ADDRESS2: NANTW
TELEPHONE: 321-623451               TELEPHONE: 481-
AGE: 13                             AGE: 11
```

JUST LOOK WHAT THIS PACKAGE CAN DO!

WORD PROCESSOR – Ideal for writing letters or reports! *Features:* Constant time display ● Constant word count (even shows words per minute) ● Normal or double-height text on screen or printout.

SPREADSHEET – Use your micro to manage your money! *Features:* Number display in rows and columns ● Continuous updating ● Update instantly reflected throughout spreadsheet ● Save results for future amendments.

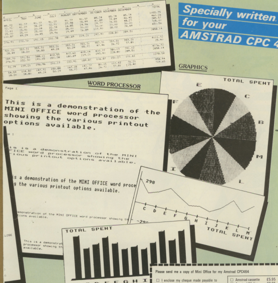
GRAPHICS – Turn those numbers into an exciting visual display! *Features:* 3D bar chart ● Pie chart ● Graph.

DATABASE – Use it like an office filing cabinet! *Features:* Retrieve files at a keystroke ● Sort ● Replace ● Save ● Print ● Search.

...and it's all a
price of just

RN, EASY TO USE

Specially written for your AMSTRAD CPC 464



```
E      F      G      H      I       J         K       L        M         N    O
APRIL  MAY    JUNE   JULY   AUGUST SEPTEMBER OCTOBER NOVEMBER DECEMBER       TOTAL

85.72  91.37  91.37  91.37  91.37  91.37     85.34   85.34    85.34     :   1055.75
32.71  41.27  38.29  32.71  27.98  35.99     40.89   39.89    46.45     :    460.26
22.41  25.41  20.04  22.34  16.85  24.96     29.77   35.55    48.23     :    385.57
25.00  25.00  25.00  25.00  25.00  50.00     50.00   50.00    50.00     :    505.00
100.87 49.29  14.45  29.96  19.49  36.89     107.90  58.02    79.49     :    651.56
----------------------------------------------------------------------------------
276.91 232.50 191.15 201.38 180.69 219.21    293.90  228.80   289.51    :   2050.14
----------------------------------------------------------------------------------
                                                                        :   4111.52
                                                                        :     25.40
----------------------------------------------------------------------------------
321.21 353.31 353.31 353.31 353.31 353.31    353.31  353.31   353.31    :   4136.72
28.49  18.20  34.80  49.24  50.29  55.73     47.46   26.72    37.81     :   2050.14
----------------------------------------------------------------------------------
349.70 371.51 388.11 402.55 403.60 409.04    400.77  380.03   391.12    :
276.91 232.50 191.15 201.38 180.69 219.21    293.90  228.80   289.51    :
----------------------------------------------------------------------------------
72.79  139.21 196.96 201.17 222.91 189.87    106.87  151.23   101.61    :   1278.58
----------------------------------------------------------------------------------
54.59  104.40 147.72 150.88 167.18 142.57    80.15   113.42   76.21     :    958.94
18.20  34.80  49.24  50.29  55.73  47.46     26.72   37.81    25.40     :    719.65
```

WORD PROCESSOR

Page 1

This is a demonstration of the MINI OFFICE word processor showing the various printout options available.

Page 1

is is a demonstration of the MINI FICE word processor showing the rious printout options available.

is a demonstration of the MINI OFFICE word proce g the various printout options available.

emonstration of the MINI OFFICE word processor showing th tions available.

LOSE

This is a demonstrat processor showing th available.

GRAPHICS

the unbelievable £5.95

CASSETTE
3" DISC £9.95

DATABASE SOFTWARE

Please send me a copy of Mini Office for my Amstrad CPC464

□ I enclose my cheque made payable to Database Publications Ltd.

□ Amstrad cassette	£5.95
□ Amstrad 3" disc	£9.95

Please tick box

I wish to pay by
□ Access □ Visa No.________________ Expiry date________

Signed______________________________

Name______________________________

Address______________________________

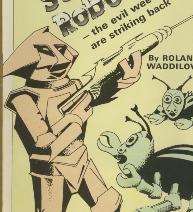
POST TO: Mini-Office Offer, Database Software,
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

CA9

Send for Robot Ron!

– the evil weevils are striking back

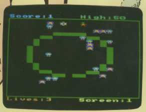
By ROLAND WADDILOVE



OH no! There's been an outbreak of mutant evil weevils again. It happens every year at about this time. It's the weevils' breeding season, you see, and they're an absolute menace. Send for Robot Ron!

Ron is a remote-controlled robot weevil-destroyer armed with a super Zap gun. One blast from this and the weevil's on its way to that great cornfield in the sky.

Unfortunately Ron isn't impervious to these little beasties. They get into his joints, seizing them up



```
10 REM Robot Ron v The Evil Weevils
20 REM By R.A.Waddilove
30 REM(c)Computing With The Amstrad
40 MEMORY &9FFF
50 GOSUB 420:REM initialise
60 GOSUB 1530:REM instructions
70 WHILE NOT fed.up
80 GOSUB 1230:REM scores
90 WHILE lives
100 GOSUB 670:REM set variables
110 WHILE a(x,y)=0 AND m
120 GOSUB 200:GOSUB 260:IF p THEN GOS
UB 390 ELSE GOSUB 360
130 WEND
140 IF m THEN GOSUB 930 ELSE screen=s
creen+(screen>1):GOSUB 910
150 WEND
160 IF sc>hiscore(5) THEN GOSUB 1350
170 WEND
180 END
190 REM -------- move man ----------
200 i=x+(INKEY(g)>-1)-(INKEY(h)>-1):j
=y+(INKEY(e)>-1)-(INKEY(f)>-1):IF a(i
,j)>1 THEN RETURN
210 IF a(i,j)>1 THEN RETURN
220 CALL &A000,x,y,i,j,2:IF p=0 AND (
i<>x OR j<>y) THEN mx=i-x:my=j-y:SOUN
D 132,100,1,4
230 x=i:y=j
240 RETURN
250 REM ------- move robots --------
260 SOUND 130,m*20+500,10,4:i=r(n,0):
j=r(n,1):IF i=0 THEN 300
270 IF a(i,j)<>1 THEN LOCATE i,j:PRIN
T " ":r(n,0)=0:a(i,j)=0:m=m-1:GOTO 30
0
```

and causing Ron to overheat and explode.

Don't panic, though. We can rebuild him – we have the technology. But this can only be done three times. After that his parts are unserviceable.

There are nine screens to test your skill and a high-score table to chart your progress. Prelude in G for one Amstrad accompanies the title page and high-score table.

As you progress through each screen the number of weevils gets fewer – but they also get more agile. You'll need to keep your wits about you and your finger on the fire button.

A machine code routine is used to print the weevils, Ron and laser bolt. This creates a multicoloured character about ten times faster than Basic can print a single-colour one.

It also allows large numbers of bright, colourful characters to be moved around very quickly.

Each subroutine has a title describing its function and the program is fairly well structured. Be careful when entering the data.

VARIABLES

x,y	Ron's coordinates.
a(20,25)	Screen map.
r(20,1)	Weevils' coordinates.
name$(5)	Names in high-score table.
hiscore(5)	High scores.
p,q	Coordinates of laser bolt.
sc	Score.
screen	Screen number.
m	Weevils left.
n	Weevil to be moved.

```
280 i=i-(i<x)+(i>x):j=j+(j>y)-(j<y)
290 IF a(i,j)=0 THEN CALL &A000,r(n,0
),r(n,1),i,j,0:a(r(n,0),r(n,1))=0:a(i
,j)=1:r(n,0)=i:r(n,1)=j
300 i=r(n+1,0):j=r(n+1,1):IF i=0 THEN
 340
310 IF a(i,j)<>1 THEN LOCATE i,j:PRIN
T " ":r(n+1,0)=0:a(i,j)=0:m=m-1:GOTO
340
320 i=i-(i<x)+(i>x):j=j+(j>y)-(j<y)
330 IF a(i,j)=0 THEN CALL &A000,r(n+1
,0),r(n+1,1),i,j,1:a(r(n+1,0),r(n+1,1
))=0:a(i,j)=1:r(n+1,0)=i:r(n+1,1)=j
340 n=(n+2)MOD screen*2:IF r(n,0) OR
r(n+1,0) OR m=0 THEN RETURN ELSE 340
350 REM ---------- fire ----------
360 IF INKEY(d)<0 THEN RETURN
370 p=x+mx:q=y+my:IF a(p,q)=3 THEN p=
0:RETURN ELSE IF a(p,q)=1 THEN CALL &
A000,p,q,p,q,4:p=0:a(p,q)=4:RETURN
380 CALL &A000,p,q,p,q,3:a(p,q)=2:SOU
ND 129,10,300,15,1,1,1:RETURN
390 i=p+mx:j=q+my:IF a(i,j)=1 THEN a(
p,q)=0:a(i,j)=4:CALL &A000,p,q,i,j,4:
p=0:SOUND 129,1500,100,15,2,0,15:sc=s
c+1:LOCATE 7,1:PRINT MID$(STR$(sc),2)
:RETURN
400 a(p,q)=0:IF a(i,j)=3 THEN LOCATE
p,q:PRINT " ":p=0:SOUND 129,0,0,0 ELS
E CALL &A000,p,q,i,j,3:a(i,j)=2:p=i:q
=j
410 RETURN
420 REM ------- initialise ---------
430 DEFINT a-z
440 DIM a(20,25),r(20,1),name$(5),his
core(5)
450 FOR i=1 TO 5:name$(i)="The A Team
":hiscore(i)=60-i*10:NEXT
460 ENT 1,200,2,1:ENV 1,15,-1,25
470 ENV 2,15,-1,10
480 FOR i=0 TO 98
490 READ a$:POKE &A000+i,VAL("&"+a$)
500 NEXT
510 DATA DD,7E,00,87,87,87,87,87,32,4
7,A0,26,C0,DD,7E,04,3D,87,87,6F,11,50
,00,DD,46,02,05,19,10,FD,E5,26,C0,DD,
7E,00,3D,87,87,6F,11,50,00,DD,46,06,0
5,19,10,FD,11,FD,07,06,08,36,00,23,36
,00,23,36,00,23,36,00
520 DATA 19,10,F2,E1,11,00,A1,06,08,1
A,77,13,23,1A,77,13,23,1A,77,13,23,1A
,77,13,78,01,FD,07,09,47,10,E9,C9
530 FOR i=0 TO 159:READ j:POKE &A100+
i,j:NEXT
540 REM Robot 1
550 DATA 20,60,60,0,105,131,195,40,10
5,169,233,40,41,3,3,40,20,0,20,0,0,40
,40,0,0,160,160,0,80,160,240,0
560 REM Robot 2
570 DATA 80,240,240,160,160,1,2,80,16
0,169,86,80,161,3,3,82,80,2,1,160,0,2
40,240,0,20,0,0,40,40,0,0,20
580 REM Man
590 DATA 0,64,128,0,0,128,64,0,0,64,1
28,0,4,12,12,8,8,76,140,4,32,12,12,16
,0,8,4,0,4,8,4,8
600 REM Bolt
610 DATA 0,0,0,0,0,8,4,0,0,0,0,0,0,84
,168,0,0,84,168,0,0,0,0,0,0,8,4,0,0,0
,0,0
620 REM Explosion
630 DATA 8,84,168,4,4,136,68,0,168,8,
4,84,84,216,228,168,84,216,228,168,16
8,8,4,84,4,136,68,8,0,84,168,4
640 FOR i=1 TO 20:a(i,2)=3:a(i,24)=3:
NEXT
650 FOR i=2 TO 24:a(1,i)=3:a(20,i)=3:
NEXT
660 RETURN
670 REM ------ set variables ------
680 CLS #1:PEN 1
690 FOR i=2 TO 19:FOR j=3 TO 23:a(i,j
)=0:NEXT:NEXT
700 j=1+INT(RND*5):DEG
710 FOR i=0 TO 350 STEP 10
720 x=10+6*COS(i*j):y=12+6*SIN(i):LOC
ATE x,y:PRINT CHR$(207):a(x,y)=5
730 NEXT
740 x=10:y=12:p=0:a(x,y)=0
750 m=screen*2
760 FOR i=0 TO m-1
770 IF RND>0.5 THEN r(i,0)=2+INT(RND*
5) ELSE r(i,0)=15+INT(RND*5)
780 IF RND>0.5 THEN r(i,1)=3+INT(RND*
5) ELSE r(i,1)=15+INT(RND*5)
790 IF a(r(i,0),r(i,1)) THEN 770 ELSE
 a(r(i,0),r(i,1))=1
800 NEXT
810 REM ----- set up screen -----
820 MOVE 0,18:DRAW 0,380,1:DRAW 638,3
80:DRAW 638,18:DRAW 0,18
830 PEN 15:LOCATE 1,1:PRINT "Score:"M
ID$(STR$(sc),2);:PEN 14:PRINT TAB(12)
"High:"MID$(STR$(hiscore(1)),2)
840 PEN 13:LOCATE 1,25:PRINT "Lives:"
MID$(STR$(lives),2);:PEN 12:PRINT TAB
(12)"Screen:"MID$(STR$(10-screen),2)
850 CALL &A000,x,y,x,y,2
```

```
860 PEN 4
870 FOR i=0 TO screen*2-1
880 LOCATE r(i,0),r(i,1):PRINT CHR$(2
31)
890 NEXT
900 PEN 15
910 FOR i=0 TO 2000:NEXT
920 RETURN
930 REM ---------- dead ----------
940 SOUND 129,1500,100,15,2,0,15
950 lives=lives-1
960 CALL &A000,x,y,x,y,4
970 FOR i=0 TO 5000:NEXT
980 IF lives THEN RETURN
990 TAG
1000 PLOT -10,-10,2:MOVE 160,200:PRIN
T "Game Over";
1010 TAGOFF:PRINT CHR$(23);CHR$(1);:T
AG
1020 PLOT -10,-10,6:MOVE 164,202:PRIN
T "Game Over";
1030 TAGOFF:PRINT CHR$(23);CHR$(0);
1040 FOR i=0 TO 5000:NEXT
1050 RETURN
1060 REM -------- tune --------
1070 WHILE INKEY(47)<0 AND INKEY(76)<
0
1080 RESTORE 1190:READ pitch:SOUND 12
9,213,200,4:SOUND 130,253,2000,4:SOUN
D 1,127,2000,4
1090 WHILE pitch>0 AND INKEY(47)<0 AN
D INKEY(76)<0
1100 SOUND 4,pitch,10,6:SOUND 4,159,1
0,6:READ pitch
1110 WEND
1120 READ pitch:SOUND 129,319,200,5:S
OUND 1,213,200,4:SOUND 1,159,200,4
1130 WHILE pitch>0 AND INKEY(47)<0 AN
D INKEY(76)<0
1140 SOUND 4,pitch,12,6:READ pitch
1150 WEND
1160 WEND
1170 IF INKEY(47)>-1 THEN d=47:e=69:f
=71:g=39:h=31 ELSE d=76:e=72:f=73:g=7
4:h=75
1180 RETURN
1190 DATA 142,127,119, 142,127,119, 1
27,119,106, 127,119,106, 119,106,95,
119,106,95, 106,95,84, 106,95,84,0
1200 DATA 80,84,95, 84,95,106, 95,106
,119, 106,119,127
1210 DATA 80,84,95, 84,95,106, 95,106
,119, 106,119,127
1220 DATA 119,127,142,159,169,190,213
,239,253,284,319,253,284,319,239,253,
284,213,239,253,190,213,239,169,190,2
13,190,179,169,159,142,159,0
1230 REM --------- scores ---------
1240 MODE 1:BORDER 1
1250 DRAW 0,398,1:DRAW 638,398:DRAW 6
38,0:DRAW 0,0
1260 PAPER 2:PEN 3:LOCATE 11,2:PRINT
"   High  Scores   ":PAPER 0
1270 FOR i=1 TO 5
1280 LOCATE 7,5+i*2:PEN 3:PRINT i;:PE
N 1:PRINT " ";name$(i);" ";:PEN 2:WHI
LE POS(#0)<29:PRINT ".";:WEND:PEN 3:P
RINT hiscore(i)
1290 NEXT
1300 PAPER 1:PEN 0:LOCATE 5,24:PRINT
" Press SPACE or FIRE to start ":PAPE
R 0
1310 WHILE INKEY$<>"":WEND:GOSUB 1060
1320 SOUND 129,0:SOUND 130,0:SOUND 13
2,0
1330 BORDER 0:MODE 0:WINDOW #1,2,19,3
,23:sc=0:screen=9:lives=3
1340 RETURN
1350 REM ------- high score -------
1360 MODE 1
1370 DRAW 0,398,1:DRAW 638,398:DRAW 6
38,0:DRAW 0,0
1380 PEN 1:PAPER 0:LOCATE 7,5:PRINT "
This  is  a  new  high  score"
1390 LOCATE 10,10:PEN 3:PRINT "Please
 type in your name"
1400 LOCATE 14,20:PAPER 2:PEN 3:PRINT
 " ?.......... "
1410 LOCATE 16,20:k$="":name$(5)=""
1420 WHILE INKEY$<>"":WEND
1430 WHILE k$<>CHR$(13)
1440 IF k$>CHR$(31) AND k$<"z" AND LE
N(name$(5))<10 THEN name$(5)=name$(5)
+k$:PRINT k$;
1450 IF k$=CHR$(127) AND LEN(name$(5)
) THEN name$(5)=LEFT$(name$(5),LEN(na
me$(5))-1):PRINT CHR$(8);CHR$(16);
1460 k$=INKEY$
1470 WEND
1480 hiscore(5)=sc
1490 FOR i=5 TO 2 STEP -1
1500 IF hiscore(i)>hiscore(i-1) THEN
k$=name$(i):name$(i)=name$(i-1):name$
(i-1)=k$:sc=hiscore(i):hiscore(i)=his
core(i-1):hiscore(i-1)=sc
1510 NEXT
1520 RETURN
1530 REM ----- instructions ------
1540 MODE 1:CALL &BB4E:CALL &BC02
1550 INK 2,2:INK 0,11:INK 3,11:BORDER
 11
1560 WINDOW #1,11,31,1,3:PAPER #1,3:C
LS #1
1570 WINDOW #1,10,30,2,4:PAPER #1,1:C
LS #1
1580 PEN 3:LOCATE 1,25:PRINT "Robot
Ron":PLOT -10,-10,2
1590 FOR x=0 TO 158 STEP 2:FOR y=2 TO
 14 STEP 2
1600 IF TEST(x,y) THEN PLOT x+230,y+y
+344:PLOT x+230,y+y+346
1610 NEXT:NEXT:PEN #1,2
1620 WINDOW #1,2,40,5,24:PAPER #1,3:C
LS #1:INK 3,6
1630 WINDOW #1,1,39,6,25:PAPER #1,1:C
LS #1
1640 PRINT #1
1650 PRINT #1," Oh no, there's been a
n outbreak of"
1660 PRINT #1," mutant evil weevils a
gain. These"
1670 PRINT #1," nasty creatures must
be exterminated"
1680 PRINT #1," at all costs.":PRINT
#1
1690 PRINT #1," Send Robot Ron in to
their den to"
1700 PRINT #1," destroy them with his
 Zap gun."
1710 PRINT #1
1720 PRINT #1," Ron can be controlled
 using the"
1730 PRINT #1," following keys...
1740 PRINT #1
1750 PRINT #1,"   A=up    Z=down    <
=left   >=right"
1760 PRINT #1
1770 PRINT #1,SPC(11)"SPACE bar fires
"
1780 PRINT #1
1790 PRINT #1,SPC(11)"Or use Joystick
"
1800 PEN #1,1:PAPER #1,2:LOCATE #1,5,
19:PRINT #1," Press SPACE or FIRE to
start ":PAPER #1,0
1810 WHILE INKEY$<>"":WEND
1820 GOSUB 1060
1830 RESTORE 1840:FOR i=0 TO 15:READ
j:INK i,j:NEXT:INK 7,26,13
1840 DATA 0,10,6,24,2,8,20,26,15,3,7,
9,25,16,22,23
1850 RETURN
```

Hey presto, it's XORcised!

MIKE BIBBY concludes his series on how your Amstrad works

LAST month we saw how the binary operators AND and OR can be used to combine pairs of binary numbers. The example we used was that of turning machines on and off under computer control.

Of course these operators have far more uses than this. To illustrate one, consider the Ascii character set. The codes for A to Z are in the range 65-90, while their lower case equivalents, a to z, are in the range 97-122.

Looked at in this decimal way, there seems little relation between the upper and lower case sets. If we look at them in hex, though, we can see that:

A . . . Z runs from &41 to &5A

a . . . z runs from &61 to &7A

I hope you can see the pattern.

In fact the numerical Ascii difference between a lower case character and its upper case equivalent is always &20. Looked at in binary, this difference is %00100000. In other words, bit five is set for lower case, and is clear for upper case – remember, we start with the zero bit.

For example, the code for A is:

%01000001

whereas the code for a is:

%01100001

Similarly, the code for Z is:

%01011010

and the code for z is:

%01111010

In both cases the only difference is in bit five.

So if we have an Ascii code for a letter, we can *force* it to be upper case by clearing bit five to zero. We can do this by ANDing the code for the letter with the mask %11011111 (&DF).

Remember, the bits in the mask that contain 1 will leave the corresponding bits in the Ascii code for the letter unchanged in the resultant byte, whether they be 0 or 1. On the other hand, the bit in the mask with 0 in it will force the matching result bit to be zero. So:

