A Database Publication

Computing with the AMSTRAD

No. 10
October 1985
£1

The independent magazine for Amstrad computer users

They're here!

Full details of the new power-packed PCW8256 and CPC2168 – Page 25

Create your own full colour sprite editor

FREE CONTEST!

We're giving away 50 Hobbit games

LE MANS

Go Grand Prix racing – in 3D!

PONTOON

Take your Amstrad to the cleaners!

BEACH-HEAD

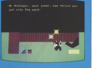
Fantastic offer on the game everyone's talking about!

Amstrad teach-in

Learn about STR$ and VAL, explore the Amstrad's SOUND command and create loops in machine code.

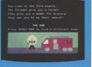
MIRRORSOFT

INTRODUCING A NEW GENERATION OF BED-TIME STORIES


Mr McGregor, your owner, has thrown you out into the yard.

Young Caesar, in a bout of over enthusiastic mousing, has broken the best crockery in the larder. Mr MacGregor, his owner is understandably displeased, and gives Caesar the grand order of the boot!

Children make simple choices on each screen, using the keycard provided, to decide where Caesar goes next in his search for a new home. With 38 different routes through the story, and 18 different endings, there's many an evening of entertainment in store. Plus, the pack includes a complete 64-page storybook which expands the storylines in the program and can be read with an adult or alone by older children.


You ride in the fire-engine.
The firemen give you a helmet.
They give you a medal for bravery!
They ask you to be their mascot!

THE END
Press SPACE BAR to find a different home

"If you have children under ten, they should be transfixed by this one, and I might add that I was, too." *— Popular Computing Weekly*

Available for:	Amstrad CPC464	£8.95
	BBC B (disk only)	£10.95
	Commodore 64	£7.95
	Spectrum 48K	£7.95

MIRRORSOFT

Maxwell House, 74 Worship Street, London EC2A 2EN Tel: 01-377 4644
Trade orders to: Purnell Book Centre, Paulton, Bristol BS18 5LQ

ESP THE PEN THAT LIKES TO SAY YES

ELECTRIC STUDIO PEN

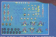
WHILST OUR COMPETITORS MOSTLY SAY NO!

LOOK!

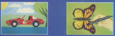
FEATURES/ FUNCTIONS	ESP	dk'tronics	Any Other
SINGLE COMPLETE ON SCREEN MENU	YES	NO	
DRAG SCREEN OBJECTS	YES	NO	
FLIP SCREEN OBJECTS	YES	NO	
CURSOR REMOVAL	YES	NO	
ELASTIC BOXES	YES	YES	
ELASTIC LINE	YES	YES	
ELASTIC TRIANGLE	YES	NO	
ELASTIC ELLIPSE	YES	NO	
ELASTIC DIAMOND	YES	NO	
ELASTIC POLYGON	YES	NO	Please compare any other pen package currently available
ELASTIC HEXAGON	YES	NO	
ELASTIC OCTAGON	YES	NO	
ELASTIC CUBE	YES	NO	
ELASTIC PYRAMID	YES	NO	
CIRCLES	YES	YES	
SOLID CIRCLES	YES	NO	
SOLID BOXES	YES	NO	
SOLID ELLIPSES	YES	NO	
WEDGES	YES	NO	
BEZIER SIMULATIONS	YES	NO	
ZOOM EDIT	YES	YES	
REVERSE/MIRROR IMAGES	YES	NO	
REFERENCE BACKGROUND	YES	NO	
GRID BACKGROUND	YES	NO	
X,Y DISPLAY OPTION	YES	NO	
PAINT FILL	YES	YES	
COLOUR WASHING	YES	NO	
RESIDENT SCREEN DUMP	YES	NO	
3D EDGE PLOTTING	YES	NO	
TEXT	YES	YES	
9 BRUSH SIZES	YES	NO	
18 SPRAY NOZZLES	YES	NO	
4 BASIC TEXTURES	YES	NO	
TEXTURE VARIATION	YES	NO	
XOR TEXTURE SHADING	YES	NO	
RESIDENT SYMBOL/SHAPE FILE	YES	NO	
RESIDENT FLECK PRINT FILE	YES	NO	
26 PAPER COLOURS	YES	NO	
15 COLOUR/TONE PALETTE	YES	NO	
POINT SETTING	YES	YES	
FIXED POINT RAYS	YES	NO	
MIRROR DRAWING	YES	NO	
HOME FUNCTION	YES	NO	
KEY CONTROL NUDGE	YES	YES	
JOYSTICK NUDGE	YES	NO	
AVAILABLE FOR 464	YES	YES	
AVAILABLE FOR 664	YES	?	
MODE 1 & 2 AVAILABLE	YES	?	

DUE TO LACK OF SPACE, WE ARE NOT ABLE TO LIST THE OTHER 20+ FUNCTIONS OUR PEN IS CAPABLE OF.

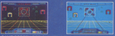
JUST SOME OF THE THINGS YOU CAN DO WITH THIS COMPLETE LIGHT PEN PACKAGE



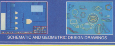
USER DEFINED CHARACTERS



FREEHAND DRAWING WITH THE GRAPHICS PROGRAM



GAME SCREEN DESIGN WITH COLOUR EDITING FACILITY



SCHEMATIC AND GEOMETRIC DESIGN DRAWINGS

AVAILABLE FOR: CPC464 ON TAPE £19.95

CPC464 ON DISC (incl. Interface) £26.95

CPC664 ON DISC £26.95

Also available: High Res Graphics Program for more serious applications (includes both Mode 1 and Mode 2)

TAPE £9.95 OR DISC £14.95

AVAILABLE FROM ANY GOOD COMPUTER STORE

If you have any difficulty obtaining our products, please send cheque/P.O. to:

THE ELECTRIC STUDIO
P.O. BOX 96, LUTON LU3 2JP Tel: (0582) 595222

Vol. 1 No. 10 October 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: 72:MAG001
Telex: 256871 MONREF G
Quoting Ref. 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.