```
      %01100001  ( the code for a )
  AND %11011111  ( the mask - &DF )
gives %01000001  ( the code for A )
```

It won't surprise you to learn that we can reverse the procedure – forcing upper case into lower case – by using OR to set bit five. This time the mask will be %00100000, the 0s leaving things unchanged in the resultant byte, the 1 forcing a corresponding 1 in bit five of the result bit. So:

```
      %01011010  ( the code for Z )
   OR %00100000  ( the mask - &20 )
gives %01111010  ( the code for z )
```

One further use for AND is to test if a particular bit in a byte is set. We just AND that byte with a mask consisting of a 1 in the bit being tested, with 0s in all the rest. The bits with 0 in them, of course, set the corresponding bits in the resultant byte to zero.

Since the rest of the bits are already cleared to zero by the mask, the only thing that could stop the entire resultant byte being zero is the value derived from the bit under investigation:

- If that bit is set, the corresponding result bit will be set also (1 AND 1 = 1) so the resultant byte will be non-zero.
- If the bit being checked is clear, the corresponding result bit will be clear (0 AND 1 = 0) so the resultant byte is zero.

Those of you following the machine code series in *Computing with the Amstrad* will soon know, if you don't know already, that we can differentiate between zero and non-zero bytes fairly easily.

Let's see how this works in practice. If we were testing for bit four being set, the mask would be %00010000.

Try ANDing this value with %00110100, where bit four is set, and also with %00101100, where bit four is clear, and you'll see that the resulting bytes are non-zero and zero respectively.

So what of EOR/XOR? Well, its function is to return a 1 if the pair of bits being combined differ, and 0 if they're identical. Given this, we can use XOR to test which bytes in a bit differ. For example:

```
    %10101110
XOR %11001101
gives %01100011
```

where the set bits neatly mark out the differing pairs.

We can also use EOR/XOR to complement or NOT a byte, by XORing it with a mask of %11111111. Bits that contain 1s will give 0 (since 1 XOR 1 = 0), while bits that contain zero will give 1, since 0 XOR 1 = 1.

This is exactly what we want to happen with a NOT – change the 0s to 1s and vice versa. For example:

```
      %10101101
  XOR %11111111
gives %01010010 ( the complement )
```

We can also use EOR/XOR to test if two bytes are identical. If the result when we XOR is zero, they must have been identical since every pair of bits must have given zero, which only happens when the bit values are the same.

If there's a non-zero result there must have been a pair of bits that differ, so the two bytes under consideration must differ. For example:

```
      %10101010
  XOR %10101010
gives %00000000
```

whereas:

```
  %10101110
XOR %10101010
  %00000100
```

which is, of course, non-zero, since the bytes differ.

You've probably already come

across EOR/XOR in graphics application programs where it's widely used for its "hey presto" effect. This is based on the fact that if you XOR a first byte with a second and then XOR the result of that once more with the second byte, the first byte reappears. Look at this, if you don't believe me:

```
    %01011100    ( first byte )
XOR %01110010    ( second byte )
    %00101110    (   result    )
XOR %01110010 ( second byte again )

    %01011100 (  first byte back! )
```

We use this XORing technique to draw things on a background and then move on, leaving the background unchanged. In this case the first byte is the background paper number. If we then XOR our second byte – corresponding to the pen number of whatever it is we're drawing – on to the background, it will be displayed in the resultant pen number. It's rather like mixing colours mathematically.

```
10 REM PROGRAM I
20 REM XOR Demo
30 MODE 1
40 PRINT CHR$(23);CHR$(1)
50 WHILE NOT bored
60 MOVE 0,0
70 DRAW 400,400
80 FOR delay=0 TO 1000
90 NEXT
100 WEND
```

Program I: Using XOR in graphics

To get rid of what we've drawn, we draw it again with the same pen number, once more under the influence of XOR. Of course XORing twice with the same byte gives us the original byte back. This results in whatever it is being drawn appearing in the original, background colour. Hey presto – it's gone!

Suppose we clear the background to paper zero and then draw a line across it in pen one, not just by "sticking it on" but by XORing it on – never mind how. The resulting line will also be in pen 1 since 0 XOR 1 = 1.

But if we XOR the exact same line onto the screen again, still in pen one, it will be going on top of itself. As the line on the screen is already in pen one, the new line will be drawn in pen zero – since 1 XOR 1 = 0. And, since zero is our background colour, the line "disappears".

Program I gives a demonstration of the sort of technique. The actual details of how Amstrad Basic works it are beyond the scope of this series. It shouldn't be too hard to see what's going on, though.

Well, that's the end of the series. Hopefully you'll have gained some idea of the power of binary numbers and the ways they can be combined. I've only touched on a fraction of the potential uses, but you'll be well equipped to work things out for yourself from now on.

And if you're looking to take these ideas further, why not try my machine code series? These articles should get you off to a flying start.

When it all goes quiet...

JO STORK provides some valuable advice on how to avoid the disasters of a computer breakdown

AS a full-time computer consultant I am asked one question more frequently than any other. The query may be summarised as: "Why does computer 'A' cost so much more than computer 'B' when both seem to be doing much the same task?"

My answer is always the same. This is that virtually the whole price differential is . . . *insurance.*

With a puzzled look, my client next shows me the manufacturer's advertising blurb or else repeats the retailer's sales pitch.

Every word implies that a Wonderbyte 8k handles the books of three building societies with enough spare capacity for the payroll of a local council. Nowhere is "insurance" ever mentioned.

They realise I am serious once I explain that a 16-bit processor, more memory and a couple of extra hardware widgets do not treble the price.

I will explain what I mean by "insurance" by using an example based on my own company.

Just recently, by the most conservative of estimates, my firm could have suffered a £2,300 turnover shortfall thanks to a £10 chip failure.

This horrifying figure is easily calculated. My business sends invoices totalling thousands of pounds each month to its clients, plus having to record those bills which must be paid.

An Amstrad with disc drive and simple printer can finish this work in a couple of hours each week. I purposely do not include the mounds of reports and manuals which need word processing each month, since that is the nature of my normal work.

When my clients do similar calculations based on their own organisations, they likewise conclude an extra £1,250+ may need spending on equipment which will be idle most of the week.

Firms which are unable to request payments tend to have limited life expectancy. However, before you create your business system is the time to remember that a year after computerisation it is difficult to revert to manual working. Even when it is possible to use an HB Pencil + Grey Matter, the extra hours spent incur huge costs.

On this occasion the errant computer was not my Amstrad, but during the working life of any computer you must expect some breakdowns.

Extended Warranty Insurance is of limited use to a businessman since he is less concerned with the cost of the repair than with the time it takes to make this repair.

I mean no criticism of any home-computer manufacturer when I warn you that a month will probably pass before it returns from the repairers. Service times of far longer are common, particularly if your dealer is inefficient or you live in a more inaccessible corner of the land.

Since I am keen to stay in business, I had provided real "insurance" long before the chip went fizz.

I have raised this point of long repair times with several manufacturers of home micros and their *perfectly reasonable* reply is always the same.

It goes like this: "If you wish our products to be considered for serious business use, then a greater level of end-user support would be required and supporting this level of back-up will unfortunately greatly push up the cost".

Since the manufacturers are not interested, you must tackle this problem yourself.

Protecting your organisation may require buying a computer from sources whose operations are geared to the businessman, rather than to selling Ghetto-blasters.

They will add to the price of their expensive kit and software a further £20+ each month for a maintainance agreement which guarantees a three-day time to fix.

Alternatively you could adopt double-redundancy – that is, having two of everything essential to the continued working of your business system. Just as you back-up your files and programs, so you can back-up your hardware.

As the "games" sector is now virtually saturated, increased emphasis will be placed on the Amstrad's suitability for the small business, club or similarly-sized organisation.

Unlike many home micros, these claims for the Amstrad are feasible, for it appears to be a reliable machine. Furthermore the performance justifies the sales impact and the available software is rapidly improving, both in variety and quality.

All that is lacking is the end-user hardware support, which currently is worse than you accept for a rental TV or washing machine.

If some organisation can step in with a financially attractive lifebelt for the customer who finds his equipment has suddenly gone quiet, then Mr Sugar's little wonder will prove a major force in the commercial sector.

Some sort of short-term rental club may be the most cost-effective solution.

Until such time as the problem of insuring your commercial viability is resolved, I can only advise you to think long and hard before placing all your financial eggs in an electronic basket of limited security.

Part II of ROLAND WADDILOVE's series on how to produce a better class of graphics using machine code

The multi-coloured aliens are landing

LAST month we looked at how the Mode 0 screen memory was organised using a few simple programs . . . Now we're going to try some short machine code routines to print a multi-coloured character on the screen.

First I'll briefly recap what we learnt last time.

The screen is organised into rows of pixels and a single character occupies 32 bytes of memory, eight rows of four bytes. The bit pattern of each byte in the screen memory holds the information for two horizontal pixels.

Going down the screen, the rows of pixels are in groups of eight. There are 25 groups on the screen – these are the lines we LOCATE and PRINT on. The address of each row of pixels in the group is &800 more than the previous row and each group starts &50 more than the one above. Figure I shows the top left corner of the screen.

```
10 REM PROGRAM I
20 POKE &9000,8:POKE &9001,4
30 FOR i=0 TO 31
40 READ j:POKE &9002+i,j
50 NEXT
60 REM alien
70 REM Rows=8/Columns=4
80 DATA 4,12,12,8,72,148,104,132,72
90 DATA 156,108,132,28,60,60,44,28
100 DATA 32,16,44,4,48,48,8,8,0,0,4
110 DATA 4,0,0,8
```

Program I

To display a normal-size character on the screen all that is necessary is to work out the data required and store it in eight rows of four bytes somewhere in the screen RAM. To make it easy, first we'll print a normal-size character exactly on a line.

We need some character data, so run Program I to poke the data for an alien to &9001. This information will be used by the machine code routines.

Program II is an assembler listing of the routine to print the alien. You can either use an assembler to enter the mnemonics, or enter the hex codes one at a time using Program III.

To see Program II's print routine in action enter:

```
MODE 0:?:?:?
CALL &8000
```

The alien is printed in the top left corner of the screen at &C000.

First BC is loaded with the number of columns and rows which is stored at &9000, DE with the address of the data, &9002 and HL the screen address where we want it to be printed, &C000. B is the inner loop counter, the number of columns, and C is the outer loop, the number of rows.

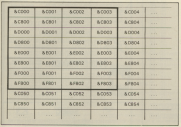
The loop counters and address of the row are first saved, then the inner loop runs along the row collecting the data pointed to by DE and storing it at

```
RAW Assembler V.3

Pass... 2           ORG &8000

                    PROGRAM II
8000:ED 4B 00 90    LD BC,(&9000)
8004:11 02 90       LD DE,&9002
8007:21 00 C0       LD HL,&C000
800A:               .loop1
800A:C5             PUSH BC
800B:E5             PUSH HL
800C:               .loop2
800C:1A             LD A,(DE)
800D:77             LD (HL),A
800E:23             INC HL
800F:13             INC DE
8010:10 FA          DJNZ loop2
8012:E1             POP HL
8013:01 00 08       LD BC,&800
8016:09             ADD HL,BC
8017:C1             POP BC
8018:0D             DEC C
8019:20 EF          JR NZ,loop1
801B:C9             RET
801C:               END
```

Program II

```
10 REM PROGRAM III
20 a=&8000
30 PRINT HEX$(a);" : ";
40 INPUT h$
50 POKE a,VAL("&"+h$)
60 a=a+1:GOTO 30
```

Program III



&C000	&C001	&C002	&C003	&C004	...
&C800	&C801	&C802	&C803	&C804	...
&D000	&D001	&D002	&D003	&D004	...
&D800	&D801	&D802	&D803	&D804	...
&E000	&E001	&E002	&E003	&E004	...
&E800	&E801	&E802	&E803	&E804	...
&F000	&F001	&F002	&F003	&F004	...
&F800	&F801	&F802	&F803	&F804	...
&C050	&C051	&C052	&C053	&C054	...
&C850	&C851	&C852	&C853	&C854	...
...	...	...	...	...	...

Figure I . . . Memory map of top left of the Mode 0 screen RAM

the address pointed to by HL. HL and DE are then incremented to get the next data item and screen address.

At the end of the row the address of the start of the row is restored and &800 added to HL to get the address of the next row. The loop counters are restored and C, the number of rows, is decremented.

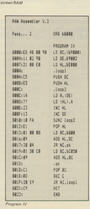
If you study the routine you'll see that the width of the character is irrelevent. It doesn't have to be four bytes wide – this just happens to be the width of our alien.

B is loaded with the width at the start and is decremented every time round the inner loop until it's zero. HL and DE are incremented to give the correct addresses.

What about the height? The alien is eight pixels deep and we're printing it at &C000. The outer loop adds &800 to the address in HL each time to get the start of the next row so the address of the last row is &F800.

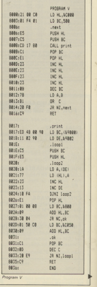
Suppose that the character isn't exactly on the line. It might be printed at the fifth pixel down so that it's half on one line and half on the next.

We're going to have problems here because the character is split over two groups of eight rows. When we get down to the bottom row of a group adding &800 to the address



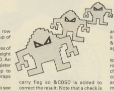
```
RAW Assembler V.3

Pass... 2           ORG &8000

                    PROGRAM IV
8000:ED 4B 00 90    LD BC.(&9000)
8004:11 02 90       LD DE,&9002
8007:21 00 E0       LD HL,&E000
800A:               .loop1
800A:C5             PUSH BC
800B:E5             PUSH HL
800C:               .loop2
800C:1A             LD A,(DE)
800D:77             LD (HL),A
800E:23             INC HL
800F:13             INC DE
8010:10 FA          DJNZ loop2
8012:E1             POP HL
8013:01 00 08       LD BC,&800
8016:09             ADD HL,BC
8017:30 04          JR NC,ok
8019:01 50 C0       LD BC,&C050
801C:09             ADD HL,BC
801D:               .ok
801D:C1             POP BC
801E:0D             DEC C
801F:20 E9          JR NZ,loop1
8021:C9             RET
8022:               END
```

Program IV



```
RAW Assembler V.3

Pass... 2           ORG &8000

                    PROGRAM V
8000:21 00 C0       LD HL,&C000
8003:01 F4 01       LD BC,500
8006:               .next
8006:E5             PUSH HL
8007:C5             PUSH BC
8008:CD 17 80       CALL print
800B:C1             POP BC
800C:E1             POP HL
800D:23             INC HL
800E:23             INC HL
800F:23             INC HL
8010:23             INC HL
8011:0B             DEC BC
8012:78             LD A,B
8013:B1             OR  C
8014:20 F0          JR NZ,next
8016:C9             RET

8017:               .print
8017:ED 4B 00 90    LD BC,(&9000)
801B:11 02 90       LD DE,&9002
801E:               .loop1
801E:C5             PUSH BC
801F:E5             PUSH HL
8020:               .loop2
8020:1A             LD A,(DE)
8021:77             LD (HL),A
8022:23             INC HL
8023:13             INC DE
8024:10 FA          DJNZ loop2
8026:E1             POP HL
8027:01 00 08       LD BC,&800
802A:09             ADD HL,BC
802B:30 04          JR NC,ok
802D:01 50 C0       LD BC.&C050
8030:09             ADD HL,BC
8031:               .ok
8031:C1             POP BC
8032:0D             DEC C
8033:20 E9          JR NZ,loop1
8035:C9             RET
8036:               END
```

Program V



will not give the address of the row which is the top of the next group of eight rows.

In Figure I &C050 is the address of the first row in the next group of eight rows, but &F800+&800 will be 0. An overflow will occur because a register pair can only hold numbers up to &FFFF. if this is exceeded it wraps round the 0 again.

What we need to do is check to see if there has been an overflow, and if there has then add a correction factor – &C050. If there hasn't been an overflow we're OK.

Program IV is the same routine as before but an overflow check has been added. To test it we'll print our alien five pixels down, at &E000.

The first four rows are OK, they start at addresses &E000, &E800, &F000 and &F800. Then there will be an overflow when &800 is added to HL for the fifth row. This sets the carry flag so &C050 is added to correct the result. Note that a check is made with each row and not just the fifth. This makes the routine general.

Again enter:

```
MODE 0:?:?:?
CALL &8000
```

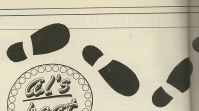
and you'll see the alien printed half on the first line and half on the second.

This short routine will now print any size multi-coloured character at any screen address. It doesn't matter whether it's exactly on a line or split over two or more, the code checks and corrects whenever necessary. Try it and see! Set HL to any value from &C000 on, assemble the routine again and call &8000.

It's difficult to get any idea of the speed advantage of the machine code routine over Basic when only one character is being printed. Program V completely fills the screen with aliens, and considering that each alien is made up of several different colours it's amazingly fast.

HL is used to store the address and BC is the loop counter. These are saved before printing the alien and restored afterwards. The print routine itself has been kept separate and has been labelled *print* for obvious reasons and is called as a subroutine. The HL register pair is used to pass the address to print the character.

● *I think that's enough to digest for this month. Next time we'll see how to get things moving.*



Al's beat

ALAN McLACHLAN completes his megagame Smiley Hunt!

EVENIN' all. If you followed last month's session with the Smileys you should have a listing which matches Program I identically.