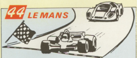
News trade distribution:
Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Tel: 0293 27053.

Enjo
atte
the

Delv
our
clev

Enj
All-
dk'i

Contents



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

Now you can teach your Amstrad to talk!

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!

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!

These are the sounds – and pauses – you can create on your Amstrad

A	AE	26	fat
A	AY	20	great
AI	AIR	47	hair
AR	AR	59	farm
AU	AO	24	aught
B	BB1	28	rib
B	BB2	63	big
C	KK1	42	common
'C	C	8	uncle
K	KK2	41	sky
CH	CH	50	church
D	DD1	21	could
D	DD2	33	do
E	EH	7	bend
E	EE	19	see
ER	ER	51	cater
F	FF	40	fire
G	GG2	61	go
IG	GG3	34	wig
GU	GG1	36	guest
GE	ZH	38	beige
H	HH1	27	he
H	HH2	57	hoe
I	IH	12	fitting
I	Y	6	sky
IR	ER2	52	bird
J	JH	10	jury
L	LL	45	luck
L	EL	62	angle
M	MM	16	milk
N	NN1	11	earn
N	NN2	56	no
NG	NG	44	bans
O	O	23	cot
O	OW	53	snow
O	UW1	22	do
OO	UW2	31	food
OR	OR	58	store
OU	OW	32	ouch
OY	OY	5	toy
P	PP	9	pub
R	RR1	14	real
R	RR2	39	brain
R	YR	60	fear
S	SS	55	sat
SH	SH	37	shirt
T	TT1	17	its
T	TT2	13	top
TH	TH	29	thin
TH	DH1	18	they
TH	DH2	54	bathe
U	AX	15	succeed
UH	OO	30	cook
U	YY1	49	compute
V	VV	35	even
W	WW	46	wool
WH	WH	48	whig
Y	YY2	25	yes
Z	ZZ	43	zoo
PA1	PA1	0	10 mS
PA2	PA2	1	30 mS
PA3	PA3	2	50 mS
PA4	PA4	3	100 mS
PA5	PA5	5	200 mS

Column 1: Sound. Column 2: Allophone name. Column 3: Allophone number. Column 4: Example word.

!

Save £5 (plus FREE post and packing)
Offer price only £34.95
Amstrad Speech Synthesiser
PLUS
STEREO
dk'tronics
Speaks for itself

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. ____________ Expiry date / /

□ Visa Card No. ____________

Signed ____________

Name ____________

Address ____________

____

POST TO: Speech Synthesiser Offer, Database Publications,
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Allow 28 days for delivery

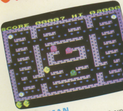
CA10

4 GREAT GAMES FOR THE PRICE OF ONE!

```
SCORE 00060 HIGH 05000 LIVES 3
```

ALIEN INTRUDERS

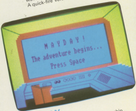
With only your laser for protection, destroy the waves of aliens who threaten to engulf you. A quick-fire version of an all-time great!



```
SCORE 00007 HI 02000
```

SNAPMAN

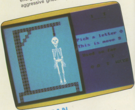
Steer your man around the maze, gobbling up energy pellets while keeping away from the aggressive ghosts who are out to get you.



```
MAYDAY!
The adventure begins...
Press Space
```

MAYDAY

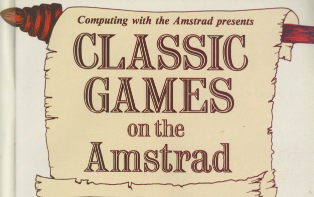
Guide the sole survivor of a stricken spaceship through the wreckage of his craft. Recover vital medical supplies . . . or a planet is doomed!



```
Pick a letter O
This is move 9
Used so far
```

DEADMAN

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



Computing with the Amstrad presents

CLASSIC GAMES on the Amstrad

Here's something really special from *Computing with the Amstrad!* We've commissioned four rip-roaring programs that no games collection is complete without – the kind of games that really stand out in the short history of microcomputing.

This value-for-money package includes two top-rate machine code arcade classics plus a traditional word game and a futuristic adventure.

There's hours of enjoyment and something to suit everyone in this superb collection.

Please send me Classic Games on the Amstrad Vol. 1 (✓)

Cassette £5.95	*6048*	□
3" Disc £9.95	*6049*	□

Payment: please indicate method (✓)

□ Access/Mastercard/Eurocard

No. └┴┴┴┘ └┴┴┴┘ └┴┴┴┘ └┴┴┴┘

□ Barclaycard/Visa

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. CA10

3D GRAPHICS
Vortex SOFTWARE
ANDROID TWO
AMSTRAD CPC 464
3D GRAPHICS
Vortex SOFTWARE
TORNADO LOW LEVEL
AMSTRAD CPC 464
NOW.
TWO VORTEX GIANTS
TURN ON AMSTRAD.

£100 off CPC 464 prices

AMSTRAD has slashed the prices of its two CPC464 models by £100 – and thrown in free software to boot.

Effective from September 2 the green screen version will retail for £199 while the 464 with colour monitor will carry a price tag of £299.

And both will be offered with 12 free software titles.