```
10 REM Al's Smiley Hunt
20 MODE 0
30 DIM box (9,9)
40 WINDOW #0,1,20,6,25:INK 0,0
50 WINDOW #1,1,20,1,5:PAPER #1,3:CLS
#1
60 turns=1:smileys=0
70 GOSUB 700 :REM draw/init grid
75 GOTO 75
700 REM draw grid
705 CLS #0
710 PEN #0,1:FOR boxx = 5 TO 14:FOR b
oxy = 6 TO 15
720 LOCATE #0,boxx,boxy:PRINT #0,CHR$
(233)
725 box(boxx-5,boxy-6)=0
730 NEXT boxy:NEXT boxx
740 FOR numx=5 TO 14:PEN #0,2:LOCATE
#0,numx,4:PRINT #0,MID$(STR$(numx-5),
2);:NEXT numx
750 FOR numy= 6 TO 15:LOCATE #0,16,nu
my:PRINT MID$(STR$(numy-6),2,1);:NEXT
 numy
760 PEN #0,1:LOCATE #0,10,2:PRINT #0,
 "x"
770 LOCATE #0,18,11:PRINT #0,"y"
775 FOR count=1 TO 1000:NEXT count
780 RETURN
```

Program I

We can now progress further and make the game a little more interesting.

First of all add lines 65 and 67:

```
65 LOCATE #1,3,3:PEN #1,1:PRINT #1,"A
l's Smiley Hunt"
67 FOR count=1 TO 1000:NEXT count
```

They make crafty use of the upper window while the grid is being drawn in the lower one.

If you're anything like me and enjoy seeing your name on the screen, include your own version of line 65 – be my guest, but don't you dare tell your grannie that you wrote the program all on your own.

Right, now that we've indulged ourselves a little, type in line 80:

```
80 GOSUB 800:REM position smileys
```

This calls the subroutine starting at line 800 which uses the random number generator to hide 10 Smileys within our array *box ( )*.

A FOR ... NEXT loop first of all places random numbers in *smileyx* and *smileyy* – the coordinates of our smiling face. Then using line 840 we place 1s in those elements in our array.

Line 830 checks to see whether any selected element already has 1 in it, and if so, sends the program back until an array element is encountered with a 0.

```
800 REM position Smileys
810 FOR count = 1 TO 10
820 smileyx = INT(RND*10):smileyy = I
NT(RND*10)
830 IF box(smileyx,smileyy)=1 THEN GO
TO 820
840 box(smileyx,smileyy)=1
850 NEXT count
860 RETURN
```

You can check whether your Smiley generator has worked by first of all removing line 75 and replacing it with "dummy" line 85:

```
85 GOTO 85
```

Now type in line 842, which is a temporary line and will be removed later: 350

```
842 LOCATE #0,smileyx+5,smileyy+6:
PRINT #0,"*"
```

If you run the program you should find 10 asterisks in the grid boxes showing the locations of the "hidden" Smileys. You can leave this line in for a while as it will be useful later for testing purposes.

OK, we've drawn the grid, initialised it, and hidden the Smileys. We are now ready for the input routine.

Remove line 85 and type in line 90:

```
90 GOSUB 900:REM take input
```

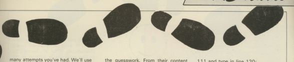
This, as you can see, calls the subroutine starting at line 900 and deals solely with your input to the computer.

It prints two lines of text at lines 901 and 903 and then uses the INKEY$ command to await your

input. Lines 910 to 913 first of all validate your input, only accepting numbers between 0 and 9.

We assign the numeric value of the input string to the variables *guessx* and *guessy,* then MID$ strips the spaces from the numbers, which are then printed with a comma already between them.

Line 915 simply adds 1 to the variable *turns* to keep tabs on how



many attempts you've had. We'll use this later to print out a result.

```
900 REM input
901 CLS #1:PEN #1,0:LOCATE #1,3,2:PRI
NT #1,"Guess Number:";turns:PRINT CHR
$(7)
902 FOR count =1 TO 1000:NEXT count
903 LOCATE #1,3,4:PEN #1,2:PRINT #1,"
Input x,y - ";
910 k$="":WHILE k$<"0" OR k$>"9":k$=I
NKEY$:WEND
911 guessx=VAL(k$):PRINT #1,MID$(k$,1
,1);",";
912 k$="":WHILE k$<"0" OR k$>"9":k$=I
NKEY$:WEND
913 guessy=VAL(k$):PRINT #1,MID$(k$,1
,1)
915 turns=turns+1
920 RETURN
```

Unfortunately you can't really check whether this routine is working correctly at this stage without a routine to process the information that you are inputting. Therefore let's continue by typing in line 100:

```
100 GOSUB 1000:REM check for smiley
```

This calls the next subroutine at line 1000, checking our input to see whether we have found a Smiley or not. Line 1030 does quite a lot here. It first of all checks to see if there is a 1 in the array element chosen.

If we have a bullseye, we go immediately to a subroutine at line 2000 where a Smiley face, CHR$(224) is printed at the location, and a suitably-triumphant noise is generated.

Then *smileys* is incremented by one, and finally a 2 is placed in the array element to show that this particular location has been used. Line 1001 checks for this number 2 and displays a message to that effect.

Should we fail to find a Smiley, a blank space is printed at the location and the line and column of that guess are checked to see if either contains a Smiley, in order to facilitate clues.

The clues are selected by setting *flagx* or *flagy* to 1 according to whether a column, or row, or both actually hold an undiscovered Smiley. These clues are intended to remove the guesswork. From their content you should be able to plan your next choice.

```
1000 REM check for hits
1001 IF box(guessx,guessy)=2 THEN LOC
ATE #0,1,18:PEN #1,1:PRINT #0,"You've
 had that one!":FOR count = 1 TO 1000
:NEXT count:LOCATE #0,1,18:PRINT #0,S
PACE$(20):RETURN
1010 flagx=0:flagy=0
1030 IF box(guessx,guessy)=1 THEN GOS
UB 2000 ELSE LOCATE #0,guessx+5,guess
y+6:PRINT #0," ":FOR z= 1 TO 10:SOUND
 132,1000,5,5:NEXT z
1040 FOR x= 0 TO 9:IF box(x,guessy)=1
 THEN flagy=1
1045 NEXT x
1050 FOR y= 0 TO 9:IF box(guessx,y)=1
 THEN flagx=1
1055 NEXT y
1070 IF flagx=1 AND flagy=0 THEN LOCA
TE #0,1,18:PRINT #0,"Right column"
1080 IF flagy=1 AND flagx=0 THEN LOCA
TE #0,1,18:PRINT #0,"Right row"
1090 IF flagy=1 AND flagx=1 THEN LOCA
TE #0,1,18:PRINT #0,"Right column & r
ow"
1095 FOR count= 1 TO 1000: NEXT count
:LOCATE #0,1,18:PRINT #0, SPACE$(20);
1100 RETURN
```

```
2000 LOCATE #0,guessx+5,guessy+6:PEN
#0,3:PRINT CHR$(224):FOR snd = 200 TO
 100 STEP -1:SOUND 1,snd,1:NEXT:smile
ys=smileys+1:box(guessx,guessy)=2:RET
URN
```

We can now check that the game is running correctly so far. Enter lines 85, 110 and 111:

```
85 WHILE smileys<10
110 WEND
111 STOP
```

Now you can test out your input and detection routines by running the program again. Once the Smiley target figure in line 85 is reached, the program stops with a Break at line 111.

When you are happy that everything is working all right, remove line 111 and type in line 120:

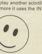
```
120 FOR count =1 TO 1000:NEXT count:C
LS #1:LOCATE #1,1,3:PEN #1,0:PRINT"Th
at's the lot !":FOR count =1 TO 1000:
NEXT count
```

This detects the last Smiley and prints a message on the screen to that effect.

The final subroutine is entered via line 130 and a REM statement in line 140 signifies the actual end of the program:

```
130 GOSUB 1300:REM print results
140 REM ***********************
```

Line 130 calls the subroutine at line 1300 which prints out the end result of your efforts. It simply takes the variable *turns* and prints it as part of a message. It then prompts to see if you wish to play another scintillating game. Once more it uses the INKEY$

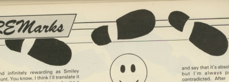


command to await your key press. Note also how I've used *upper$* to catch both upper and lower-case entries.

```
1300 REM print results
1310 CLS
1320 PEN #0,1:LOCATE 1,10:PRINT"That
took";turns;"guesses
1330 LOCATE 1,12:PRINT"Try again Y/N
?"
1340 k$="":WHILE k$<>"Y" AND k$<>"N":
k$=INKEY$:k$=UPPER$(k$):WEND
1350 IF k$="Y" THEN CLS #1:GOTO 60 EL
SE END
1360 RETURN
```

A positive response takes you back to line 60 carefully avoiding the arrays which must not be re-DIMmed. A negative response throws you unceremoniously out of the program, and quite rightly, too.

It's not every ,day you get the chance to play something as exciting



and infinitely rewarding as Smiley Hunt. You know, I think I'll translate it for the Plectrum, it might make me a mint!

If, having played the game a few times, you are happy that it is working correctly, you can remove line 842. You'll find it's a different game altogether now!

It may not be the most sparkling program in the world, but it does contain some interesting techniques. For example, the input routines could be used in any program – they probably already have – and the checking routines could always prove useful.

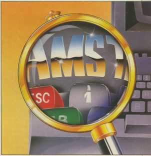
The important thing to me, though, is that you should have been able to follow it through line by line, subroutine by subroutine, and see how it was put together.

I'm going to stick my neck out now and say that it's absolutely bug-free – but I'm always prepared to be contradicted. After all, learning to de-bug programs is all part of the game, and I might have been crafty enough to have slipped one in for you to find . . .

Finally, we've made one or two alterations to the program as we've gone on. So, just in case someone has left a line in that should have been removed, here is the final listing of my megagame. You can now safely renumber it if you so wish.

```
10 REM Al's Smiley Hunt
20 MODE 0
30 DIM box (9,9)
40 WINDOW #0,1,20,6,25:INK 0,0
50 WINDOW #1,1,20,1,5:PAPER #1,3:CLS
#1
60 turns=1:smileys=0
65 LOCATE #1,3,3:PEN #1,1:PRINT #1,"A
l's Smiley Hunt"
67 FOR count=1 TO 1000:NEXT count
70 GOSUB 700 :REM draw/init grid
80 GOSUB 800:REM position smileys
85 WHILE smileys<10
90 GOSUB 900:REM take input
100 GOSUB 1000:REM check for smiley
110 WEND
120 FOR count =1 TO 1000:NEXT count:C
LS #1:LOCATE #1,1,3:PEN #1,0:PRINT"Th
at's the lot !":FOR count =1 TO 1000:
NEXT count
130 GOSUB 1300:REM print results
140 REM **********************
700 REM draw grid
705 CLS #0
710 PEN #0,1:FOR boxx = 5 TO 14:FOR b
oxy = 6 TO 15
720 LOCATE #0,boxx,boxy:PRINT #0,CHR$
(233)
725 box(boxx-5,boxy-6)=0
730 NEXT boxy:NEXT boxx
740 FOR numx=5 TO 14:PEN #0,2:LOCATE
#0,numx,4:PRINT #0,MID$(STR$(numx-5),
2);:NEXT numx
750 FOR numy= 6 TO 15:LOCATE #0,16,nu
my:PRINT MID$(STR$(numy-6),2,1);:NEXT
 numy
760 PEN #0,1:LOCATE #0,10,2:PRINT #0,
 "x"
770 LOCATE #0,18,11:PRINT #0,"y"
775 FOR count=1 TO 1000:NEXT count
780 RETURN
800 REM position Smileys
810 FOR count = 1 TO 10
820 smileyx = INT(RND*10):smileyy = I
NT(RND*10)
830 IF box(smileyx,smileyy)=1 THEN GO
TO 820
840 box(smileyx,smileyy)=1
850 NEXT count
860 RETURN
900 REM input
901 CLS #1:PEN #1,0:LOCATE #1,3,2:PRI
NT #1,"Guess Number:";turns:PRINT CHR
$(7)
902 FOR count =1 TO 1000:NEXT count
903 LOCATE #1,3,4:PEN #1,2:PRINT #1,"
Input x,y - ";
910 k$="":WHILE k$<"0" OR k$>"9":k$=I
NKEY$:WEND
911 guessx=VAL(k$):PRINT #1,MID$(k$,1
,1);",";
912 k$="":WHILE k$<"0" OR k$>"9":k$=I
NKEY$:WEND
913 guessy=VAL(k$):PRINT #1,MID$(k$,1
,1)
915 turns=turns+1
920 RETURN
1000 REM check for hits
1001 IF box(guessx,guessy)=2 THEN LOC
ATE #0,1,18:PEN #1,1:PRINT #0,"You've
 had that one!":FOR count = 1 TO 1000
:NEXT count:LOCATE #0,1,18:PRINT #0,S
PACE$(20):RETURN
1010 flagx=0:flagy=0
1030 IF box(guessx,guessy)=1 THEN GOS
UB 2000 ELSE LOCATE #0,guessx+5,guess
y+6:PRINT #0," ":FOR z= 1 TO 10:SOUND
 132,1000,5,5:NEXT z
1040 FOR x= 0 TO 9:IF box(x,guessy)=1
 THEN flagy=1
1045 NEXT x
1050 FOR y= 0 TO 9:IF box(guessx,y)=1
 THEN flagx=1
1055 NEXT y
1070 IF flagx=1 AND flagy=0 THEN LOCA
TE #0,1,18:PRINT #0,"Right column"
1080 IF flagy=1 AND flagx=0 THEN LOCA
TE #0,1,18:PRINT #0,"Right row"
1090 IF flagy=1 AND flagx=1 THEN LOCA
TE #0,1,18:PRINT #0,"Right column & r
ow"
1095 FOR count= 1 TO 1000: NEXT count
:LOCATE #0,1,18:PRINT #0, SPACE$(20);
1100 RETURN
1300 REM print results
1310 CLS
1320 PEN #0,1:LOCATE 1,10:PRINT"That
took";turns;"guesses
1330 LOCATE 1,12:PRINT"Try again Y/N
?"
1340 k$="":WHILE k$<>"Y" AND k$<>"N":
k$=INKEY$:k$=UPPER$(k$):WEND
1350 IF k$="Y" THEN CLS #1:GOTO 60 EL
SE END
1360 RETURN
2000 LOCATE #0,guessx+5,guessy+6:PEN
#0,3:PRINT CHR$(224):FOR snd = 200 TO
 100 STEP -1:SOUND 1,snd,1:NEXT:smile
ys=smileys+1:box(guessx,guessy)=2:RET
URN
```

Create magnificent magnified characters



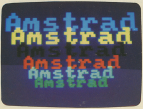
. . . and give your display screens more impact with this great graphics utility

by GLYNNE DAVIES

AS its name suggests, this program allows you to design a screen of enlarged characters for use in titles or advertising media.

The whole of the character set plus the graphic symbols can be used, or you can re-design a complete new character set.

Magnified Characters is compatible with Easydraw – see *Computing with the Amstrad,* June 1985 – so you can display large text on designs created with that program. The



sub-routine from line 820 can be used in your own program and a demonstration of this method is given in the title sequence.

After the title sequence demonstration – which by taking out line 130 can be bypassed to allow quick access to the program – there are a series of on-screen prompts which should be dealt with as follows:

Load character set Y/N: A special character set can be loaded into memory by selecting Y. This could have been designed using Character Maker – see *Computing with the Amstrad,* May 1985 – or re-defined with this program and saved to tape or disc.

Easydraw format Y/N: If you select the Easydraw format the available screen is smaller, surrounded by a red rectangle. Both options can be loaded into Easydraw but if you have not selected the Easydraw format you may lose part of your characters.

Enter Mode 0, 1 and 2: Any mode can be selected, but if Mode 2 is chosen you have only one plotting colour.

Download screen Y/N: The program reserves 16k of memory for the screen display. If you have loaded the memory with a display you can reproduce it on the screen.

Type in word: Type in a series of letters, or alternatively, by first pressing the small Enter key, the Ascii code above 126 and below 256 can be entered, producing the appropriate graphic character. When the word is complete press Enter. If Enter is pressed before a word is typed the message "Another word Y/N" will appear. If you reply N to this prompt the options load/save become available.

Arrow keys – Enter: The arrow keys control a cursor showing where the top left of the first letter will be drawn. If this is moved to the far left of the screen the word will be drawn in the centre. When the position is correct press Enter.

Scale 1 to N: Depending upon the length of the word a scale choice is given. Scale 1 is smaller than the normal character set and only readable in Mode 2, although this scale can be used for underlining.

Slope Y/N: This option produces sloping characters.

Shadow Y-N: If shadow is selected, the word can be drawn with a 3-D effect. The arrow keys are used to position the shadow, which can be set to any of eight positions. To produce a shadow above the word, press the up arrow key twice. A beep is made with each arrow key press and two beeps must be produced.

Pen 0 to N: Choose the drawing pen number. If numbers 16 or 17 are chosen in Mode 0 these are multicoloured pens – 4 and 5 in Mode 1. When the shadow option is chosen a background – shadow – pen is also selectable.

xt=   yt= : As the word is drawn on the screen, the start position is displayed as xt and yt. Make a note of these positions if you are going to use the sub-routine at line 820 in your own program.

Another word Y/N or D to delete word: Enter Y to draw another word, N to move on to the next options or D to remove the last word drawn. If D is selected the last word will be deleted.

Save screen Y/N or F character save: Enter Y and the screen is saved into the reserved memory, N to move on or F to save to tape or disc the character set – file – that you have re-defined. An option to save the contents of the reserved memory is given after a screen save.

Clear screen Y/N or R to re-define: Enter Y to erase the whole screen, N to move on or R to re-define a character. If R is pressed, prompts will ask for eight numbers from 0 to 255. Design the character you require on 64 square grid and then enter the eight numbers from top to bottom.

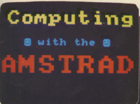
As you enter the numbers the space is re-defined so you can see the build-up. If the shape is correct when finished ehter Y and enter the Ascii number of the character you wish to re-define. The space will clear to normal.

Load to memory Y/N or C for characters: A previously saved screen from this program or Easydraw can be loaded into the reserved memory by Entering Y. N will move back to the main program or C will allow you to load in a new character set.

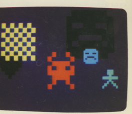
The saved screen can be loaded on to the screen without a program using LOAD "filename", &C000.

If you are going to use the sub-routine in your own program, build up the screen using this program making notes of the xt and yt positions, pens used and so on. Use the 'SYMBOL AFTER 32 at the beginning of your program and the sub-routine from lines 820 to 1350 and follow the demonstration example.

```
10 REM *** MAGNIFIED CHARACTERS ***
20 REM
30 REM *** by Glynne Davies ***
40 REM
50 REM(c)Computing with the Amstrad
60 IF HIMEM=24410 THEN INPUT "Press D
 for disk";y$:IF y$="d" OR y$="D" THE
N topmem=40955:ELSE topmem=42239
70 IF HIMEM=24410 THEN GOTO 110:REM a
llow re-run
80 IF HIMEM=43903 THEN topmem=42239 E
LSE topmem=40955:REM * tape or disc
90 SYMBOL AFTER 32:REM this downloads
 character set into ram at 42240 or 4
0956(disc)
100 MEMORY 24410:REM * reserve memory
 for screen dump and disc drive *
110 GOSUB 2070:REM * load machine cod
e for memory dump *
120 REM ************************
130 GOSUB 2190:REM demonstration,remo
ve for fast access to program
140 CLS:y=200:x=1:left=0:bottom=32
150 LOCATE 10,10:INPUT #0,"Load chara
cter set Y/N";y$:IF y$="y" OR y$="Y"
THEN GOSUB 2140
160 CLS
170 LOCATE 9,12:INPUT "Easydraw optio
n Y/N";y$
180 IF y$="Y" OR y$="y" THEN bottom =
48:left=64:e=1
190 CLS
200 REM ** screen format **
210 LOCATE 9,12:INPUT "Enter mode 0
1 or 2";screenmode:IF screenmode>2 TH
EN GOTO 210
220 MODE screenmode
230 IF y$="d" OR y$="D" THEN GOTO 510
:REM return to question
240 IF screenmode=2 THEN screenmode=4
:movx=1:choicepen=1:cp=1
250 IF screenmode=1 THEN screenmode=2
:movx=2:choicepen=3:cp=1
260 IF screenmode=0 THEN screenmode=1
:movx=4:choicepen=15:cp=14
270 WINDOW #0,1,20*screenmode,24,25
280 ORIGIN left,bottom,left,640,400,b
ottom
290 IF e=1 THEN DRAW 574,0,3:DRAW 574
,350,3:DRAW 0,350,3:DRAW 0,0,3
295 PRINT CHR$(2)
300 WHILE 1
310 CLS #0
320 LOCATE #0,1,1:INPUT #0,"Download
screen Y/N";y$:IF y$="Y" OR y$="y" TH
EN CALL 24423
330 CLS #0:LOCATE #0,1,1:PRINT "Type
in word";
340 mag$=""
350 ON ERROR GOTO 360
```



Computing
with the
AMSTRAD

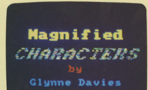


VARIABLES

topmen	Start position of character set.
screenmode	Mode and window multiplier.
movx	Pixel movement.
choicepen	Pens available.
letter$	Input character.
mag$	Word to be magnified.
mag	Scale choice.
slope	Letter slope: 1=on, 0=off.
shadow	1=on, 0=off.
enlarge	Maximum possible scale.
xt	Horizontal graphics cursor position.
yt	Vertical graphics position.
rubx ruby	Erase coordinates.
blockcol	Pen+2 coloured blocks.
random	Pen+1 multicoloured blocks.
bnum	Number of character byte.
num$	Graphics character Ascii code in a string.
p	Pen number.
bit	Re-defined character byte.
memno	Re-defined Ascii character number.