The reduction is seen by most observers as a result of the company introducing its more powerful CPC6128 at the former price level of the 464, but Amstrad itself doesn't go along with this.

According to chairman Alan Sugar: ''The reduced prices are in fact the original prices intended for the machine when it was introduced in April 1984.

''But because of the weakness of the pound and the effect that had on buying the components from abroad, we had to increase prices.

''Now we feel the 464 is being offered at its true current market value''.

What about Amstrad customers who had already purchased their CPC464s before the prices fell?

''Naturally they won't be too happy about this'', admitted Alan Sugar, ''but the situation was created by forces beyond our control.

''As a company, our reputation has been built on giving people value for money – and we are still doing just that''.

Production boost for Christmas

AMSTRAD has dramatically ramped up its computer production to almost 100,000 machines a month to cope with the anticipated demand in the run up to Christmas.

The company has instructed its South Korean manufacturing facilities to effectively double the levels achieved over the last 12 months.

Of the new monthly total some 40,000 will be made up of the newly launched PCW8256. The remainder will be almost equally divided between the CPC464 and the CPC6128.

Said Amstrad chairman Alan Sugar: "The new machines follow our policy of producing products which will sell in volume. We don't make machines just to get rave reviews".

To ensure their success Amstrad is embarking on a major promotional campaign involving extensive television and magazine advertising.

"This will be the first time we have really pushed our products", says Alan Sugar. "We are making sure we don't end up with too many machines after Christmas".

In all, Amstrad predicts it will sell at least 600,000 computers world wide next year, with its more advanced machines winning an ever increasing market share.

However sales of the CPC464 are likely to be cut back after Christmas as the company believes the market in which the machine finds itself will become even more seasonal.

The company believes that for the time being at least the UK will be its main market. It currently accounts for 45 per cent of all world wide sales.

Exit the CPC 664

AMSTRAD has killed off the CPC664. The company announced that it had ceased production of the machine when it launched its successor – the CPC6128 – on to the UK market.

"It suffered from a leap in technology", explained Amstrad chairman Alan Sugar.

What of unsold stock? "We don't have any sitting on shelves – well, none to talk about any way", said Alan Sugar. "The only 664s we have left are some that were kept back to fulfil our commitment to mail order catalogue firms".

Asked about dealers left with CPC664s on their hands, a company spokesman said: "We honestly believe there is very little out there. But what there is they shouldn't have much difficulty of getting rid of at reduced prices under the circumstances".

Software policy switch

AMSTRAD has done a complete U turn on its policy over the provision of software for its range of micros.

It has now agreed to assist third party companies to produce programs for its computers. Previously it insisted on them coming under the wing of its software division, Amsoft. And that meant paying a fee for the priviledge.

Malcolm Miller, Amstrad's marketing boss, now claims that the wrong interpretation had been placed on the company's previous stand on software.

"We may have been slightly selfish in the past", he told *Computing with the Amstrad,* "but that was because we wanted to get the show on the road with what we thought would be the best possible titles.

"Now we not only intend to support our own Amsoft projects but those of outsiders as well. I'm hoping ,the two situations can co-exist and provide the best possible support for our machines".

THE RELIABLE ONE

AMSTRAD now claim to have the most reliable computers.

In a recent survey by a chain of computer centres in Wigan and the North West of England, Amstrad came out with the lowest returns on faulty machines, with only 1.8 per cent returned. Percentage returns ranged from as much as 30.6 per cent on the Sinclair Spectrum, down to the lowest figure of 1.8 per cent for Amstrad. The figures were based on "dead on arrival" stock, and did not include machines returned after 30 days of purchase.

BIG DIXONS DEAL

DIXONS has won an exclusive deal with Amstrad to be the sole High Street chain to sell the PCW8256 up until the end of the year.

The retail giant's staff have already received training on the machine before it goes on sale in the group's 250 stores.

"We were so impressed with the product and the market positioning", said Eddie Styring, Dixons' managing director, "that we demanded exclusivity in the high Street multiple sector".

Independent dealers throughout the UK will still be able to get supplies from Europa Electronics, Amstrad's wholly owned subsidiary which services that sector of the industry.

Pressure reprieves MicroLive

BBC 2 Television's MicroLive series has been brought back as a result of pressure from computer buffs.

Starting on October 13 the now weekly series will cover subjects ranging from electronic music, micros in schools, the Data Protection Act and the use of computers in timing races.

The battle for the personal computer market is to be featured in a serious item with a lightweight title, Big Blue and the 700 Dwarves.

MicroLive will also take an in-depth look at work being done by the Carnegie-Mellom University in the States on molecular memories and robots, at communications and networking.

The program's producers have also taken heed of viewer's requests by introducing more items on personal computers. But air time will be given to the growing business and communications sectors.

Presenters will be Lesley Judd, Ian McNaught-Davies and Fred Harris. MicroLive will go out on BBC 2 at 7.30pm each Friday.

PCW for US market

AMSTRAD is planning to launch the new PCW8256 onto the American market, where it recently established a foothold with the CPC6128.

However the company will not be laying out any of its own money on the States-side invasion.

Instead it will rely solely on the American importer to fund the operation,

Amstrad decided to adopt this low profile approach to North American sales after witnessing the Acorn debacle.

Failure to penetrate the lucrative market cost the manufacturers of the BBC Micro some £6 million and almost brought the company to its knees earlier this year.

So Amstrad opted for passing on the financial risk to Indescomp, a Chicago-based distributor.

Amstrad's chairman and managing director Alan Sugar told *Computing with the Amstrad:* "All we'll invest in the US market is the price of an aeroplane ticket out there – and even that's only maybe".

Finance package for businessmen

BUSINESSMEN who do not have a detailed knowledge of accounting principles can manage their financial affairs as long as they have an Amstrad.

Latest on the scene is Quest International with its suite of Cash Trader, Sales Trader and Analyser programs.

The first two offer profit and loss accounts and balance sheet together with audit facilities, VAT returns and summaries.

Both have 140 user-definable analysis headings, 10 of which are reserved for bank and building society accounts and credit cards and a further 20 to record credit sales and purchases.

Payments, receipts, credit sales purchases and adjustments can all be handled.

Sales Trader, which can only be used with an Epson PX-8, has a free standing sales invoicing module. Priced £125, it can hold details of 300 customers and 300 products.

Analyser, a sort utility, is only available for Cash Trader. It enables data relating to any heading to be extracted for use in reports or summaries.

The Cash Trader with Analyser costs £125. Separately, they are £95 and £45 respectively.

Train with Capes

SOFTWARE house Martech has introduced two new games for the Amstrad, both scheduled for release this month.

The Battle Begins is the first in a series of computer games based on the range of Zoid robot toys, UK sales of which have topped £2 million.

Developed in conjunction with toy manufacturers Tomy and comic publishers Marvel, it centres on a battle for supremacy between two warring factions of the dinosaur-like Zoids. Price is £7.95.

TV personality and strong man Geoff Capes has helped Martech develop a new game based on his exploits.

It involves bringing Geoff to the peak of physical fitness for each of several demanding events.

Once an event is underway, success depends on timing and quick reactions.

Geoff Capes Strongman costs £7.95.

BUMPER CHANCE FOR PINBALL WIZARDS

PINBALL wizards get a chance to win their own "table" when they buy the latest release for the Amstrad from PSS.

The software house has signed an agreement with French company Ere Informatique for its pinball computer game Macadam Bumper.

Included with the game is a program which enables users to design their own table. They can then be entered in PSS's Best Pinball Table Design competition.

First prize is a pinball machine and a gold cassette containing the winning design. The 10 runners up will receive a similar cassette and vouchers worth £25.

Released for the Amstrad at the same time by PSS is The Covenant, which the company describes as a multi-screen extravaganza. The two games cost £7.95 and £6.95 respectively.

Sole British rights to PPS's adventure Wizard have been won by Ariolasoft UK, who will also be responsible for other future releases. Price is £8.95 on disc.

EXTRACTING CHARGES

DESIGNED to take the pain out of calculating National Health Service charges for dentists is a new program for the Amstrad called the Charge Master from Dentron Computers.

Its cassette program calculates charges in seconds and allows professional estimates to be printed out.

Should there be any changes in NHS charges the company says it will provide low-cost updates.

Price of the system, which includes the Amstrad Charge Master program and a printer, is £399.85.

Tolkien spoof

THE spoof adventure Bored of the Rings is to be relaunched by Silversoft.

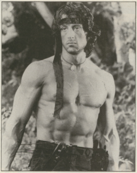
Available for Amstrad users shortly, the newly-packaged game will be an expanded version of the original, incorporating new locations and graphic scenes.

A parody of JRR Tolkien's classic trilogy Lord of the Rings, it uses mainly the standard verb-noun format.

But it will also accept less conventional commands and players will have to make full use of their wits, imagination and vocabulary, says the software house.

Adventurers will encounter many difficulties and challenges on their journey through locations such as Berkwood and across obstacles like the River Anadin.

The Amstrad version is expected to sell at £7.95.



Sylvester Stallone as Rambo, now to star in an Amstrad game

Rambo goes micro

CASHING in on the publicity surrounding the movie hit Rambo, Ocean Software has acquired exclusive rights to adapt it as a game in the UK.

The Amstrad version is due out this month.

Rambo is described by the film makers as a Vietnam veteran and "mean killing machine" who plunges into the south east Asian jungles to free American prisoners of war. There has been a move to ban it in this country.

"We are not worried about the publicity surrounding the launch of the film in the UK", said Ocean director Jon Woods, who was responsible for negotiating the deal.

"Games of this kind are fantasy – addictive and entertaining but still only fantasy".

What concerns him more is that the story line provides the company with enormous scope for computer graphic scenarios. "Scenes like the slime pit when Rambo dangles over the edge will make a superb arcade sequence", he said.

Also on the movie front, James Bond turns to Amstrad users for help in Domark's newly launched A View to a Kill – the Computer Game.

The player assists the secret agent in his fight against the insane Max Zorin who is seeking to control the world's microchip empire.

The cassette version costs £10.99 and disc £12.99.

Prices up

RISING costs of materials and labour are blamed by Addictive games for its having to increase the price of three games for the Amstrad from £7.95 to £8.95.

One is Football Manager, with sales of over 130,000 to date. The others are Software Star and Stringer.

There'll be bargains galore at the Big Show!

EXCITEMENT is growing throughout the Amstrad world about the first-ever show to be held exclusively for Amstrad users.

Because of its importance to users everywhere, it is being supported by the two leading user magazines – *Computing with the Amstrad* and Amstrad's own magazine, *Amstrad User*.

All the leading software houses and manufacturers of Amstrad peripherals will be there, showing their latest products and asking you, the users, what additional goodies you would like to see produced for your Amstrad computer in the future.

The Amstrad Computer Show is being held at the Novotel Exhibition Centre in Hammersmith on Saturday and Sunday, October 5 and 6, and will be open both days from 10am to 6pm.