```
360 word=0
370 WHILE word=0
380 letter$=INKEY$:IF letter$="" THEN
 GOTO 380
390 IF INKEY(23)=128 THEN GOTO 380
400 IF INKEY(6)=0 THEN SOUND 1,110,10
:GOSUB 1530
410 IF INKEY(18)=0 THEN word=1:letter
$="":GOTO 410:REM wait until Enter is
 released then finish
420 IF INKEY(79)=0 THEN l=LEN(mag$):m
ag$=LEFT$(mag$,l-1):letter$="":LOCATE
 #0,1,2:PRINT #0,SPACE$(20);
430 IF letter$<>"" THEN mag$=mag$+let
ter$:letter$=""
440 LOCATE #0,1,2:PRINT #0,mag$
450 IF mag$="" THEN GOTO 510:REM goto
 options
460 WEND:word=0
470 GOSUB 1360:REM * word position ch
oce by cursor *
480 GOSUB 620:REM check length give m
agnification options
490 GOSUB 820:REM read string and dra
w magnified
500 blockcol=0:random=0
510 CLS #0:LOCATE #0,1,1:INPUT #0,"An
other word Y/N or D to delete word";y
$:IF y$="Y" OR y$="y" THEN GOTO 610:R
EM repeat loop
520 IF y$="d" OR y$="D" THEN xt=rubx:
yt=ruby:p=0:GOSUB 820:IF shadow=1 THE
N xt=shadxt:yt=shadyt:GOSUB 820:shado
w=0:REM rubout
530 IF y$="d" OR y$="D" THEN GOTO 510
:REM return to question
540 CLS #0:LOCATE #0,1,1:INPUT #0,"Sa
ve screen Y/N or F character save";y$
:IF y$="Y" OR y$="y" THEN GOSUB 1880
550 IF y$="f" OR y$="F" THEN GOSUB 25
50
560 CLS #0:LOCATE #0,1,1:INPUT #0,"Cl
ear screen Y/N or  R to redefine";y$:
IF y$="Y" OR y$="y" THEN CLG
570 IF y$="r" OR y$="R" THEN GOSUB 23
00
580 IF m=1 THEN DRAW 574,0,3:DRAW 574
,350,3:DRAW 0,350,3:DRAW 0,0,3
590 CLS #0:LOCATE #0,1,1:INPUT #0,"Lo
ad to memory Y/N  C for characters";y
$:IF y$="Y" OR y$="y" THEN GOSUB 2020
600 IF y$="c" OR y$="C" THEN GOSUB 21
40
610 slope=0:shadow=0:WEND
620 REM ** magnification options **
630 length=LEN(mag$)
640 size=641-left-x
650 enlarge=size\(length*8)
660 IF enlarge*8>y THEN size=size-10:
GOTO 650
```

```
670 CLS #0:LOCATE 1,1:PRINT "Scale 1
to ";enlarge;:INPUT enlarge$:IF enlar
ge$=" " OR enlarge$="" THEN 670
680 IF ASC(enlarge$)<49 OR ASC(enlarg
e$)>57 THEN GOTO 670
690 mag=VAL(enlarge$)
700 xt=x:yt=y
710 IF x<2 THEN xt=(640-left-length*m
ag*8)/2
720 rubx=xt:ruby=yt
730 CLS #0:LOCATE #0,1,1:INPUT "Slope
 Y/N";y$:IF y$="Y" OR y$="y" THEN slo
pe=1
740 IF choicepen =1 THEN GOTO 760
750 CLS #0:LOCATE #0,1,1:INPUT "Shado
w Y/N";y$:IF y$="Y" OR y$="y" THEN GO
SUB 1640
760 CLS #0:PRINT "Pen 0 to";choicepen
+2:INPUT p$:IF p$=" " OR p$="" THEN 7
60
770 IF ASC(p$)<48 OR ASC(p$)>57 THEN
GOTO 760
780 p=VAL(p$)
790 IF p>choicepen+2 THEN 760
800 IF mag=1 AND bnum>128 THEN bnum=b
num-128
810 RETURN
815 REM ******************************
**
820 REM *** read string and magnify *
**
825 REM ******************************
**
830 IF p=choicepen+2 THEN blockcol=1:
choice=choicepen+1
840 IF choicepen=15 THEN choice=12
850 IF nodisplay=1 THEN GOTO 870:REM
miss xt,yt printing in display mode
860 CLS #0:LOCATE #0,1,1:PRINT #0,"xt
=";xt;" yt=";yt
870 FOR letter=1 TO LEN(mag$)
880 IF slope=1 THEN xt=xt+8*movx
890 MOVE xt,yt
900 letter$=MID$(mag$,letter,1)
910 position=ASC(letter$)
920 mem=topmem+1+(position-32)*8
930 FOR count=mem TO mem+7
940 bnum=PEEK(count)
950 IF slope=1 THEN xt=xt-movx
970 IF bnum>=128 THEN GOSUB 1260:bnum
=bnum-128
1000 xt=xt+mag:MOVE xt,yt
1010 IF bnum>=64 THEN GOSUB 1260:bnum
=bnum-64
1030 xt=xt+mag:MOVE xt,yt
1040 IF bnum>=32 THEN GOSUB 1260:bnum
=bnum-32
1060 xt=xt+mag:MOVE xt,yt
1070 IF bnum>=16 THEN GOSUB 1260:bnum
=bnum-16
1090 xt=xt+mag:MOVE xt,yt
1100 IF bnum>=8 THEN GOSUB 1260:bnum=
bnum-8
1120 xt=xt+mag:MOVE xt,yt
1130 IF bnum>=4 THEN GOSUB 1260:bnum=
bnum-4
1150 xt=xt+mag:MOVE xt,yt
1160 IF bnum>=2 THEN GOSUB 1260:bnum=
bnum-2
1180 xt=xt+mag:MOVE xt,yt
1190 IF bnum=1 THEN GOSUB 1260
1200 bnum =0
1210 xt=xt-(7*mag):yt=yt-mag:MOVE xt,
yt
1220 NEXT
1230 xt=xt+(8*mag):yt=yt+(8*mag)
1240 NEXT
1250 RETURN
1260 REM *** draw blocks ***
1265 IF blockcol=1 THEN p=(p+1) MOD c
hoice:IF p=0 THEN p=1
1270 IF p=choicepen+1 THEN random=1
1275 xl=mag-movx
1280 WHILE inc<mag
1290 IF random=1 THEN p%=RND(1)*choic
epen:p=p%:IF p=0 OR p>=14 THEN p=1
1300 DRAWR xl,0,p
1310 inc=inc+1
1320 MOVE xt,yt-inc
1330 WEND
1340 MOVE xt,yt:inc=0
1350 RETURN
1355 REM ***** end of sub-routine ***
**
1360 REM * cursor position choice *
1370 tes=TEST(x,y)
1380 LOCATE #0,1,2:PRINT #0,"Arrow ke
ys - Enter"
1390 WHILE curset=0
1400 IF INKEY(1)=0 THEN PLOT x,y,tes:
tes=TEST(x+movx,y):x=x+movx:PLOT x,y,
cp
1410 IF INKEY(8)=0 THEN PLOT x,y,tes:
tes=TEST(x-movx,y):x=x-movx:PLOT x,y,
cp
1420 IF INKEY(0)=0 THEN PLOT x,y,tes:
tes=TEST(x,y+2):y=y+2:PLOT x,y,cp
1430 IF INKEY(2)=0 THEN PLOT x,y,tes:
tes=TEST(x,y-2):y=y-2:PLOT x,y,cp
1440 IF x<0 THEN x=0:PLOT 0,y,cp
1450 IF x>639-left THEN x=639-left:PL
OT 639-left,y,cp
1460 IF y>398-bottom THEN y=398-botto
m:PLOT x,398-bottom,cp
1470 IF y<1 THEN y=1:PLOT x,1,cp
1480 IF INKEY(18)=0 THEN PLOT x,y,tes
:curset=1
1490 IF INKEY(18)=0 THEN GOTO 1490:RE
M hold on line until Enter is release
d
1500 WEND
1510 curset=0
1520 RETURN
1530 REM * Graphic shapes input *
1540 CALL &BB18:IF INKEY(6)=0 THEN GO
TO 1540
1550 LOCATE #0,1,1:PRINT #0,SPACE$(20
)
1560 LOCATE #0,1,1:INPUT #0,"Enter nu
mber";num$:IF num$="" THEN GOTO 1560
1570 IF LEN(num$)<>3 THEN GOTO 1550
1580 IF VAL(num$)<127 OR VAL(num$)>25
5 THEN GOTO 1550
1590 LOCATE #0,1,1:PRINT #0,SPACE$(20
):LOCATE #0,1,1:PRINT #0,"Type in wor
d"
1600 letter$=CHR$(VAL(num$))
1610 IF INKEY(18)=0 THEN GOTO 1610
1620 IF INKEY(6)=0 THEN GOTO 1610
1630 RETURN
1640 REM *** Shadow ***
1650 yyt=yt:xxt=xt
1660 IF mag<4 THEN reduce=1 ELSE redu
ce=2
1670 shadow=1
1680 CLS #0:LOCATE #0,1,1:PRINT "Two
arrow keys"
1690 WHILE ks<1
1700 IF INKEY(0)=0 THEN SOUND 1,100,2
5:yt=yt+mag\reduce:ks=ks+0.5
1710 IF INKEY(2)=0 THEN SOUND 1,100,2
5:yt=yt-mag\reduce:ks=ks+0.5
1720 IF INKEY(8)=0 THEN SOUND 1,100,2
5:xt=xt-mag\reduce:ks=ks+0.5
1730 IF INKEY(1)=0 THEN SOUND 1,100,2
5:xt=xt+mag\reduce:ks=ks+0.5
1740 FOR n=1 TO 300:NEXT
1750 WEND
1760 WHILE hold<1:REM hold until rele
ased
1765 IF INKEY(0)<>0 AND INKEY(2)<>0 A
ND INKEY(8)<>0 AND INKEY(1)<>0 THEN h
old=1
1770 WEND
1780 hold=0
1800 ks=0:shadxt=xt:shadyt=yt
1810 FOR n=1 TO 100:NEXT:CALL &BB18
1815 PRINT:PRINT
1820 CLS #0:LOCATE #0,1,1:PRINT #0,"S
hadow 0-";choicepen+2;" ";:INPUT #0,p
$:IF p$=" " OR p$="" THEN GOTO 1820
1830 IF ASC(p$)<48 OR ASC(p$)>57 THEN
 GOTO 1820
1840 p=VAL(p$)
```



```
1850 GOSUB 820:REM * background lette
rs *
1860 xt=xxt:yt=yyt:random=0
1870 RETURN
1880 REM * Save screen to memory *
1890 SOUND 1,100,20:MOVE 0,0
1900 CLS #0
1910 IF e=1 THEN DRAW 574,0,3:DRAW 57
4,350,3:DRAW 0,350,3:DRAW 0,0,3
1920 CALL 24411 : REM m/c code memory
 dump
1930 CALL 24423:REM dump back to scre
en to check
1940 LOCATE #0,1,1:INPUT #0,"Save mem
ory Y/N";y$:IF y$="Y" OR y$="y" THEN
GOSUB 1970
1950 CLS #0
1960 RETURN
1970 REM * save screen to tape or dis
c *
1980 SPEED WRITE 1
1990 CLS #0:LOCATE #0,1,1:PRINT #0,"P
lease print name":INPUT name$
2000 SAVE name$,B,&5F75,&4000
2010 RETURN
2020 REM *** load to memory from tape
 or disc ***
2030 CLS #0:LOCATE #0,1,1:INPUT #0,"N
ame";picname$
2040 LOAD picname$,&5F75
2050 RETURN
2060 REM *** machine code for screen
dump ***
2070 FOR n=24411 TO 24434
2080 READ x
2090 POKE n,x
2100 NEXT n
2110 RETURN
2120 DATA 1,0,64,33,0,192,17,117,95,2
37,176,201
2130 DATA 1,0,64,33,117,95,17,0,192,2
37,176,201
2140 REM * load character set *
2150 CLS #0:LOCATE #0,1,1:INPUT #0,"C
har. files name";fil$
2160 IF LEN(fil$)>8 THEN GOTO 2150
2170 LOAD fil$,topmem+1
2180 RETURN
2190 REM *** Demonstration of use wit
hin own program ***
2200 MODE 0:choicepen=15:ORIGIN 0,32,
0,640,400,32:nodisplay=1
2210 mag$="Magnified":mag=6:slope=0:p
=1:xt=104:yt=270:GOSUB 820:REM * draw
 scale 6 in yellow *
2220 mag$="CHARACTERS":mag=7:slope=1:
movx=4:p=5:xt=28:yt=200:GOSUB 820:REM
 background
2230 mag$="CHARACTERS":mag=7:slope=1:
movx=4:p=16:xt=32:yt=196:GOSUB 820:RE
M foreground
2240 random=0:REM turns of special co
lour
2250 mag$="by":slope=0:mag=4:xt=288:y
t=120:p=3:GOSUB 820
2260 mag$="Glynne Davies":mag=4:xt=11
2:yt=60:p=15:GOSUB 820
2270 FOR N=1 TO 500:NEXT
                         3000
2280 MODE 1
2290 RETURN
2300 REM ** simple character redefine
 **
2310 CLS #0:LOCATE #0,1,1:INPUT #0,"R
edefine character  Y/N";y$:IF y$="y"
OR y$="Y" THEN GOTO 2320 ELSE RETURN
2320 FOR define=1 TO 8
2330 CLS #0:LOCATE #0,1,1:PRINT #0,"N
umber ";define;" ";:INPUT #0,bit$
2340 IF bit$="" OR bit$=" " THEN GOTO
 2330
2350 IF LEN(bit$)>3 THEN GOTO 2330
2360 IF ASC(bit$)<48 OR ASC(bit$)>57
THEN GOTO 2330
2370 bit=VAL(bit$):IF bit>255 THEN GO
TO 2330
2380 POKE topmem+define,bit
2390 NEXT
2400 CLS #0:INPUT #0," Y/N";y$:IF y$=
"y" OR y$="Y" THEN GOTO 2410 ELSE GOT
O 2480
2410 CLS #0:INPUT #0,"Enter(CHR$)numb
er";memno
2420 IF memno>255 OR memno<32 THEN GO
TO 2410
2430 memno=memno-32
2440 FOR n=1 TO 8
2450 look=PEEK(topmem+n)
2460 POKE topmem+(memno*8)+n,look
2470 NEXT
2480 REM clear space chr$ 32
2490 FOR n=1 TO 8
2500 POKE topmem+n,0
2510 NEXT
2520 CALL &BB1B
2530 CLS #0:INPUT "Another Y/N";y$:IF
 y$="y" OR y$="Y" THEN GOTO 2320:REM
repeat redefinition
2540 RETURN
2550 REM * save character file *
2560 CLS #0:LOCATE #0,1,1:INPUT #0,"S
ave characters Y/N";y$:IF y$="y" OR y
$="Y" THEN GOTO 2570 ELSE RETURN
2570 CLS #0:LOCATE #0,1,1:INPUT #0,"F
ile name";charfile$
2580 IF LEN(charfile$)>8 THEN GOTO 25
70
2590 CLS #0
2600 SPEED WRITE 1
2610 SAVE charfile$,B,topmem+1,1792
2620 CALL &BB1B
2630 RETURN
```

Let your fingers do the talking...with this specia

Now you can teach your Amstrad to talk!

YOU can add an exciting new dimension to computing with your Amstrad – with the help of this remarkable new product from dk'tronics.

It comes complete with the latest and very versatile speech chip, a powerful stereo amplifier and two high-quality 4in speakers, specially designed to match the Amstrad CPC464.

And because this is a special reader offer it comes to you at £5 off the normal retail price of £39.95!

Fitting it is simplicity itself. All you have to do is to plug the synthesiser's interface into the floppy disc port at the back of the Amstrad and the jack plug into the stereo socket – and away you go!

With its volume and balance controls you will find you can put dramatic realism into the sound output of your Amstrad. All sounds that previously came from the Amstrad's 1½in mono speakers are now sent out via the interface in stereo.

So even when you're not using it as a speech synthesiser, it can bring startling depth and drama to the music and sound effects of all your favourite games!

How it works

AT the heart of the dk'tronics speech synthesiser lies an incredibly powerful chip that has split the English language into its component parts – or allophones as they are known.

Altogether there are 59 allophones and five pauses stored in the speech chip's internal ROM. These can be combined to create a virtually unlimited vocabulary.

The potential of this chip is realised by dk'tronic's sophisticated, yet simple to use software. The brilliant program design enables the Amstrad to actually speak the words you type, in straightforward English, without having to resort to complicated phonetic spelling or difficult programming techniques.

Written to be as user friendly as possible, the synthesiser adds eight powerful commands to Amstrad Basic.

If you prefer complete control over your programs, though, full details are given for Basic and machine code programmers to exploit the tremendous scope of the synthesiser without using the software supplied.

In fact this system supports four different modes of use.

The first mode allows you to sound words using only the Amstrad's normal Basic commands. However, as you get more ambitious with your speech, a second mode is provided. This gives eight extra commands to use from Basic, making using the synthesiser even easier.

The third mode is the text to speech converter. When this is in operation speech can be typed in using normal English and the Amstrad does the rest. There's no need to work out the allophones as in the other two modes – the Amstrad does it for you.

As if all this wasn't enough there's the fourth mode. This has the synthesiser converting whatever appears on the screen into speech. Using this, you can literally listen to your listings!

These are the sounds – and pauses – you can create on your Amstrad

A	AE	26	fat	F	FF	40	fire	NG	NG	44	bans	TH	TH	29	thin
A	AY	20	great	G	GG2	61	go	O	O	23	cot	TH	DH1	18	they
AI	AIR	47	hair	IG	GG3	34	wig	O	OW	53	snow	TH	DH2	54	bathe
AR	AR	59	farm	GU	GG1	36	guest	O	UW1	22	do	U	AX	15	succeed
AU	AO	24	aught	GE	ZH	38	beige	OO	UW2	31	food	UH	OO	30	cook
B	BB1	28	rib	H	HH1	27	he	OR	OR	58	store	U	YY1	49	compute
B	BB2	63	big	H	HH2	57	hoe	OU	OW	32	ouch	V	VV	35	even
C	KK1	42	common	I	IH	12	fitting	OY	OY	5	toy	W	WW	46	wool
'C	C	8	uncle	I	Y	6	sky	P	PP	9	pub	WH	WH	48	whig
K	KK2	41	sky	IR	ER2	52	bird	R	RR1	14	real	Y	YY2	25	yes
CH	CH	50	church	J	JH	10	jury	R	RR2	39	brain	Z	ZZ	43	zoo
D	DD1	21	could	L	LL	45	luck	R	YR	60	fear	PA1	PA1	0	10 mS
D	DD2	33	do	L	EL	62	angle	S	SS	55	sat	PA2	PA2	1	30 mS
E	EH	7	bend	M	MM	16	milk	SH	SH	37	shirt	PA3	PA3	2	50 mS
E	EE	19	see	N	NN1	11	earn	T	TT1	17	its	PA4	PA4	3	100 mS
ER	ER	51	cater	N	NN2	56	no	T	TT2	13	top	PA5	PA5	5	200 mS

Column 1: Sound. Column 2: Allophone name. Column 3: Allophone number. Column 4: Example word.

ial Computing with the Amstrad reader offer!

Save £5 *(plus FREE post and packing)*
Offer price only £34.95

Look at what this package offers you:

- ★ Speech synthesiser with almost unlimited vocabulary
- ★ Easy-to-use commands – it accepts normal English words
- ★ Built-in stereo amplifier with twin speakers
- ★ Programs can run while the speed chip talks

Eight additional Basic commands

¦SPON	Speech on.
¦SPOF	Speech off.
¦FEED,n	Feed speech buffer direct.
¦FLUS	Clear speech and text buffers.
¦SPED,n	Speech speed.
¦OUTM,1	PRINT text to speech.
¦OUTM,2	Screen output to speech.
¦OUTM,3	Output to screen and speech.

Please send me the dk'tronics speech synthesiser for my Amstrad CPC464

□ I enclose cheque for £34.95 *(incl. VAT, p&p)* made payable to Database Publications Ltd.

I wish to pay by

□ Access Card No. ______________________

□ Visa Card No. ______________________

Expiry date / /

Signed ______________________

Name ______________________

Address ______________________

POST TO: Speech Synthesiser Offer, Database Publications,
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

CA9

Allow 28 days for delivery

TWO WAYS TO ENSURE YOU GET

Computing with the AMSTRAD

EVERY MONTH

1. Complete and mail subscription form on Page 77.
2. Hand this form to your newsagent.

Please reserve me a copy of Computing with the Amstrad magazine every month until further notice.

□ I will collect

□ I would like it delivered to my home.

Name

Address

Note to newsagent: Computing with the Amstrad should be obtainable from your local wholesaler, or contact Steve Fletcher, Circulation Manager on 0293 27053.

TAPE OR DISC

MORE

Provides 32 columns in mode 0, 64 or 32 in mode 1. 32, 64, 128, in mode 2 plus standard columns – characters unchanged – use on same line. Just call More X, Y, size when required. User designed characters with designer included, but supplied complete with file for instant use **£6.95**

CODATA

Codata is a revolutionary machine code utility aid. It works in parallel with any assembler or HEX Loader and enables fast machine code writing. Look out for rave reviews.
Disk £10.95 Tape £7.95

Trade and Export enquiries invited

BINARY BANANA
c/o Pitec Limited.
Park Crescent, Peterborough PE1 4BG.
Telephone: 0733 312120 ext. 26

THE FABULOUS
CASSETTE
50
FROM
cascade
THE NATIONAL NEWSPAPER MOPS MAIL ORDER PROTECTION SCHEME
YEAH. PLUS A FREE CALCULATOR WATCH
VALUE THATS OUT OF THIS WORLD.
50 SUPER COMPUTER GAMES ON ONE CASSETTE
YOURS FOR ONLY
£9.95
POSTAGE FREE
Worth over £10.00

FREE with every order for a cassette 50

20 Key, Multifunction Calculator Watch

- Normal 12 and 24 hour time mode
- Calender mode
- Calculator mode
- Alarm mode

CASSETTE 50 IS AVAILABLE ON
BBC A/B ATARI DRAGON ELECTRON
VIC-20 ORIC-1 ZX81 apple Atmos
Spectrum commodore 64 AMSTRAD

Amstrad CPC 464

Avoid the rockets and anti-aircraft guns and DYNAMITE the dam, only the most skilful pilots succeed. Find yourself in different scenario's in this original adventure to search for the KINGS ORB. NEMESIS a sophisticated version of Othello with many colourful options and levels of play.
To mention only three of the 50 games on the Amstrad Cassette 50, which includes arcade type games, war games, adventure games, logical, tactical, and educational games, a flight simulator, and a business strategy game. Cassette 50 features multicoloured and user-defined graphics, joystick compatability, sound and music utilising the amazing power of your Amstrad.

1. Maze Eater
2. Cyclons
3. Space Mission
4. Lunar Landing
5. Fantasy Land
6. Startrek
7. Radar Landing
8. Attacker
9. Tanks
10. Galactic Attack
11. Ivasive Action
12. Noughts and Crosses
13. King Orb
14. Play Your Cards Right
15. Creepy Crawley
16. Hangman
17. Old Bones
18. Thin Ice
19. 3D Ice
20. Colony 9
21. Force Field
22. Nim
23. Tunnel Escape
24. Barrel Jump
25. Trucking
26. Rally 3000
27. Sitting Target
28. Nemesis
29. Space Ship
30. Jet Flight
31. Phaser
32. Intruder
33. Inferno
34. Ghosts
35. Submarines
36. Space Base
37. Planets
38. Black Hole
39. Dynamite
40. Timebomb
41. Day at the Races
42. Space Search
43. Trivia Quiz
44. Rats
45. Tanker
46. Dungeon Adventure
47. Jet Mobile
48. High Rise
49. The Force
50. Exchange

EXPRESS DELIVERY-
ORDER NOW!
and get your FREE calculator watch with every Cassette 50

Name

Address

Post Code

Country

CA9-85

Postage FREE. Goods will be despatched within 7 days.

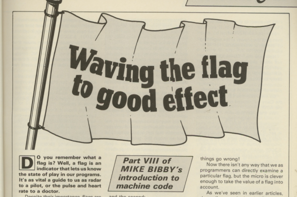
I enclose a cheque/postal order for £ made payable to Cascade Games Ltd.

or through any Post Office by TRANSCASH (Giro No. 655 6655) For even faster ordering when charging to Access, Barclaycard and Trustcard Visa use our 24 hour service. (0423) 504663

No.

COMMODORE 64	□	VIC 20	□	ORIC 1	□	BBC A/B	□
SPECTRUM	□	ELECTRON	□	ZX 81	□	APPLE	□
ATARI	□	ATMOS	□	DRAGON	□		
						AMSTRAD	□

Cascade Games Ltd.,
1-3 Haywra Crescent, Harrogate,
North Yorkshire, HG1 5BG. England.
Telephone: (0423) 504663 *Registered Number 1755554*



Waving the flag to good effect

Part VIII of MIKE BIBBY's introduction to machine code

DO you remember what a flag is? Well, a flag is an indicator that lets us know the state of play in our programs. It's as vital a guide to us as radar to a pilot, or the pulse and heart rate to a doctor.

Despite their importance, flags are essentially quite simple. You can think of them as a primitive numeric variable that's only allowed to have two values – 1 when we say the flag is set, and 0 when we say it's clear.

Another way to think of it is as a single bit register, since a single bit can only take the values 0 or 1.

Depending on what's happening in a program, the micro sets or clears the various flags. We first met flags when we were adding numbers. Because the largest number we could store in a byte was 255, when we have two numbers together to give an answer larger than 255 we hit problems.

As we've seen, going past 255 in a byte – register or memory – is rather like going round the clock on a car's odometer: You start again at zero.

So as far as the A register (the register ADD and SUB work on) is concerned:

255 + 1 = 0
255 + 2 = 1
254 + 3 = 1

and so on. Much the same thing happens when you try to go below zero in a subtraction.

Try Programs I and II if you don't believe me. The first does:

254 + 3 (&FE + &03)

and the second:

3 – 4 (&03 – &04)

In both cases the answer is stored at &2FF8, the first byte of Hexer's workspace.

Of course, the Z80 doesn't just ignore aberrant results when things go round the clock: It sets what's known as the Carry flag. That is, the Carry flag becomes 1. Conversely, if we haven't gone over (or under) the limit, the flag is cleared, to 0.

So, as far as the micro is concerned:

255+1=0	Carry flag set
254+1=255	Carry flag clear
0–0=0	Carry flag clear
3–4=255	Carry flag set

The Carry flag lets us know when things go wrong!