Look at the some of the many things you'll be able to do at the show:

★ SEE – and try out for yourself – the whole range of Amstrad computers, including the latest CPC6128 and PCW8256.

★ TALK to some of Britain's leading Amstrad experts, who'll be on hand throughout the show giving free advice on hardware and software problems.

★ DISCOVER for yourself the latest exciting ways of linking your Amstrad to the outside world – including a satellite hook-up to a giant American database.

★ BE AMONG the first to see the fantastic collection of new-season products that users everywhere will be raving about between now and Christmas.

★ CASH IN on the biggest bargains ever offered for Amstrad hardware and software – never before have users been presented with such a wide range of money-saving offers.

Admission to the show is £2 for adults and £1.50 for children.

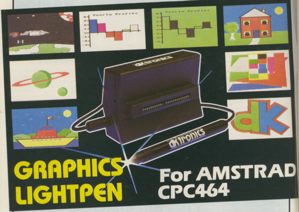
But included in this issue is a voucher, valid for up to four people, giving a 50p per head reduction.

Additional vouchers can be obtained by sending a stamped addressed envelope to:

***Amstrad Computer Show,
Europa House,
68 Chester Road,
Hazel Grove,
Stockport SK7 5NY.***

There are also specially reduced prices for school and college groups. For details phone 061-480 7863.

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 demonstrate 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 85

Now let's string along a little bit further...

PETE BIBBY solves a four letter word problem in Part 9 of our series for beginners

IF you cast your mind back to last month, you'll remember that we dealt with strings. In particular you'll recall that we learnt about the Basic functions CHR$, ASC and LEN.

So if I want you to find the LEN of a concatenation of CHR$ I hope you'll understand what I'm Asciiing you for.

```
10 REM PROGRAM I
20 entry$=""
30 WHILE LEN(entry$)<>4 OR VAL(entry$
)<>0
40 PRINT "Enter a four letter word ";
50 INPUT entry$
60 WEND
70 PRINT entry$
```

Program I

And if you've understood all that, this month's Program I – last time's Program X – should cause you no fears.

The problem was that this program, designed to allow only four letter words, accepted 1234 as input. The challenge was to mugtrap it so 1234 was not allowed. I hope you were able to see that a line such as:

```
55 IF ASC(entry$)<65 THEN GOTO 40
```

partially solved the problem. Now the program won't accept *entry$* beginning with numbers but will accept things such as q234. We'll be able to deal with this when we come across more of the string-handling commands in a future article.

This month we'll be taking a further look at string variables and seeing how they can be changed into numbers and vice versa. We'll also be learning why we should want to do it in the first place.

Before we start, however, it's important that you grasp the difference between a number and a string. If you can't explain the different results obtained from:

```
PRINT 2+2
```

and:

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

then you'd better re-read some of the very first articles. However by now most of you will see that the first PRINT takes the 2+2 as numbers, adds them and displays the answer.

The second PRINT finds the inverted commas and displays, unchanged, everything that follows until it comes across another set of inverted commas.

To be formal, 2+2 is treated as an expression, or sum, while "2+2" is treated as a string, the inverted commas being the delimiters. As you'll remember, a string is just a collection of letters, punctuation marks, spaces and numbers, all treated as one lump. That may not be a technically exquisite description but it works.

We tend to think of strings as collections of letters such as:

```
name$="Peter"
```

where the string variable *name$* holds the letters that make up my christian name. However, combinations of letters and numbers such as:

```
name$="C3PO"
```

are allowed, and even strings that consist completely of figures, as:

```
name$="666"
```

shows. It's strings made up of combinations of letters and numbers, and the Basic functions that deal with them, that we're concentrating on this time.

The first function we'll look at it STR$. This is used to change a number into a string. To convert the number 3 into a string variable called *fred$* we use:

```
fred$=STR$(3)
```

When you enter:

```
PRINT fred$
```

you'll find that it now contains 3. Notice, however, that it's a string variable. The Amstrad won't like it if you try to do maths with it.

```
PRINT fred$+3
```

results in the "type mismatch" error message, showing you you've tried to use a string as a number.

STR$ not only works on numbers, it will work on numeric variables (variables that hold numbers) and expressions as well. Hence:

```
number=24
number$=STR$(number)
```

puts 24 into the string variable *number$* while:

```
result$=STR$(72/12)
PRINT result$
```

shows that the string variable *result$* holds the character for 6.

Notice that in the last case the expression inside the brackets is worked out before STR$ converts the lot to a string.

Using STR$ on negative numbers helps to highlight the difference between strings and numerics. It should come as no surprise to find that entering:

```
abc=-1.23
fred$=STR$(abc)
```

results in the string variable *fred$* holding the string −1.23.

There's one point to bear in mind when using STR$ to turn numbers into strings – spaces count as characters. If you find the length of *fred$* with:

```
PRINT LEN(fred$)
```

it's given as five. This makes sense as the string consists of the negative sign and the decimal point as well as the figures. All count as characters.

However, set up a string with:

```
abc=123
fred$=STR$(abc)
```

and then find the length of *fred$* with:

```
PRINT LEN(fred$)
```

The result is 4, and not 3 as you might expect. This is because, for reasons best known to Locomotive, who wrote the Basic, STR$ puts a space in front of any positive number it turns into a string. Hence the length of *fred$* is 4, made up of one space followed by three figures. This invisible extra character can lead to problems if you're not careful when using STR$.

And if you've followed all that, try explaining the result of:

```
PRINT LEN(abc)
```

The more thoughtful, or perhaps cynical, reader may be wondering what the point of all this is. After all, you can put a number into a string without using STR$. You can use an INPUT, as Program II shows.

This amazingly trivial program asks you for a number and a street name and then prints out the address. It's hardly epic programming but it does make a point or two, so let's take a closer look at it.

Line 20 asks for the house number and stores it in the string variable *number$*. There we have an example of a number being turned into a string courtesy of INPUT.

```
10 REM Program II
20 INPUT "House number? ",number$
30 INPUT "Street name? ", street$
40 addres$=number$+CHR$(32)+street$
50 PRINT "The address is "addres$
```

Program II

The next line puts the street name in *street$* and line 40 concatenates (joins) the two together, putting the result in *address$*. The CHR$(32) sandwiched between them is just there to separate the number from the name.

Notice that *address$* is holding two pieces of information – the street and the house number – in just one variable. What took two variables to hold is now contained in one. There'll be more about this later.

As you'll find out if you try:

```
PRINT number$+2
```

while *number$* may hold a number, you can't do sums with it. In the program above this may be all right but for more complicated examples we may need to use the house number in our calculations.

Program III is just such a one. It asks for a street and number as before, but now the FOR ... NEXT loop ensures that the next 25 addresses on that side of the road are displayed.

```
10 REM Program III
20 INPUT "House number? ",number
30 INPUT "Street name? ", street$
40 FOR loop=number TO number+50 STEP
2
50 addres$=STR$(loop)+CHR$(32)+street
$
60 PRINT "The address is "addres$
70 NEXT loop
```

Program III

It's very similar to Program II, but notice that now the house number is held in the numeric variable *number*. This is because we'll be using *number* to determine the values that loop control variable takes.

Try changing all the *numbers* to *number$* and see what happens. The program crashes because you've tried to do calculations on a string. Never mind that the string may contain a number, you can't use it as a number, only as a string.

Line 50 has us using STR$ for the first time in a program. Here it takes the value of *loop* (increasing by 2 each time round the loop) and turns it into a string courtesy of STR$. This is then promptly joined with a space and *street$* as before and stored in *address$*. Line 60 prints out the information stored in *address$*.

Of course it's not a very practical program, but it does have potential. It should only take a little imagination to see how it could be used as the basis of a larger program which kept track of, say, a paper round. Instead of displaying all the addresses on screen they could be printed out or, better, saved to tape or disc for further use.

Before we leave Program III I'd like to make two points about it. The first is that, unlike Program II, the street number is kept in a numeric variable *number* and not a string variable, *number$*.

While it doesn't make too much difference in this case, I much prefer Program III's way of arranging the variables. After all, you expect to find a number in a numeric variable, while a string might contain all sorts of characters.

This may not be a problem in the above examples, but in long, complicated programs keeping numbers in numeric variables and non-numerics in string variables can save some elementary but time-consuming errors.

The second point is that after what I've just said about keeping them separate, I've used STR$ to make *number* into a string! This may seem a bit contradictory but it's not, honestly.

While I advocate keeping numbers as numerics and non-numerics as strings when you're using them, I don't object to numbers being turned into strings so they can be stored more efficiently.

In Program III *address$* manages to hold the information from two variables *(number,street$)* in one variable. This can be quite a saving, as Program IV shows.

This gem of the programmer's art calculates the wages for three

```
10 REM Program IV
20 FOR loop=1 TO 3
30 INPUT "Name? ",name$
40 INPUT "Age? ", age
50 wage=age*1000
60 PRINT name$,age,wage
70 NEXT loop
```

Program IV

‘The trade off between flexibility and efficiency becomes a headache’

employees using a rather strange formula. The pay is £1,000 for each year of age. No doubt in this concern they expect the work to kill you off young!

You should have no difficulty following how it works. The main body of the program is a FOR ... NEXT loop which cycles three times. Each time round the loop an employee's name is entered and stored in *name$*. Similarly the employee's age is held in *age*. Line 50 calculates the pay by multiplying *age* by 1000 and putting the result in *wage*. Line 60 prints out the details.

Can you arrange them a little more neatly, maybe in columns with headings?

As you'll have seen if you've run it – and if you haven't you should have – the program works. But, having said that, it's all you can say about it. Does it really need three separate variables to keep track of things?

```
10 REM Program V
20 FOR loop=1 TO 3
40 INPUT "Name? ",name$
50 INPUT "Age? ",age
60 wage=age*1000
70 record$=name$+STR$(age)+STR$(wage)
80 PRINT record$
90 NEXT loop
```

Program V

Line 60 is just printing out a simple message but it has to search for three variables to find the data it wants. Program V does the same job, but does it in a different way.

This program only uses one variable, *record$*, to keep track of all the information about an employee. Each time round the FOR ... NEXT loop lines 40 and 50 use INPUT to store the employee data in *name$* and *age*. The next line calculates the pay as before, recording it in *wage*. Line 70 is the one that makes the difference.

Here STR$ is used to turn the numeric variables *age* and *wage* into string variables. These are immediately concatenated with *name$* and the whole lot is stored in *record$*. So three pieces of information are now held in one variable instead of three variables as before.

It's a lot more efficient way of storing data, although you might have a bit of trouble adding table headings to neaten the display. And efficient as it is, beware of one thing.

When line 80 displays *record$* the employee's name, age and wage are neatly separated by gaps. This is because the STR$ function of line 70 has introduced a space before each of the figures. In this case it's worked out, but if you use this method of holding data in one string, beware.

If you concatenated two normal strings they'll just be "glued" together and there'll be no space to show where the "join" is.