```
address  hex code  mnemonics
 3000    3E FE     LD A, &FE
 3002    C6 03     ADD A,&03
 3004    32 F8 2F  LD (&2FF8),A
 3007    C9        RET
```

Program I

```
address  hex code  mnemonics
 3000    3E 03     LD A, &03
 3002    D6 04     SUB &03
 3004    32 F8 2F  LD (&2FF8),A
 3007    C9        RET
```

Program II

Now there isn't any way that we as programmers can directly examine a particular flag, but the micro is clever enough to take the value of a flag into account.

As we've seen in earlier articles, depending on whether the Carry flag is set or clear, we can jump to different parts of our programs with instructions such as:

```
JP C    ( opcode &DA )
```

which is JumP with Carry set and:

```
JP NC   ( opcode &D2 )
```

which is JumP if the Carry flag is Not set.

```
address  hex code  mnemonics
 3000    3E ?1     LD A, ?1
 3002    C6 ?2     ADD A, ?2
 3004    32 F8 2F  LD (&2FF8),A
 3007    D2 0F 30  JP NC,&300F
 300A    3E 07     LD A,&07
 300C    CD 5A BB  CALL CharOut
 300F    C9        RET
```

Program III

Program III is a nice little bleeper! What it does is to add the two numbers ?1 and ?2. Of course, ?1,?2 are only labels – you're meant to insert the two bytes you want to add here – memory locations &3001, &3003, respectively, if you want to POKE them into memory.

The first two lines of Program III do the addition. The ADD instruction will set or clear the Carry flag appropriately to warn us if we've exceeded 255.

```
LD (&2FF8),A
```

then puts the answer at the beginning of Hexer's workspace, for later examination.

Now hear this – LD instructions do not affect any flags ever. (There are only two minor exceptions to this, LD A,R and LD A,I which we're never likely to meet.) They leave the flags completely alone. This is a very useful piece of information to bear in mind.

So even after this LD, the state of the Carry will depend on the result of the ADD.

We then test this with:

```
JP NC, &300F
```

This checks to see if the carry flag is set. If not – that is, if our answer didn't exceed the byte limit – the program jumps to memory location &300F, which, since it contains &C9, RET, simply ends the machine code routine.

Notice that for the first time we're jumping forward and will actually skip some code if Carry is not set.

If, however, Carry is set – that is, we have exceeded 255 in our answer – we carry on directly with the next bit of code after the JP, which is:

```
LD A,&07
CALL CharOut
```

which, I'm sure you'll recall, causes a bleep. We then encounter RET.

The outcome of all this is that our program will bleep for sums that cause the Carry flag to be set – the ones that gave us "wrong" answers – yet let the other sums pass without question.

All right, a warning bleep isn't all that useful, but it does illustrate that we can detect the Carry flag's signal and act on it. In practice we'd use this branching technique to do some more arithmetical work before we get the right answer.

Experiment by changing the values to be added – memory locations &3001, &3003 – and see whether you get the expected leaps or not.

Next replace our:

```
JP NC, &300F
```

with a

```
JP C, &300F
```

by changing the fourth line of Program III to:

```
3007    DA 0F 30  JP C, &300F
```

Notice how neatly these three bytes replace the old ones.

This time you jump when the Carry is set. This only happens when an addition gives a bigger than byte-sized answer, so there's no beep for "out of limits" answers.

If the answer stays within bounds, and Carry is cleared, you don't take the jump, and the beep is heard.

So by changing just one byte, the opcode for JP NC – &D2 – to the opcode for JP C – &DA – we have completely reversed the program's effect.

However the main point of the demonstration is that when we ADD two single bytes, if the answer exceeds 255 the Carry flag is set to warn us.

You may be wondering why we call it the Carry flag. Well if you can remember that far back, at school you used to mark the tens and units columns when you did sums – and those columns were very important!

If you were doing the sum:

```
 tu
 16
+ 9
 —
```

your thought processes went *9 add 6 is 15, which is too big for the units' column, so I'll put 5 units down and carry one ten to the tens' column.*

We work in much the same way in machine code. This time our "units' column" – a single byte – can hold up to the number 255, after which we have to carry, or add one, to a byte representing the second column of our sum.

Hence it's very useful that our flag is set to one automatically when we go past 255. Just when we need to carry one our flag is conveniently set to one, so we call it the Carry flag. We'll explain exactly how the whole process works later in the series, so don't worry if you don't follow it too well.

As we've mentioned, this flag is also set when the result of a calculation goes below zero. Program IV is a variant of Program III, adapted for subtraction.

```
address  hex code  mnemonics
 3000    3E ?1     LD A, ?1
 3002    D6 ?2     SUB ?2
 3004    32 F8 2F  LD (&2FF8),A
 3007    D2 0F 30  JP NC,&300F
 300A    3E 07     LD A, &07
 300C    CD 5A BB  CALL CharOut
 300F    C9        RET
```

Program IV

Once again, ?1 and ?2 are merely labels – you're meant to put your own numbers here at memory locations &3001, &3003.

The program works in much the same way as Program III. If there's No Carry generated, the JP NC simply jumps to the terminating RET. If there is a Carry, the bleeping code is performed.

As you'll discover when you experiment, the bleep occurs when the second number is larger than the first. That is, when you try to subtract a number bigger than the one already in the A register.

Think about it. If the number you're taking away is bigger than the number you're taking away from, you must cross that important zero boundary, cycling round the clock again and triggering the carry flag.

The following shows what happens as we subtract increasingly greater numbers from 2, say:

2 – 0 =	**2**	Carry clear
2 – 1 =	**1**	Carry clear
2 – 2 =	**0**	Carry clear
2 – 3 =	**255**	Carry set
2 – 4 =	**254**	Carry set

So the Carry flag is set when the second number is greater than the first. If you like, it's doing a sort of comparison, comparing the number in the A register with the one you're taking from it. If the latter is greater, the Carry flag is set.

You'd be amazed how often you actually want to compare bytes in machine code, but doing it as we did above – using SUB – isn't too useful. You see, although you can use the Carry flag with SUB in this manner to do a rudimentary comparison between a byte and what's in the A register, the contents of the A register will almost certainly be changed in the process. After all, you are taking a number away from it.

SUB n takes the number n from what is in the A register and stores the answer back in the A register. We represent this as:

```
A <--- A-n
```

However the Z80 designers have provided us with a comparison instruction that gets around this problem – CP n. This takes the number in the A register, subtracts n from it, sets the flags accordingly but does *not* put the answer back in A. It simply discards it, leaving the number in A unchanged.

So memory bytes and the registers we've met are unaffected by CP n – only the flags are set or cleared as appropriate. That is:

- If the number compared with A is greater than that in A, the Carry flag is set.
- If the number compared with A is less than or equal to A, the Carry flag is cleared.

Mathematically:

```
A  <  n  ------> Carry set
A  >= n  ------> Carry cleared
```

The effect on the flags is the same as for SUB, but the number in A is unaltered.

```
address  hex code  anemonics
 3000    3E ?1     LD A,?1
 3002    FE ?2     CP ?2
 3004    32 F8 2F  LD (&2FF8),A
 3007    D2 0F 30  JP NC,&300F
 300A    3E 07     LD A,&07
 300C    CD 5A BB  CALL CharOut
 300F    C9        RET
```

Program V

Program V is essentially Program IV with the SUB replaced by CP. Again, insert your own numbers into memory locations &3001,&3003 and see if you can predict the bleeps accurately.

Take a good look at &2FF8 after you've run it, though. This should prove to you that the contents of A have indeed been unchanged by CP.

To illustrate how we can use CP to effect, have a look at Program VI. We've met it before – in fact it's the first loop we met.

```
address  hex code  anemonics
 3000    3E 20     LD A,&20
 3002    CD 5A BB  CALL CharOut
 3005    C6 01     ADD A,1
 3007    D2 02 30  JP NC,&3002
 300A    C9        RET
```

Program VI

As you'll see, it prints out all the characters with codes from &20 to &FF by loading A with &20, printing the character with that Ascii code, increasing what's in A by one, jumping back with a JP NC to print A once again, then increase it, and so on . . .

Finally the character corresponding to &FF is printed and the value in A increased by one, taking it "round the clock" to zero, setting the Carry flag and dropping out of the loop.

What we've done is to take advantage of the fact that Carry is set when we go round the clock. Alternatively, we can use CP to check for the last character of the loop, and it has some advantages as well. Program VII shows the idea.

```
address  hex code  anemonics
 3000    3E 1F     LD A,&1F
 3002    C6 01     ADD A,1
 3004    CD 5A BB  CALL CharOut
 3007    FE FF     CP &FF
 3009    DA 02 30  JP C,&3002
 300C    C9        RET
```

Program VII

Notice that this time we're adding one to the A register *before* we print it out. We then compare it directly with &FF, since this is the last character we want printing. This means we load the A register with &1F initially, which is immediately increased to &20 by the ADD A,1.

Now until the A register gets to &FF the number we're comparing with A is greater than that in A, so the Carry flag is set, so we jump back to the top of the loop again with JP C.

Once the A register gets to &FF, the numbers we're comparing are equal and the Carry flag is cleared.

You might wonder why we didn't just add a CP 0 after the ADD A,1 of Program VI. Well, if you think about it, you can't go round the clock by taking zero away from a number, so Carry will never be set.

The nice thing about using CP in loops is that we can easily vary the number we end with. For instance, if we just wanted to print up to Z – Ascii &5A – we'd change the CP &FF to CP &5A. And, if we wanted to start with A – Ascii &41 – we load our accumulator initially by replacing LD A,&1F with LD A,&40. Remember, we're going to add one to it straight away.

Another advantage of using CP to set Carry rather than just going round the clock is that it allows us to do things such as print out the alphabet backwards, as in Program VIII.