If you don't see what I mean, change line 70 to:

```
70 record$=name$+STR$(age)
+"anystring"+STR$(wage)+"anystring"
```

Now when you run Program V you'll see that the information is messed up. It's difficult to see the ends of the age and the wage figures. The fields of some of the records, as they are known, are joined.

So bear in mind that a few spaces added during a concatenation might make things more intelligible. We'll be dealing with these fields when we come to using the LEFT$, RIGHT$ and MID$ functions.

While Program V is arguably more efficient than Program IV, it doesn't have its flexibility. With Program IV it would be easy to print out the employee data in the order age, name, wage if we wanted it that way. We'd just change the order of the variables in line 60, leaving the main structure of the program intact.

With Program V, however, we'd have to change line 70, altering the way the data is stored in *record$*. This problem can be eased using some of the string-handling techniques we haven't covered yet, but even so the changes aren't that simple.

So while the program may be efficient, it's not so flexible. You'll find as your programming experience grows that this trade off between flexibility and efficiency becomes a regular headache. It's up to each programmer to choose, though as computers get faster and memories larger and cheaper I suspect flexibility will become prized over efficiency.

For the time being, however, just notice how STR$ has been used to turn numbers into strings and store them in another string.

Information from several variables is compacted into one string variable. Later on we'll learn how to search these data strings for their information.

For the moment see if you can neaten up the display of Program V. Is there any way of getting rid of the annoying questions that come between the display of the contents of *record$*? One solution lies with the control characters we dealt with last time. The following additions to Program V will make things neater. Can you explain how they work?

```
15 record$=""
70 record$=record$+CHR$(13)+CHR$(10)+
name$+STR$(age)+STR$(wage)
80 CLS
100 PRINT record$
```

Now have a look at Program VI, a yet more efficient version of Programs IV and V. It is shorter and uses fewer variables than the other two, yet does the same job. As you can see, there's no variable *name,record$* being used instead. Similarly, rather than have a separate line and variable for calculating and storing the wage, STR$ is used with the expression *age*1000* in line 50.

```
10 REM Program VI
20 FOR loop=1 TO 3
30 INPUT "Name? ",record$
40 INPUT "Age? ", age
50 record$=record$+STR$(age)+STR$(age
*1000)
60 PRINT record$
70 NEXT loop
```

Program VI

While this may be a more efficient program, I don't like it all that much. Not only has it lost its flexibility, it's also a lot harder to understand. Dropping the variables *name$* and *wage* tends to hide where things are happening.

When I come back to look at the program in a month's time how long will it take me to find where the wages are calculated? And if I start messing around with the program, will I realise that the *record$* of line 30 contains a completely different set of information from the *record$* of line 50?

I think of the three I prefer Program V. It does the job, as far as I'm concerned, it does it at an acceptable level of efficiency, flexibility and comprehensibility. Also, its method of storing numbers in strings brings us onto the next topic.

So far we've been busy converting numbers into strings using STR$. Using it we've seen that we can combine a lot of numeric and string variables into one long string variable packed with information.

But what if, once we've converted our numbers into strings, we want the numbers back again? Is there a way for converting strings, or parts of them, back into numbers? It's a good question.

Of course, the ASC function we covered last time could be said to convert a string to a number, but that's not what we want. While:

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

may give 65, so does:

‘Is there a way of converting strings back into numbers?’

```
PRINT ASC("AB")
```

and:

```
PRINT ASC("ABC")
```

Similarly,

```
PRINT ASC("1")
```

gives the same result as:

```
PRINT ASC("12")
```

which is hardly going to be much use extracting numbers from where we've stored them in strings. What we want is the aptly named VAL function. This turns the numeric part of a string back into a number again so:

```
PRINT VAL("12")
```

gives 12 and, to show that we really have transformed it into a number and that you can do maths with it, try:

```
PRINT VAL("12")*12
```

which gives 144.

For VAL to work on a string that string has to begin with a plus or minus sign or a number. If it begins with anything else, you get a 0 for your trouble. So:

```
PRINT VAL ("-56.67")
```

gives –56.67 while:

```
age$="+7"
age=VAL(age$)
PRINT age
```

gives 7. Incidentally, this last example shows that VAL can work on a string variable.

Beware, however, of strings that contain expressions, as VAL only works on the first number. Hence:

```
sum$="25+45"
PRINT VAL(sum$)
```

only gives 25 as the answer. And remember that it must start with a number or plus or minus sign. Try:

```
nono$="*123"
returned=VAL(nono$)
PRINT returned
```

and you'll get 0 returned. Similarly:

```
PRINT VAL ("BODGER")
```

gives 0 as it starts with a letter. Notice the difference between:

```
PRINT VAL("PETE34")
```

and:

```
PRINT VAL("34PETE")
```

The first returns 0 as the string starts with a letter, while the second returns 34. VAL takes all the numbers it can and then ignores the rest.

The VAL function allows another partial solution to the problem posed by Program I. Using it, a line like:

```
55 IF VAL(entry$)<>0 THEN GOTO 40
```

solves the problem as efficiently as using ASC.

And that's all we're going to cover this time. We've seen how to use STR$ to convert numbers into strings and VAL to do the reverse. We've also come across an extremely compact way of storing information, one which we'll be dealing with a lot more.

See if you can use what we've covered in the past couple of months to create your own program to store and display information. And, if you've got any time left after that, can you mugtrap the above programs and maybe make Program VI even more efficient (and less comprehensible)?

That should keep you busy until next time, when we'll leave strings for a while and READ all about DATA.

It's all as easy as pi...