```
Address  hex code  anemonics
 3000    3E 5A     LD A,&5A
 3002    CD 5A BB  CALL CharOut
 3005    D6 01     SUB 1
 3007    FE 41     CP &41
 3009    D2 02 30  JP NC,&3002
 300C    C9        RET
```

Program VIII

As a final example of the use of CP, let's use it to filter out a range of inputs. I'd better explain what I mean by that.

Often in a program you give people a choice of, say, five options from a menu. They respond by pressing a number in the range 1 to 5, corresponding to their choice. If they press a wrong key, it's ignored and the program waits until a valid choice is made.

So let's write a program that waits until a 1, 2, 3, 4 or 5 is input and then prints it out.

We'll use CharIn to input a key, and then check it's in range with two comparisons. Remember that CharIn puts the Ascii value of the character pressed in to the A register, so we'll be wanting to check for values in the range &31 to &35, the Ascii codes for 1 to 5.

In other words, the number in the A register must be greater than or equal to &31, our first comparison. It must also be less than &36, our second comparison.

Program IX puts this into practice:

```
address  hex code  anemonics
 3000    CD 18 BB  CALL CharIn
 3003    FE 31     CP &31
 3005    DA 00 30  JP C,&3000
 3008    FE 36     CP &36
 300A    D2 00 30  JP NC,&3000
 300D    CD 5A BB  CALL CharOut
 3010    C9        RET
```

Program IX

The CP &31 checks that the number in the A register is &31 or greater. If not, Carry is set and we jump back to the start to get another key since the character entered was "too low".

If we get past the check though, we than compare the number with &36. We need the number in the A register to be less than that, so this time we want Carry to be set. If it isn't, the key is "too high" so we jump back to get another key.

If, however, Carry is set, we simply continue. If the program has got this far the number in A must be in the range we want so we CALL CharOut and RET.

Once you've seen how it works, why not try to write it to accept only the upper case letters of the alphabet, &41 to &5A? It shouldn't be too hard.

Well that's all for now. There's lots more you can do with CP and loops in general, as we'll see next month.

Logically, you should know where to draw the line

Part VIII of the Amstrad graphics series by GEOFF TURNER and MICHAEL NOELS

YOU'VE probably already noticed that when you DRAW a line with a selected graphics pen, the line is always drawn in the ink colour that is filling that pen.

It seems fairly obvious that whatever the colours already on the screen, our line will always overwrite them. And whatever the background, the line is the colour of the ink filling the selected graphics pen.

Put the Amstrad into Mode 0 and try changing the background colour to any colour of your choice using CLG. Then draw a line across the screen in pen 3 like this:

```
CLG n
DRAW 639,399,3
```

Notice that whatever number (n) you use to clear the graphics screen, the line will always be drawn in red. This is as we'd expect, since we've specified pen 3 with the DRAW command and, until we do something about it, pen 3 is filled with bright red ink – ink number 6.

For the moment, only enter lines 10 to 140 from Program I plus line 300 which prevents the ready message reappearing after the drawing is complete. We can add the other lines later.

In this shortened form, Program I plots 16 strips of colour down the screen, then draws a single line across them with graphics pen 1. You'll see that the line is drawn in bright yellow – pen 1, filled with ink number 24 – across all 16 strips of colour. This is much as we'd expect from what we said earlier.

It appears, then, that the graphics pen completely disregards the colour of the background when it's used to draw lines. The same thing happens when using the graphics pen to plot individual points on the screen. The point plotted overwrites the background.

However, it doesn't have to be like this – it is possible to change the way that the graphics pen writes over the background colours.

The background can be allowed to interact with the graphics pen so that the line drawn isn't always the colour of the ink filling the pen used. In other words, the background can affect the colours that are drawn on it.

There are in fact four different ways that the graphics pen can interact with the background colours. The first we've already come across – it's where the background is ignored by the line being drawn.

For the other three ways in which the colours are affected, we need to study some rules of logic. This is because the way that the background affects the drawn line is determined by three logical operations, AND, OR and EOR.

So, before we see how pen and paper colours interact, we'll use some simple numbers to illustrate the rules of logic. If you find these hard to follow or your appetite is whetted for more, have a look at Mike Bibby's excellent Bits and Bytes articles in the April and May 1985 issues of *Computing with the Amstrad*.

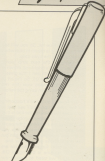
Before that, however, enter this command into your computer:

```
PRINT 2 AND 1
```

The micro should respond with the answer 0. If you're unfamiliar with

```
10 REM PROGRAM I
20 BORDER 0
30 MODE 0
40 FOR colour=0 TO 15
50 FOR strip=1 TO 40 STEP 4
60 MOVE 40*colour+strip,0
70 DRAW 40*colour+strip,399,colour
80 NEXT
90 NEXT
100 LOCATE 1,4
110 PRINT"NORMAL"
120 PRINT CHR$(23);CHR$(0)
130 MOVE 0,320
140 DRAW 639,320,1
150 LOCATE 1,9
160 PRINT"AND"
170 PRINT CHR$(23);CHR$(2)
180 MOVE 0,240
190 DRAW 639,240,1
200 LOCATE 1,14
210 PRINT"OR"
220 PRINT CHR$(23);CHR$(3)
230 MOVE 0,160
240 DRAW 639,160,1
250 LOCATE 1,19
260 PRINT"EOR"
270 PRINT CHR$(23);CHR$(1)
280 MOVE 0,80
290 DRAW 639,80,1
300 WHILE INKEY$="":WEND
```

Program I



logic functions, you may well have expected the answer to be 3, as you've probably always been taught that 2 and 1 equals 3.

The truth is that 2 PLUS 1 equals 3, but in our examples we have used the logical AND function instead of adding the two numbers together. It's just maths using a different rule. You already know the rule for adding – in the next paragraph you'll learn the rule for ANDing.

The AND function examines the binary values of the two numbers, and then compares each corresponding pair of bits. If both bits are set to the value 1 then the result of the AND function will also be 1.

You see, the AND function is saying "If the bit from the first binary number is 1 and the corresponding bit from the second binary number is 1 then the resulting bit will be 1. Otherwise the result will be 0".

If you can't see that, take it step by step applying the rule to each of the bits in turn. The number 2 in binary is 10, while the number 1 in binary is 01. If we consider each bit of the two numbers and apply our AND logic rule, then the result will be 00 in binary, which is 0 in decimal.

Compare the last bit or figure in the two numbers. The first is 0, the second 1 so ANDing them gives us 0. The second pair of bits are 1 and 0 so again we get 0 from the AND. We soon arrive at the answer. 10 AND 01 is 00.

To simplify things we use what is known as a truth table. This shows every possible combination of bits for a particular logical function. Figure I shows how the AND function affects each combination of two binary bits.

0 AND 0	0
0 AND 1	0
1 AND 0	0
1 AND 1	1

Figure I: AND logic truth table

Notice that in the above example we're only using two bits for simplicity, but the rules can be applied equally well to any number of bits. Usually bits are dealt with in bytes of eight at a time.

You might like to try your hand at ANDing some other numbers together. What is 7 ANDed with 9? Remember, change the numbers to binary first – 0111 and 1001 in this case. Then apply the AND rule to each successive pair of bits. Finally change the binary result (0001) back into decimal (1).

If you're unsure about converting decimal into binary, get your Amstrad to do all the work by using the following command:

```
PRINT BIN$ (n,8)
```

where n is the number in decimal – maximum 255 – and the 8 produces eight bits of binary or a byte.

One more example. Try finding the value of 12 AND 7. Now 12 in binary is 1100 while 7 is 0111. So ANDing them bit by bit gives 0100 which is 4 in decimal.

Now that we understand the use of the AND function when applied to numbers, we can investigate how colours are affected by it.

We can make the micro draw lines whose colour depends on the paper number of the background and the pen number of the graphics pen being used. These are ANDed together to give the number of the pen that actually draws the line.

To instruct the graphics pen to draw with AND logic, we need to use another of our control codes. Character number 23 is used, followed by a number in the range 0 to 3 – 0 is the default value which is used for normal drawing. Parameter number 2 is used to specify the AND function, so we need to enter:

```
PRINT CHR$(23);CHR$(2)
```

to see AND in action.

Add lines 150 to 190 to Program I. Now when you run the program, a second line will now be drawn across our coloured strips. This is because line 170 has told the micro to draw using AND to interact the background colour with the graphics pen.

Notice that although we have used pen 1 – bright yellow – the drawn line actually changes colour as it passes over each strip. Instead of the paper being ignored and overwritten as before, it's now affecting the resultant line.

The second line has been ANDed with the background colours. The rules for ANDing the paper and pen colours are exactly the same as for ANDing numbers.

In fact, all we need to do is find the numbers of the background and foreground pens and AND them together. When we know the result we have the number of the pen used to draw the line when the graphics pen crosses that particular background colour.

As an example of this, let's take a background colour of cyan as produced by pen 2. This is the third strip along from the left. We are going to draw over this in bright yellow with graphics pen 1.

If we AND the numbers 2 and 1, we end up with the result 0 and the resultant line is drawn in pen 0 which is filled with blue ink. So you will see our second line is drawn in blue when it passes over the cyan strip.

Looking at the next strip along, you'll see that it's bright red, drawn in pen 3. ANDed with pen 1 this gives 1 – 11 AND 01 gives 01. So the bit of the line that passes over the red background is drawn using the yellow ink of pen 1.

Try figuring out what's happening for the other strips. You'll see that in each case the colour of the line is that of the ink filling the pen number found by ANDing the numbers of the paper and graphics pen. As you can see, you might pick a yellow graphics pen but you don't always get a yellow line.

The second function that we are going to examine is the logical OR. Once again we need to convert our numbers to binary before applying the OR rule.

The OR rule says that "If either the bit from the first binary number is 1 or the corresponding bit of the second is 1 then the result will be 1. Otherwise the result will be zero".

Figure II is our truth table for the OR function.

0 OR 0	0
0 OR 1	1
1 OR 0	1
1 OR 1	1

Figure II: OR logic truth table

Notice that the result is 1 if one or both of the bits is 1.

Once again we can apply the OR function to the pen and paper colours on screen. If you now add lines 200 to 240 on to the end of Program I we will end up with a third line drawn this time using OR logic. The command for the OR function to be used is:

```
PRINT CHR$(23);CHR$(3)
```

You'll find this in line 220 which specifies that the paper number is to be ORed with the graphics pen to find the pen number that the line will be actually drawn in.

Notice that the third line also changes colour as it passes over the strips. This time the colour change sequence is completely different from the previous one when we used AND logic.

Looking at the fourth strip, we are ORing pen 1 – bright yellow – with paper 3 – bright red. This results in pen 3 being used to draw the line – 01 OR 11 gives 11. This leaves the line bright red, which, of course, we cannot see as it blends in with the background.

The next strip has pen 4 – white – ORed with pen 1 – yellow – resulting in pen 5 being used to draw a black line over the white strip. Try figuring out the logic of the rest of the colours.

The third and final function that we are going to look at is the EOR function, sometimes known as the Exclusive OR or even XOR.

The rule for EOR says that "If one bit and only one bit is set to 1 then the result will be 1 otherwise the result will be zero".

Once again a look at the truth table, shown in Figure III, will show us all the different combinations. Notice that its logic is subtly different from that of the OR operator.

To demonstrate the EOR function, we need to add the final lines 250 to 290 on to Program I. The:

```
PRINT CHR$(23);CHR$(1)
```

of line 270 has the micro EORing the background and foreground colours.

Now when we run the program, we have four lines drawn across the coloured strips, one for each logic function. Again, the fourth line changes colour as it passes over the strips, but as you see, the resulting colours are different from the AND and the OR lines.

Consider the fifth strip in paper 4 – bright white – which we are EORing with our chosen graphics pen, pen 1 – bright yellow. This time the result of the EORing is 5, so pen 5 is used giving us a black line on the white background. The sixth strip in paper 5 – black – EORed with pen 1 has that part of the line drawn in pen 4 – white.

And so we've covered the logical colours as they are known. Figure IV shows the parameters used to achieve them.

0 XOR 0	0
0 XOR 1	1
1 XOR 0	1
1 XOR 1	0

Figure III: XOR logic truth table

An important point to note is that when we apply any of the logic functions to our drawing, the resultant numbers refer to the pen numbers to be used and not to ink numbers.

Parameter	Function
0	Normal
1	EOR
2	AND
3	OR

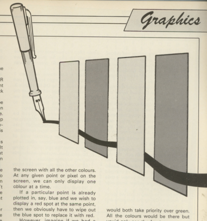
Figure IV: Logical colour selection

All the ANDing, ORing and EORing does is select a pen and use the ink colour that happens to be in filling pen at the time. This is demonstrated in Program II.

Here, two coloured strips are drawn using graphics pens 3 and 4 – lines 60 and 100. These colours will be bright red and bright white. At line 140 a line is drawn across the strips using graphics pen 1 which, as we

```
10 REM PROGRAM II
20 MODE 0
30 INK 3,6
40 FOR strip=192 TO 319 STEP 4
50 MOVE strip,100
60 DRAW strip,300,3
70 NEXT
80 FOR strip=320 TO 448 STEP 4
90 MOVE strip,100
100 DRAW strip,300,4
```

Program II



haven't done anything about it, is the default colour, bright yellow.

The line is drawn using the EOR function, which results in a bright cyan line over the red strip and a black line over the white strip.

Line 150 waits for a key to be pressed before the ink colour for pen 3 is changed to 26 – bright white. After the change of ink our red strip immediately turns white and following another key-press a second line is drawn across the strips.

Notice that the second line is identical in colour to the first one. It appears to have ignored the fact that the left-hand strip is now white when before it was red.

This is correct, however, as the logic function is still being applied to the pen numbers 3 and 4 resulting in a bright cyan/black line. It doesn't matter what the ink colours filling the pens are, it's the pen numbers that affect the resulting colour.

In the last case, it didn't matter that pen 3 has been filled with white ink, the EOR logic works in just the same way.

It is important to understand this point now, as later we will be changing ink colours quite often to demonstrate some useful techniques.

Combining colours logically as we have seen affects the colour of the lines drawn. Instead of just getting the colour filling the graphics pen, we get other colours depending on the logic used.

Having learnt all this, how can we put these effects to good use? We are going to look at involves producing multiplane images.

When we draw or plot anything in colour, we know that it has to share the screen with all the other colours. At any given point or pixel on the screen, we can only display one colour at a time.

If a particular point is already plotted in, say, blue and we wish to display a red spot at the same point, then we obviously have to wipe out the blue spot to replace it with red.

However, imagine if we had a number of different screens laid on top of one another, each one to be used for its own particular colour.

We could, for example, plot a red point on the front screen, and a yellow point on the second screen. Of course we wouldn't see the yellow spot as it would be obscured by the red one. If, however, we now removed the red spot, then the yellow one would now become visible through the clear front screen.

In this example the red screen has been placed at the front and so has priority over the yellow screen. We could have another screen representing green placed behind the yellow screen. In this case, yellow and red would both take priority over green. All the colours would be there but you'd only see the front one.

If you have ever played arcade-type games on your micro, you will probably have seen this effect when applied to animation. Objects moving around the screen will appear to go behind or in front of other objects depending upon the priority given to each colour.

Let's see how this works in practice.

Program III once again draws two coloured strips, this time in red and green. Lots of random bright yellow lines are drawn across the screen. Notice how the lines appear to pass in front of the green strip but behind the red strip.

In this example we have given

```
110 NEXT
120 PRINT CHR$(23);CHR$(1)
130 MOVE 0,250
140 DRAW 639,250,1
150 WHILE INKEY$="":WEND
160 INK 3,26
170 WHILE INKEY$="":WEND
180 MOVE 0,150
190 DRAW 639,150,1
```

```
10 REM PROGRAM III
20 MODE 0
30 PAPER 0
40 CLS
50 FOR strip=192 TO 319 STEP 4
60 MOVE strip,100
70 DRAW strip,300,12
80 NEXT
90 FOR strip=320 TO 448 STEP 4
100 MOVE strip,100
110 DRAW strip,300,3
120 NEXT
130 PRINT CHR$(23);CHR$(3)
140 INK 13,24
150 FOR lines=1 TO 20
160 DRAW RND(1)*640,RND(1)*400,1
170 NEXT
180 INK 7,26
190 FOR lines=1 TO 20
200 DRAW RND(1)*640,RND(1)*400,4
210 NEXT
```

Program III

priority to red over yellow and yellow over green. To achieve this it was necessary to change one ink colour at line 140 in order to produce the desired effect. Let's examine how Program III works.

First of all the background is drawn in the default pen 0 – blue. As we are using the OR function – line 130 – then the background colour can be effectively forgotten about as any number ORed with 0 stays the same. We have then drawn one strip in pen 12 – bright green, ink 18 – and the other one in pen 3 – bright red, ink 6. The lines are drawn using pen 1 – bright yellow, ink 24.

Now when pen 1 passes over something drawn with pen 12, the result of the OR function is 13. Pen 13 would normally produce pastel green, but in line 140 we've changed this to ink 24 – bright yellow – thus resulting in bright yellow lines over the bright green strip.

The point to grasp is that we found the result of ORing the two pens and then filled this pen with the colour we wanted to see. In this case we replaced the pastel green ink with yellow.

When a line drawn with pen 1 passes over the bright red strip – pen 3 – the result of the OR function is 3. This results in bright red lines over the bright red strip, which of course we cannot see. These lines appear to pass behind the bright red strip.

The same results could be achieved several ways by choosing an appropriate combination of inks linked to one of the logic functions.

Probably the easiest way of achieving this multiplane effect is by deciding what the end result is going to be and then working backwards selecting suitable colours.

As an exercise you may like to try changing the colour priority in Program III so that the yellow lines pass in front of red and behind green.

So that you don't have to calculate the values of every possible AND, OR and EOR combination, we can use Program IV to print out the result of every calculation.

You will find the tables produced by Program IV invaluable when selecting ink combinations for a particular application.

Copies of the tables are reproduced in Figure V.

If you're fortunate enough to have a printer connected to your Amstrad, you may like to amend Program IV to print out a copy of the tables. You'll need to change every PRINT statement to redirect the output to stream number 8 which is the printer stream.

As an extension to Program III try adding the following lines:

```
180 INK 7,26
190 FOR lines= 1 TO 20
200 DRAW RND(1)*640,RND(1)*400,4
210 NEXT
```

This results in bright white lines which pass in front of red and behind green. We had considered that red was the foreground colour followed by yellow and then green. Now we have white lines which appear in front of red but behind green. A genuine optical illusion. . . or is it just a bit of computer trickery? Incidentally have you observed what happens when white lines collide with yellow lines? Black holes, maybe?

And that's it for this month. Play around with the logical functions until you feel at home with them. They're one of those things that seem complicated in theory but soon become easy with practice.

Always remember that the AND, OR, and EOR refer to the pen/paper numbers, not the ink numbers.

And when you've grasped all that, you'll be ready for next time, when we find out more about logical colours.

```
10 REM PROGRAM IV
20 MODE 1
30 choice=0
40 PRINT"SELECT LOGIC FUNCTION"
50 PRINT
60 PRINT"1.   EOR"
70 PRINT"2.   AND"
80 PRINT"3.   OR"
90 PRINT
100 WHILE choice<1 OR choice>3
110 INPUT"Choose 1, 2 or 3 ";choice
120 WEND
130 IF choice=1 THEN logic$="EOR"
140 IF choice=2 THEN logic$="AND"
150 IF choice=3 THEN logic$="OR"
160 MODE 2
170 PRINT logic$;
180 FOR background=0 TO 15
190 PRINT TAB(4*background+10);backgr
ound;
200 NEXT
210 PRINT:PRINT
220 FOR foreground=0 TO 15
230 PRINT TAB(0);foreground;
240 FOR background=0 TO 15
250 IF choice=1 THEN result=foregroun
d XOR background
260 IF choice=2 THEN result=foregroun
d AND background
270 IF choice=3 THEN result=foregroun
d OR background
280 PRINT TAB(4*background+10);result
;
290 NEXT
300 NEXT
310 WHILE INKEY$="":WEND
320 GOTO 20
```

Program IV

EOR

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AND

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
4	0	0	0	0	4	4	4	4	0	0	0	0	4	4	4	4
5	0	1	0	1	4	5	4	5	0	1	0	1	4	5	4	5
6	0	0	2	2	4	4	6	6	0	0	2	2	4	4	6	6
7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
8	0	0	0	0	0	0	0	0	8	8	8	8	8	8	8	8
9	0	1	0	1	0	1	0	1	8	9	8	9	8	9	8	9
10	0	0	2	2	0	0	2	2	8	8	10	10	8	8	10	10
11	0	1	2	3	0	1	2	3	8	9	10	11	8	9	10	11
12	0	0	0	0	4	4	4	4	8	8	8	8	12	12	12	12
13	0	1	0	1	4	5	4	5	8	9	8	9	12	13	12	13
14	0	0	2	2	4	4	6	6	8	8	10	10	12	12	14	14
15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

OR

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	3	3	5	5	7	7	9	9	11	11	13	13	15	15
2	2	3	2	3	6	7	6	7	10	11	10	11	14	15	14	15
3	3	3	3	3	7	7	7	7	11	11	11	11	15	15	15	15
4	4	5	6	7	4	5	6	7	12	13	14	15	12	13	14	15
5	5	5	7	7	5	5	7	7	13	13	15	15	13	13	15	15
6	6	7	6	7	6	7	6	7	14	15	14	15	14	15	14	15
7	7	7	7	7	7	7	7	7	15	15	15	15	15	15	15	15
8	8	9	10	11	12	13	14	15	8	9	10	11	12	13	14	15
9	9	9	11	11	13	13	15	15	9	9	11	11	13	13	15	15
10	10	11	10	11	14	15	14	15	10	11	10	11	14	15	14	15
11	11	11	11	11	15	15	15	15	11	11	11	11	15	15	15	15
12	12	13	14	15	12	13	14	15	12	13	14	15	12	13	14	15
13	13	13	15	15	13	13	15	15	13	13	15	15	13	13	15	15
14	14	15	14	15	14	15	14	15	14	15	14	15	14	15	14	15
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15

Figure V: Colour logic

Basic just doesn't run true to type...

AFTER experimenting with the RSX commands in the May issue of your magazine – which is great by the way – I put the commands in REM statements so I could remember their syntax.

A very peculiar thing happened when I listed the program – the first letter of every character was missing!

In fact I had to type in the first letter twice to get the name to list properly. Why is this? – **E. Galore, Liverpool.**

■ We haven't a clue why this happens. It seems to be a bug in Basic 1.0 which has been cured in Basic 1.1 on the CPC664.

Help me find this

IF anyone at your magazine knows where I can buy a copy of Sterling Software's Country Cottages I would be very grateful.

Also if any reader knows of the whereabouts of a copy I would be very interested in hearing from them. – **Martin Watkins, 4 Finsbury Park Road, London N4 2JZ.**

■ The only information we have on Sterling Software is a PO Box 839, 86-88 Edgeware Road, London.

Stray arrows

I HAVE at last taken the plunge and attempted to write an arcade game that involves the PRINTing a small spider on the graphics screen using TAG, and moving it around using four direction keys.

I've achieved the latter but I can't seem to be able to get rid of a couple of arrows that have become permanently attached to my spider's rear end.

I've tried putting in a couple of spaces to remove them but they just attach themselves to the end of the spaces.

Are they something special that need to be dealt with in a special way, or have I missed something that is extremely simple? – **Darren Roberts, Newcastle upon Tyne.**

■ Your problem is caused by the fact that when you PRINT a character using TAG, the Amstrad follows it with the symbols for carriage return and line feed.

The solution is simple. The additional characters can be supressed by PRINTing a semi-colon ; immediately after the character as follows:

```
10 TAG
20 PRINT CHR$(X);
30 TAGOFF
```

For a fuller explanation of the problem and its solution see graphics article in our August issue.

Simple method

TO erase a file called GAME.BAS on disc using my CPC464 you have to enter:

```
A$="GAME.BAS"
¦ERA,@A$
```

This seems a bit longwinded. To rename something is even worse. You have to enter:

```
O$="GAME.BAS"
N$="ALIEN.BAS"
¦REN,@N$,@O$
```

This takes ages if there are several files to rename or delete. I thought I could enter:

```
¦ERA,"GAME.BAS"
```

or

```
¦REN,"GAME.BAS","ALIEN.BAS"
```

but it says "Type mismatch". Why is this? Surely there's a better way of doing it? – **D. Smith, Leeds.**

■ AmsDOS commands on the CPC464 do tend to be a bit longwinded. This is because they are RSXs and not ordinary Basic commands.

An RSX can only be given the address of a string, it can't be passed directly.

This problem has been overcome on the CPC664 and the second method will actually work.

Try using CP/M – it's even easier. You can simply enter:

```
ERA NAME.BAS
```

Making it OK for sound

I TYPED in a couple of programs from your magazine into my CPC664. I've gone through them line by line correcting all my typing errors and the programs seem to work OK except for the sound, which is very quiet.

I've checked the sound commands over and over again but they are exactly as they appear in the listing. Are the sound effects quiet in your programs or is my Amstrad faulty? – **J. O'Toole, Chester.**

■ Don't worry, it's not your Amstrad. The volume parameter on the CPC464 has a range of 0-7 if you're not using a volume envelope, but on the CPC664 it's 0-15. Simply double the volume parameter in any sound commands when entering a listing.

One over the seven

IN your correspondence columns I notice that other people are having the same trouble as I have been with an 8 bit printer.

My machine is a Tandy DMP-110, which must be similar to that of Mr Steel (see Computing with the Amstrad, June 1985), and you asked for assistance in this matter.

I also am a relative novice, my problem being getting the £ sign to print, and to produce graphics.

Fortunately I noticed the advert on page 36 by K.D.S. Electronics, who answered my query very promptly and I received their unit three days after placing an order.

It does everything that is claimed, that is prints the 8th bit, and removes the problem of doubling all the line feeds, but this is not the complete answer because if you are using commercially produced software the printer codes have been "tampered" with in order to make the 7 bit printers work.

Unfortunately Mini Office and Easy Amsword are both like this, and there is no way as far as I can see – unless you are a hacker – of dealing with this.

The program Advanced Amsword does have the facility to adjust the codes to suit us awkward blighters, but to my horror I find that if you adjust the numbers on the numeric keypad you automatically alter the codes for the numbers on the typewriter keyboard.

This means that there has to be a very careful choice of the keys that you wish to exchange for the upper Ascii set, as once you have committed yourself there is no way of changing it once you have started on your opus.

It would be rather nice if the software houses would remember that some of us would like our 8 bits unadulterated, and build this option in without losing the chance of using the numeric keypad independently if the loss of the upper set is due to the heavy use of the Ctrl key, as it is with the Advanced Amsword.

Obviously the best answer is to make your own program, and although I have not

Computing with the AMSTRAD Postbag

WE welcome letters from readers – about your experiences using the CPC464, 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:

Postbag Editor
Computing with the Amstrad
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY

studied your Text Editor (February issue) very deeply, it gives me the impression that it has possibilities.

Your magazine is the first that I have had no problems with in my attempts to come to grips with computering, I can even manage a bit of machine code, and AI has saved my sanity several times already. – **R.T. Taylor, Warley, West Midlands.**

■ Text Editor can easily be altered to your own requirements, as it is well notated and structured.

Why this line?

I HAVE just received the back issues of Computing with the Amstrad. It is without doubt the best on the market and leaves the "official" magazine in the shade.

Belated congratulations to Roland Waddilove for his Text Editor program. It not only works well but is so well written that modification was relatively easy.

I have now modified it so that I can vary the screen/paper width and placed an extra window between windows 0 and 2 for the display of screen width, margin and TAB markers in the edit mode.

All I need now is a printer (perhaps there is a spare knocking around the office).

A query relating to this program. Why is it necessary to put in line 1820 before a LOAD instruction:

```
1820 OPENOUT name$:CLOSEOUT
1830 LOAD name$,here
```

– **M.B. Smith, Tondu, Mid Glamorgan.**

■ This ensures that a buffer is set up to allow the text file to be saved or loaded. It's not the best way, but it does work!

Key to a riddle

WHAT is the Tab key for? The manual doesn't seem to admit of its very existence, and apart from printing a pretty horizontal arrow I have never been able to make it do anything.

Is it a hang over from the old standard typewriter keyboard? – **David F. Helps, Chipping Sodbury, Bristol.**

■ The Tab key forces the print position to the next tab position on the printer. Try:

```
→ PRINT #8," hello"
```

pressing the Tab key where you see the right hand arrow.

Note on strings

WITH reference to Ready Reference: String handling in the April issue of Computing with the Amstrad *it's worth noting in regard to the INSTR function that Basic considers the null or empty string "" as a substring of all other strings.*

Thus:

```
PRINT INSTR("wxyz","")
```

Returns 1.

This would be important if INSTR were used as in the following example, which could form part of a program as a means of making a selection from five numbered items in a menu:

```
10 PRINT "Enter number of s
elected item"
20 A$=INKEY$
30 IF A$="" THEN 20
40 IF INSTR("1234",A$)=0 TH
EN 20
50 A = VAL(A$)
60 ON A GOSUB 100,200,300,4
00,500
```

Line 30 is necessary to prevent A$ having the value "" at line 40. If line 30 were omitted, the INSTR statement in line 40 would immediately return the value 1, and the program would not wait for entry from the keyboard. – **Colin B. Cotterill, Middlesbrough, Cleveland.**

A tidy improvement . . .

HAVING just taken out a subscription to your magazine and received the tape of the June programs, may I make a few comments?

Da Bells is terrific. I can't get my five-year-old son away from it.

Personally, I'm dying to find a use for Easydraw. I have improved Ammon slightly by making the Dump option look more like a normal front panel display.

The changes are:

```
500 .........HEX$(base,4),
550 .........HEX$(byte,2);
```

– **R.W. Ford, Camberley, Surrey.**

■ We agree Da Bells is a superb game.

Easydraw was designed for people to make graphic introductions to their programs, or just to have fun drawing pictures.

Thanks for the ammendment to Ammon, it does tidy the screen up.

Useful for code

I HAVE written a program which may be useful for machine code writers.

It takes the raw machine code out of the computer's memory and converts it into a machine code "data" listing on tape.

```
10 REM *memory to data stat
eements
20 MODE 2
30 INPUT"Start of Code";srt
40 INPUT"End of Code";ed
50 MODE 2
60 PRINT "Start address ";s
rt
70 PRINT "End address ";ed
80 INPUT"OK";an$
90 IF UPPER$(an$)="N" THEN
20
100 IF UPPER$(an$)<>"Y" THE
N 50
110 SPEED WRITE 1
120 OPENOUT "DATA*"
130 PRINT #9,"1 DATA ";
140 lineno=1
150 FOR add=srt TO ed
160 lineno=lineno+1
170 IF lineno/10<>lineno\10
 THEN PRINT #9,HEX$(PEEK(ad
d))+",";
180 IF lineno/10=lineno\10
THEN PRINT #9,HEX$(PEEK(add
));CHR$(13):PRINT #9,lineno
;" DATA ";
190 NEXT add
200 PRINT #9,"9999";CHR$(13
)
210 CLOSEOUT
220 MODE 2
```

This is useful for using machine code in your own Basic programs, and also when submitting programs which incorporate machine code to magazines.

The program is used as follows:

- *Load/type machine code into memory.*
- *Run my program.*
- *Type in the start and end address of the machine code. The program will then save a text file under the name DATA*.*
- *The text file can then be*

loaded as a normal Basic program.

● *After loading renumber the data statements for your uses and add a reading program such as:*

```
10 FOR ADD=40000 TO 41000
20 READ dat$
30 POKE ADD,VAL("&"+dat$)
40 NEXT add
```

The VAL("&"+dat$) evaluates the string data as hex digits. The data statements are finished with 9999. – **Anthony Day, Camberley, Surrey.**

Hidden snag

I'VE found a couple of calls on the Amstrad which might interest some readers.

First type in Mode 2, then enter CALL –17909.

Lo and behold, you have four colours in Mode 2.

CALL –17910 will set the screen back to normal.

CALL –17911 will give a similar four-colour effect but with one colour flashing.

The computer will run properly with any of these calls in effect, although reading text can be a little difficult.

Maybe a programmer with some experience could use them to write a four-colour hi-res game. – **Shane Matthews, Brackwell, Berks.**

■ These calls do not give a true four-colour Mode 2. It actually switches back to Mode 1 but Basic still thinks it's in Mode 2. You won't be able to print or draw anything because of this confusion.

Garbage collection

DURING my last stay in Britain I bought your excellent magazine – Computing with the Amstrad, *June 1985.*

I was particularly interested – and helped – by the readers' letters and your answers, especially the one on page 70 pertaining to the garbage problem – "Clogged up by garbage".

I am indeed confronted with quite a lot of that time-consuming rubbish, in particular with Easi-Amscalc, a utilities software from Amsoft.

I tried your suggestion after RENUMbering the program's lines, of course, and indeed SAVEing and LOADing no longer suffer from any garbage collection whatsoever.

However, during operations, it seems thet the garbage clogging up – which was already quite a bother before – has got worse. It happens more often, although the duration of each hold-up seems slightly shorter.

My question is: Isn't there a way to correct this, just like for the SAVE and LOAD procedure?

After all, there are far more complex softwares for the Amstrad which don't seem to be subjected at all to garbage problems.

Second question: Is there a way to have "foreign characters" produced on the monitor's screen?

I know the procedure to re-program the keys – KEY DEF – but this only works on the printer.

Now foreign characters must exist somewhere in the insides of the Amstrad, since they can be produced on screen through the means of the Amsword Word Processor – but how does one achieve that feat in direct mode? – **Harry Kümel, Antwerp, Belgium.**

■ You can prevent garbage building up using X=FRE(" ") in the main loop in your program.

Foreign characters can quite easily be designed and used in your programs using the character maker in the May issue of *Computing with the Amstrad.*

At top speed . . .

HAVING just bought a CPC464, I am very impressed with the machine and your magazine is very good for general information.

My wife typed in Missile Command, which, after a few typing errors, is working fine. Can you tell us how to speed up the sight movements and how to program it to use a joystick? – **Colin Day, Bedford.**

■ The program is running as fast as it can in Basic. To speed it up you'll need to use machine code. Have a look at our Machine Code Graphics series.

It's quite easy to convert Missile Command to joystick. Simply change the inkey numbers in line 1170 and 160. Replace these two lines with:

```
160 IF f<0 THEN GOSUB 1210
ELSE IF INKEY(76)>-1 THEN a
=x%-16:b=y%+10:SOUND 129,10
,40,15,1,1,1:SOUND 1,1000,1
00,12,1,1,10:GOSUB 1210
1170 SOUND 130,2000,100,2:x
x%=x%+16*(INKEY(74)>-1)-16*
(INKEY(75)>-1):yy%=y%+8*(IN
KEY(73)>-1)-8*(INKEY(72)>-1
):IF (xx%<10 OR xx%>620) OR
 (yy%<100 OR yy%>390) THEN
RETURN
```

Read errors

WHAT exactly are Read error a and Read error b?

I know it is misreading of data off the tape, but what is the difference?

Is it possible to load a program from another cassette recorder into the CPC464 via the expansion port or printer port or whatever? Or would I need to make a hardware modification?

Please answer as I wish to use the Azimuth 2000 to align my hi-fi. – **I. Bentley, Doncaster.**

■ The "Read error" indicates that an error of some kind occurred while reading from the tape.

Read error a means an impossibly long 1 or 0 has been measured when reading the binary data. This often indicates reading past the end of the record.

Read error b means a cyclic redundancy check error and indicates data was read from the tape incorrectly.

The expansion port at the back wasn't designed to be used as a cassette port, so it might be difficult to make a suitable interface. Try checking the azimuth on your tape record heads on your Amstrad cassette deck – this may help.

Long variables

RONALD Waddilove's routine to reset the internal clock works well, but I must take him to task over much of the rest of the article.

Word wrap-around can be much more easily achieved:

```
10 a$="(A Long String)":
LOCATE 20,10: PRINT USING
"&";a$: PRINT:PRINT:END
```

will print the string without wrap-around.

Suppressing leading spaces in numbers is also very easy. There is no trailing space:

```
10 INPUT "Number:     ",n:
n$=MID$(STR$(n),2): PRINT
"*****";n$"*****":END
```

The leading space is not a bug. The space is reserved for a possible minus sign – a common feature of most micros.

I cannot agree that Basic can "create" new variables – these "new variables" are created by the programmer mis-spelling an intended variable which the operating system has no option but to accept.

Long variable names are fine, but by keeping them short the possibility of typing error is reduced and less program memory is used. I seldom use more than two characters. – **J.M. Dunnett, Wellington, Somerset.**

■ The 'Long String' feature was dealt with in the April issue, and Al's Beat explained how to get round the leading space in the August issue, but it does no harm to refresh these subjects.

Your point is taken about long variable names, but we prefer to use them as they make a program more readable.

SUPERCHARGE

SUPERPOWER SIDEWAYS ROM CARD (Ref B101)

This unit opens up a whole new field of personal computing, previously only available to owners of the BBC Micro and other top of the range computers.

The SUPERPOWER Sideways Rom Card has the following features:-

- Matching case, with easily detachable cover.
- Fits snugly to rear of computer.
- Bus extension for fitting of Disk Interface etc.
- Houses up to 7 Roms. (Foreground, Background & Extension).
- Any mix of 8K or 16K Eproms.
- No additional power supply necessary.

SUPERPOWER ROM BASED SOFTWARE

CURRENTLY AVAILABLE:

Programmer's Toolbox; Disk User's Utilities; Mailing List & Club Membership; Assembler, Disassembler & Machine-code Monitor.

COMING SOON:

Word Processor; Database ... END of AUGUST.

Spreadsheet; Graphics/Statistics ... END of SEPTEMBER.

Rom-based Software has the following important advantages:-

1. Written in machine-code, it is very fast in operation.
2. Programs are instantly available from the keyboard.
3. The program code does not use RAM, thus permitting much larger files to reside in memory, reducing the number of disk accesses and saving time when manipulating files.
4. The program itself cannot become corrupted.

SUPERPOWER PROGRAMMER'S TOOLBOX for the CPC 464 (Ref B104)

ADDITIONAL BASIC

TURTLE: Logo-like turtle graphics. TUNE: Output music string. CIRCLE: Draw a circle. FILL: Fill an area surrounded by foreground colour. GRAPHICS PEN/GRAPHICS PAPER: Set graphics and background colours. COPYCHR: Read a screen character. ECHO ON/ECHO OFF: Printer output on and off. PAGE ON/PAGE OFF: Paged screen output on and off. CLEAR INPUT: – clear input buffer. (+ extra graphics commands.)

PROGRAMMER'S AIDS

EDITOR: Gives 2 additional windows for program editing. FIND (& REPLACE): Find a string and optionally replace. XREF: List references to particular line numbers (GOTO's, GOSUB's etc). COMPACT: Remove REM statements. FRAME: Synchronise screen update with frame fly-back. PRINTER: Epson-compatible or Amstrad printer for dump routines. FDUMP: Fast two-tone screen dump, with user-definition of 'pen' and 'paper.' CDUMP: Shaded screen dump depicting up to 9 colours. PLOAD: Load program saved under 'P' option. INFO: Give details on specified disk file. MEDIT: Comprehensive HEX and ASCII memory editor. LTRON (LTROFF): Send TRACE output to printer. HELP: List commands, functions and their parameters.

SUPERPOWER DISK USER'S UTILITIES (Ref B103)

Program allows detailed inspection and modification of information held on disk and is of particular use for recovering data from corrupted disks. Individual sectors can be read from and written to. All data can be output to the screen and/or printer. Program also contains a number of functions of use to assembly language programmers.

TABULATE DIRECTORY: Displays directory and enters EDIT mode. READ DISK SECTOR: Read sector and enter EDIT mode. LOAD DISK FILE: Load first sector into buffer and remainder to memory for fast access. Enter EDIT mode. LOAD UPPER ROM: Catalogues resident roms, prompts for rom selection and enters EDIT mode. EDIT MEMORY: Displays current buffer. Data displayed is Buffer Address, Hex and ASCII. Comprehensive editing facilities. 'COPY' key gives Intelligent Data Copy. 'TAB' key gives printer output. WRITE: Write sector to disk. DISASSEMBLE: Disassembles code from specified address, giving address, object code, mnemonics and ASCII. Screen and/or Printer output. SEARCH MEMORY: Search sector or a complete file for ASCII string or series of Hex codes. UTILITIES: Includes Format, Hex/Decimal conversions, jump calculations etc. SCREEN UTILITIES: Select from four display modes; choose background and foreground colours.

SUPERPOWER SIDEWAYS ROM CARD (Ref B101)

SUPERPOWER MAILING LIST AND CLUB MEMBERSHIP PROGRAM (Ref B102)

Program handles thousands of name and address records (label and non-label fields). Twenty classification indicators make possible sophisticated selective examination, counting and printing of records. Alphabetical order is dealt with on data entry, allowing user to select 'key' word. Works with single and double disks as well as cassette.

Main Commands ... ENTER: Data entry(*) GET: Load new file. COUNT: Selective Count. FIND: Find 'Name' or any String(*). LIST: List current file on screen (*). PRINT: Print label data or whole records selectively. MERGE: Merge and Sort files. SAVE: Write a file to disk or cassette. RESET: Reset colours, label sizes, class definitions, string constants etc. (*Editing facilities available.)

FOR ONLY £39.95

SUPERPOWER PROGRAMMER'S TOOLBOX (Ref B104)	£39.95
SUPERPOWER DISK USER'S UTILITIES (Ref B103)	£39.95
SUPERPOWER MAILING LIST AND CLUB MEMBERSHIP PROGRAM (Ref B102)	£39.95
SUPERPOWER ASSEMBLER, DISASSEMBLER & MACHINE CODE MONITOR (Ref B105)	£39.95

SUPERPOWER ASSEMBLER, DISASSEMBLER & MACHINE-CODE MONITOR (Ref B105)

This suite of routines represents the complete Development Package for the Amstrad Z8Ø programmer. The assembler has a sophisticated text editor, a comprehensive set of options, is very fast and incorporates special techniques enabling large source files to be handled in memory. The full-feature disassembler produces files which can be edited and then re-assembled.

The Machine-code Monitor routines are extremely powerful, including the setting of conditional breakpoints (including loop counter option), single-step execution, alternative five column binary and mnemonic formats. Both sets of Z8Ø registers can be displayed. Other options include Intelligent Move, Modification of code to run at a new address and colour selection of border, paper and pen.

ROM-BASED SOFTWARE FULFILS THE PR

YOUR AMSTRAD.

SUPERPOWER ADVICE CENTRES

The following computer specialists stock the SuperPower Rom-card and Rom-based software for the Amstrad and are able to offer you advice and information on current products and the ever-expanding range:-

AVON
K & K Computers, 32 Alfred Street, Weston Super Mare. 0934 419324.

BEDFORDSHIRE
Chilton Computers, 76b Dallow Road, Luton. 0582 455684.
Hobbyte Limited, Unit 1G, Arndale Centre, Luton. 0582 457195.
Zedem Computers, 2 Kimbolton Road, Bedford. 0234 213645.

BERKSHIRE
Gadgets, 30 Peascod Street, Windsor. 07535 67211.

CHESHIRE
Northern Computers, Churchfield Road, Frodsham. 0928 35110.
Pyramid Micros, 25 Cairo Street, Warrington. 0925 35713

CLEVELAND
Multicoloured Micro Shop, 8 Dundas Street, Redcar. 0642 486643.

DEVON
Computerbase, 21 Market Avenue, City Centre, Plymouth. 0752 672128.
Computer Systems (Torbay), 35 Hyde Road, Paignton. 0803 524284.
Georges Booksellers, 147 Sidwell Street, Exeter. 0392 71072.
Open Channel, Central Station Buildirigs, Queen Street, Exeter. 0392 218187.
Computerware, 10 Lower North Street, Exeter. 0392 210507.

EIRE
Bit by Bit, 6 Castle Street, Bray, Co. Wicklow. 001 868 541

ESSEX
Clarke Coyston Computers, 332 London Road, Southend on Sea. 0702 337161.
Estuary Personal Computers, 318 Chartwell North, Southend on Sea. 0702 614131.
Computerama, 88-90 London Road, Southend on Sea. 0702 335443.
Colchester Computers, 3A Short Wyres Street, Colchester. 0206 47242.
Mirage Microcomputers, 24 Bank Street, Braintree. 0376 48321.

HAMPSHIRE
Electronequip, 36-38 West Street, Fareham. 0329 230671.
Alton Computers, 62 High Street, Alton. 0420 87998.
Timatic Systems, Fareham Market, Fareham. 0329 236727.

HUMBERSIDE
Computing World, 10 Swaby's Yard, Dyer Lane, Beverley. 0482 881831.
Ashby Computer Centre, 247 Ashby High Street, Scunthorpe. 0724 871756.

ISLE OF WIGHT
V M S Computer Centre, Bus Station Complex, Newport. 0983 525450.

KENT
The Village House of Computers, 87 Beckenham Lane, Bromley. 01 460 7122.
Microway Computers, 39 High Street, Medway Towns, Rainham. 0634 376702.
Marlowe Micros, 9 The Friars, Canterbury. 0227 453531.
Adventurer's Guild, 26 Harmer Street, Gravesend. 0474 334008.
Numbus Computing, 6 Station Road, West Wickham. 01 777 3160.
MD Micros, 4a Crayford Way, Crayford. 0322 524123.

LANCASHIRE
Computer World, 208 Chorley Old Road, Bolton. 0204 494304.
I M O Computers, 39-43 Standish Street, Burnley. 0282 54299.

LONDON
Adams World of Software, 779 High Road, North Finchley, London N12. 01 446 7170.
Colmore & Microtanic, 102 Lordship Lane, Dulwich, London SE22. 01 299 1419.
Vic Oddens, 6 London Bridge Walk, London SE1. 01 403 1988.
Computers Inc., 86 Golders Green Road, London NW11. 01 209 0401.
G & B Computers, 230 Tottenham Court Road, London W1. 01 580 3702.

LINCOLNSHIRE
M K D Computers, 24 Newlands, Lincoln. 0522 25907.

MANCHESTER
Mighty Micro, Sherwood Centre, 268 Wilmslow Road, Fallowfield. 061 224 8117.
Celcom Systems, 203 Corn Exchange, Manchester. 061 834 2808.

MERSEYSIDE
MicroByte, 71 Sea View, Wallasey. 051 630 6933.
Hargreaves Computers, 31-37 Warbeck, Liverpool. 051 525 1782.

MIDDLESEX
J K L Computers, 7 Windsor Street, Uxbridge. 0895 51815.
Screens Microcomputers, 6 Main Avenue, Moor Park, Northwood. 09274 20664.

NORFOLK
Abacus, 12 Pottergate, Norwich. 0603 614441.

NORTHERN IRELAND
Botanic Computer Centre, 5 Botanic Avenue, Belfast. 0232 240798.
Functional Computers, 28 High St, Lurgan, Co. Armagh. 07622 28543.

NOTTINGHAM
G A Computers, 1 Willford Road, Ruddington. 0602 211225.
Computer Market, 27 Goosegate, Nottingham. 0602 586454.

SCOTLAND
Micro Mart, 304 High Street, Kirkcaldy. 0592 206251.

STAFFORDSHIRE
Leek Computer Centre, 7 Ball Haye Street, Leek. 0538 387859.
Computer Cabin, 24 The Parade, Newcastle under Lyne. 0782 636911.

SURREY
Computerama, 17 Nork Way, Burgh Heath. 07373 54717.
Computersolve, 8 Central Parade, St Marks Hill, Surbiton. 01 390 5135.

SUSSEX
Gamer, 71 East Street, Brighton. 0273 698424.
Computermania, Southgate, Chichester.,0243 779263.
Computermania, 17 The Lanes, Littlehampton. 0903 716187.

TYNE AND WEAR
Sunderland Computer Centre, 29 Crowtree Road, Sunderland. 0783 655711/655736.

WALES
Gwynedd Computers, 10 Eastgate Street, Caernarfen, Gwynedd. 0286 2610.
Cardigan Electronics, Chancery Lane, Cardigan, Dyfed. 0239 614483.
Bucon Ltd, 35/36 Singleton St, Swansea, W Glamorgan. 0792 467980.
Teleview, 1B Alfred Street, Neath, Glamorgan. 0639 53282.

WARWICKSHIRE
O E M Computer Systems, 8-11 Regent Street, Rugby. 0788 70522.
Micro City, 1a Queens Road, Nuneaton. 0203 382049.

WILTSHIRE
Barbury Computers, 89 Victoria Road, Swindon. 0793 611487.
Trowbridge Computer Shack, 8 Timbrell St, Trowbridge. 02214 67299.

YORKSHIRE
Micro Power, Northwood House, North St, Leeds LS7. 0532 458800.
Danum Computer Systems, 17 East Laith Gate, Doncaster. 0302 20355.
The Computer Room, Unit 12, Market Hall, Barnsley. 0226 204000.
Just Micro, 22 Carver Street, Sheffield. 0742 752732.

DEALERS. Become a SuperPower Advice Centre and help us promote your expertise in home computing, with particular reference to the Amstrad market. Contact Eileen Garfield on 0532 434006 for details of our dealer support package.

HOW TO ORDER

Phone or write to: The SuperPower Project Manager, Micro Power Ltd., Northwood House, North Street, LEEDS LS7 2AA. TELEPHONE (0532) 434006. State your NAME and ADDRESS and the REFERENCE NUMBERS of the products you wish to purchase.

KEEP UP TO DATE

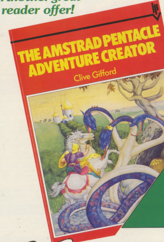
Phone (0532) 434006 to join our mailing list and receive regular bulletins on new products and release dates.

SUPER POWER

SERIOUS SOFTWARE FROM THE MICROPOWER ORGANISATION

PROSE OF YOUR AMSTRAD.

Another great reader offer!



This is the book that shows you how easy it is to create adventures on your Amstrad! Just type in the program, then follow the simple instructions to create as many exciting, hard-to-solve adventures as you like.

All you have to do is enter the data of your choice and the Pentacle program does ALL the hard work, building up an adventure you can save – and even sell!

As a bonus there's also a full set of data showing Pentacle in use with an adventure set in bygone days – Castle of Doom. Then a change of pace is provided with a full graphical adventure, The Search for Speck.

Now adventuring will never be the same again!

FREE!

The book that will really set you adventuring with your Amstrad will be sent to every subscription ordered on the form on the right.

Worth £2.99!

GIVE YOUR FINGERS A REST!

JANUARY: ***Smiley:*** *Guide Smiley round the labyrinth avoiding Grumpies.* **Code:** *Be a mastermind.* **Binary:** *Baffled by binary bits? Let our utility help you out.* **Dancer:** *Simple but fun. This is a lovely mover.* **Trapper:** *Pen the monsters of the maze.* **Scroller:** *A slick sideways scrolling routine.* **Letter Litter:** *Learn the keyboard.*

FEBRUARY: ***Digger:*** *Search for crystals and avoid the aliens.* **Simon:** *Our version of this classic game.* **Kingdom of Craal:** *A devious fantasy.* **Text Editor:** *A powerful word processor.* **Reaction Timer:** *Test out your reactions on this simple little program.* **Origins:** *Changing the graphics origin produces dramatic effects.*

MARCH: ***Starfleet:*** *Can you qualify as a starfleet pilot?* **Swamp:** *Join the battle against giant frogs.* **Parachute:** *Parachute to earth in this educational program.* **Dump:** *Screen dump utility for Modes 0 and 1.* **Hexer:** *A hexadecimal loader.* **PLUS . . . 3D Four In a Row.**

APRIL: ***Mad Adder:*** *Appease our aggravated adder.* **Egg Blitz:** *Eggs, boy scouts and a stunt pilot make up the fun.* **Simple Sprites:** *Create sprite-like effects.* **Caesar Cipher:** *Crack the code.* **Pilot:** *An easy to learn CAL language.* **Spelling:** *Learn to spell with this educational game.* **Aleatoire:** *The odd ways mariners make decisions.* **Palindrome Tester:** *Slick string handling techniques.*

MAY: ***Tronn Cycles:*** *Pit your wits in this two player arcade classic.* **Castle of Fear:** *A baffling fantasy adventure.* **Mouse:** *A simple but addictive game* **Cedric:** *Fun with shape recognition.* **Character Maker:** *Define your own characters.* **RSX:** *New commands for your micro.* **Cards:** *An apparently simple computer card trick.* **Trees:** *A clever routine to create 'trees'.*

Computing with the AMSTRAD

MAIL ORDER OFFERS

ORDER FORM

All prices include postage, packing and VAT
Overseas rate stated as Surface Mail. Air Mail price on application.

Valid to September 30
Please enter number required in box. £ p

Annual subscription

UK & Eire £12 (Sterling only)	6001	□	
Overseas (Surface) £20	6003	□	
Overseas (Airmail) £40	6004	□	
Please send me my free Amstrad Adventure Book	6052	□	
Commence with ________ issue		TOTAL	____

Tapes

£3.75 UK
£4.25 Overseas

January (Smiley)	6020	□	
February (Digger)	6021	□	
March (Starfleet)	6022	□	
April (Mad Adder)	6023	□	
May (Tronn Cycles)	6024	□	
June (Da Bells)	6025	□	
July (Missile Command)	6026	□	
August (Ah Bod!)	6027	□	
September (Robot Ron)	6028	□	
		TOTAL	____

3" Discs

£6.75 UK
£7.75 Overseas

January (Smiley)	6060	□	
February (Digger)	6061	□	
March (Starfleet)	6062	□	
April (Mad Adder)	6063	□	
May (Tronn Cycles)	6064	□	
June (Da Bells)	6065	□	
July (Missile Command)	6066	□	
August (Ah Bod!)	6067	□	
September (Robot Ron)	6068	□	
			____

Back issues

£1.25 UK
£1.50 Overseas (Surface)

January	6006	□	
February	6007	□	
March	6008	□	
April	6009	□	
May	6010	□	
June	6011	□	
July	6012	□	
		TOTAL	____

Binder

£3.95 UK
£5.00 Overseas

6038 □
TOTAL ____

Dust Cover

£3.95 UK
£4.45 Overseas

6037 □
TOTAL ____

Cassette tape and disc annual subscription

Tape	£40 UK	£46 Overseas	6005	□
3" Disc	£70 UK	£80 Overseas	6059	□

Commence with ________ (state month)

TOTAL ____

>>> TOTAL ____

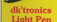
How to keep your collection complete

Bound in rich burgandy pvc and bearing the Computing with the Amstrad logo, this handsome binder will hold a year's supply of the magazines firmly secured in place with metal rods. **£3.95**

Dust Cover Only £3.95

Protect your keyboard with our luxury dust cover made of soft, pliable, clear and water-resistant vinyl, bound with strong cotton and decoeated with the magazine's logo.

NEW!



dk'tronics Light Pen

Full details on Page 37

All the program listings from each month's issue are available for only ***£3.75*** *on cassette and* ***£6.75*** *on disc.*

JUNE: **Easydraw:** *A versatile graphics utility to create and manipulate your own pictures.* **Da Bells:** *Hilarious arcade action for the whole family.* **AMmon:** *A superb machine code monitor.* **Menu:** *Create colourful menus.* **Nim:** *Baffle your friends with Alleatoire's simple routine.* **Quads:** *A clever routine to display expanding quadrilaterals.*

JULY: **Missile Command:** *Take command of your laser base and save earth from the invading hordes.* **Jumbo's Wordgame:** *Have hours of family fun with our Amstrad version of Odd One Out.* **Biorhythms:** *Find out if you're going to have "one of those days" with this superb utility.* **Raw:** *Add power to your programming with this superb machine code assembler.* **Compacter:** *Reduce the memory used while saving a screen with this handy utility. Load it back in half the time.*

AUGUST: **Ah Bod:** *Guide Bod to collect the eggs and avoid the evil Stompers.* **Dedit:** *A superb disc utility for sector editing and recovering lost programs.* **Polygons:** *Create and display superb shapes rotating about their horizontal and vertical axes.* **Events:** *Demonstrations of interrupt routines.* **Password:** *A simple routine to assist you to generate your own password.* **Graphics:** *A superb but easy to use character generator and demonstration programs.* **M/code graphics:** *Routines to examine the screen and pixel patterns.*

SEPTEMBER: **Robot Ron:** *Guide our hero in his quest to rid the screen of the evil weevils.* **Drogna:** *Can you survive on the planet Arg, with its ever-changing money system?* **Magnified Characters:** *A superb enhancement to EasyDraw, our powerful graphics utility. Create and manipulate your own colourful text screens.* **Smiley Hunt:** *It's simple but it's fun. Search for those happy smiling faces in the finished version of Al's mega-game.* **Graphics:** *All the listings from this month's graphics article.*

Red Arrows

Tape	£8.95 UK	£9.45 Overseas	6050	□
Disc	£12.95 UK	£13.95 Overseas	6051	□

TOTAL ____

dk'tronics Light Pen

£24.95
(UK & Overseas)

6057 □
TOTAL ____

dk'tronics speech synthesiser

£34.95
(UK & Overseas)

6040 □
TOTAL ____

Mini Office

Tape	£5.95 UK	£6.45 Overseas	6035	□
3" Disc	£9.95 UK	£10.45 Overseas	6036	□

TOTAL ____

Classic Games on the Amstrad

Tape	£5.95 UK	£6.45 Overseas	6048	□
Disc	£9.95 UK	£10.95 Overseas	6049	□

TOTAL ____

Payment: please indicate method (✓) **TOTAL** ____

□ Access/Mastercharge/Eurocard
Card No. ⊔⊔⊔⊔ ⊔⊔⊔⊔ ⊔⊔⊔⊔ ⊔⊔⊔⊔

□ Barclaycard/Visa
Card No. ⊔⊔⊔⊔ ⊔⊔⊔ ⊔⊔⊔ ⊔⊔⊔ Expiry date

□ Cheque/PO made payable to Database Publications Ltd.

Name ________ Signed ________
Address ________
____

Send to:
Computing with the Amstrad, **FREEPOST, Europa House,**
68 Chester Road, Hazel Grove, Stockport SK7 5NY.
(No stamp needed if posted in UK) *Please allow 28 days for delivery*

YOU CAN ALSO ORDER BY PHONE: 061-480 0171

Don't forget to quote your credit card number and full address CA/9

ADVERTISERS INDEX

Amstrad CPC464

Speech Synthesizer

The dk'tronics Amstrad speech synthesizer and powerful stereo amplifier uses the popular SLO/256 speech chip and has an almost infinite vocabulary. It is supplied with a text to speech converter for ease of speech output creation. Everything you wish to be spoken is entered in normal English, without special control codes or characters, it is therefore extremely easy to use. The voicing of the words is completely user transparent and the computer can carry on its normal running of a program while the speech chip is talking. The speech output from SLO/256 is mono and directed to both speakers.

Stereo Output

To utilise the Amstrad stereo output on the back of the computer, the interface has a built in stereo amplifier, this gives all sound output a totally new dimension and greatly improves the sound quality and volume over the computer's internal speaker. Any sound that previously came out of the mono speaker will now be sent out via the interface in stereo. All programs that use the sound in anyway (i.e. commercial software) will now output through the interface, which is fitted with volume and balance controls .

Speech Synthesis

The Amstrad speech synthesis utilises parts of the spoken word known as allophones. These are actual sounds that go to make up speech. The SP0256 allophone speech synthesis technique provides the ability to synthesize an almost unlimited vocabulary. Fifty-nine discrete speech sounds (allophones) and five pauses are stored in the speech chip's internal rom.

Text to Speech

Although there are only 26 letters in the alphabet, letters have a totally different sound when used in different words. For example, The "a" in 'Hay' is much longer and softer than in 'Hat'. When you speak you automatically make adjustments because you know just how a word should sound Not quite so easy with a computer.

The machine code software is mainly developed to this mode of operation. 3.5K is used for tables which contain the rules & exceptions to the rules of the English Language.

e.g. I before E except after C) This therefore allows the user to enter words to be spoken in normal English.

Speakers

Supplied with the Speech Synthesizer are two high quality 4" speakers these have been designed to compliment theAmstrad Computer. They are fitted with 1 metre of cable and can be positioned for the best stereo effect. The synthesizer interface fits neatly on to the rear of the computer. It has a through connector to enable other interfaces (e.g. Disc Drive) to connect to the rear of the synthesizer for ease of expansion Please send S.A.E. for a copy of the instruction manual which will give full and comprehensive details.

New Basic Commands

There are 8 new Basic Commands which control all the functions of the interface. Making the Synthesizer very easy to use. You can even control the speed at which it will talk to you. Or use the synthesizer to create sound effects like a fourth sound channel.

```
10 PRINT " 'AMSTRAD' "
```

The above is an example of the Syntax for entering speech into the computer and shows how simple it is to use.

The instruction book gives comprehensive details and examples of how to use the interface both from machine code and basic.

How to Order

The Amstrad Speech Synthesizer costs only £39.95. You can obtain your synthesizer through any good computer store or by completing the order form and returning it to:
dk'tronics Limited, Shire Hill, Saffron Walden, Essex. OR by telephone quoting your barclaycard or access number. Orders normally despatched within 24 hours.

Please rush me

.....[QTY] Amstrad Speech Synthesizer at £39.95 +£1.25 p&p
I enclose cheque/PO/Cash for Total £.......................
or debit my Access/Barclaycard No.
□□□□□□□□□□□□□□□□

Signature...
Name...
Address...

dk'tronics

Saffron Walden, Essex CB11 3AQ
Tel: (0799) 26350 10 lines