Aleatoire demonstrates how the Amstrad makes out performing the calculation that took Eniac more than 70 hours

WHICH of the following is the easiest to do? Have a conversation. Play a good game of chess. Or calculate pi to 200 places?

From a computer point of view, having a chat is well nigh impossible, and to play chess well requires very large – thousands of lines – programs. BUT calculating pi is a piece of cake.

Type in Listing I in Mode 2 and without the REMs. To test it, type RUN and, in reply to the prompt A?, enter 20 – which means calculate pi to 20 places using six significant digits per word of the array *p*.

The program will report the diminishing times for 12, 18 and 24 places of accuracy and, after 5.52 seconds, you should get the answer.

3.141592653589793238462636

The next test is to enter 770 – that is 770 places and again six places per word. This will take a lot longer but notice how the program really speeds up as it approaches the finish – clever isn't it?

The check of this test is that the digits 761 to 766 are "999999" – a remarkably unrandom sequence so early.

If that works, then you have got the program right and can now enter the big question – 2000 – to see just how your Amstrad, using Locomotive Basic, compares with the giant Eniac, the first computer to perform this calculation back in 1949 – it took 70 hours.

To explain how the program works in detail is beyond the scope of this article. However the program does use the same formula as the Eniac, namely Machin's – see last month – where:

```
PI/4=4*ARCTAN(1/5)-ARCTAN(1
/239)
```

which expands – using Gregory's

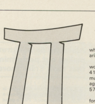
```
10 REM PI
20 REM(c)Computing With The Amstrad
30 DEFINT a-z:DEFREAL c,m,p,r
40 DIM p(2000):t=500:f=1000:s=1500:GO
TO 310
50 c=0:REM x=x/y
60 FOR i=x+skip TO x+a
70   r=p(i)+c
80   p(i)=INT(r/y)
90   c=m*(r-p(i)*y)
100 NEXT i
110 RETURN
120 c=0:REM x=x*y
130 FOR i=x+a TO x STEP-1
140   p(i)=p(i)*y+c
150   c=INT(p(i)/m)
160   p(i)=p(i)-c*m
170 NEXT i
180 RETURN
190 c=0:REM x=y+z
200 FOR i=x+a TO x+skip STEP-1
210   p(i)=p(i+y)+p(i+z)+c
220   IF p(i)>=m THEN p(i)=p(i)-m:c=1
 ELSE c=0
230 NEXT i
240 RETURN
250 c=0:REM x=y-z
260 FOR i=x+a TO x+skip STEP-1
270   p(i)=p(i+y)-p(i+z)-c
280   IF p(i)<0 THEN p(i)=p(i)+m:c=1
ELSE c=0
290 NEXT i
300 RETURN
310 INPUT"A";a:d=6:REM Start of progr
am
320 a=INT(a/d)+1
330 m=10^d
340 runtime=TIME
350 p(f+1)=INT(0.8*m+0.5):p(s)=1:REM
f=0.8
360 x=s:y=239:GOSUB 50:REM s=1/239
370 x=p:y=f:z=s:GOSUB 250:REM p=0.8-1
/239
380 sign=-1:flag=-1:n=3:skip=1
390 WHILE flag OR p(f+a)>0
400   IF p(f+skip)=0 THEN skip=skip+1
:PRINT skip*d;(TIME-runtime)/300
410   x=f:y=25:GOSUB 50:REM f=f/25
420   IF p(s+a)=0 THEN 440:REM s part
=0
430   x=s:y=239:GOSUB 50:GOSUB 50:REM
 s=s/(239*239)
440   x=t:y=t:z=f:GOSUB 250:REM t=f-s
450   y=n:GOSUB 50:REM t=t/n
460   x=0:y=0:z=t:REM p=p+t else p-t
470   IF sign=1 THEN GOSUB 190 ELSE G
OSUB 250
480   sign=-sign:n=n+2:REM Next term
490   IF p(f+a)>0 THEN flag=0:REM Nea
rly done
500 WEND
510 x=p:y=4:GOSUB 120:REM  p*4
520 PRINT p(0);".";
530 FOR i=1 TO a
540   PRINT USING"######";p(i);
550 NEXT i
560 PRINT (TIME-runtime)/300
```

series – into:

```
PI/4=4*(1/5-1/(3*5^3)+1/(5*
5^5)-1/(7*5^7)+...1/(N*5^N)
...-(1/239-1/(3*239^3)+1/(5
*239^5)-1/(7*239^7)+...1/(N
*239^N)...
```

and it is the summation of these two series in the WHILE ... WEND loop that is the heart of the multilength calculation.

The program is essentially a conversion from a high-level language into Basic and is consequently rather crude. For example there is no way of passing any variable – let alone an array – as a parameter to a GOSUB call, so I have used three global pointers, *t, f, s,* into the global array *p* to represent the successive terms of the 1/5 and 1/239 series and their accumulating sum.



To make the program more interesting – and portable – you can define how many significant digits of the result can be represented in each element of the array. This number *D* – assigned in line 310 – can be 1, 2, 3, 4, 5, or 6 where the higher *D* the faster the program runs. Two interesting questions now arise which, indirectly, test the machine's arithmetic.

The first is, why doesn't *D*=7 work? And the second concerns line 410 which is, effectively, dividing a multilength number by 239 and then again by 239. Why not just divide by 57121 once?

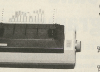
An even harder problem – strictly for whizzkids – is to re-program the calculation of pi in machine code to get a genuine comparison between the Amstrad and Eniac.

My estimate is that the Z80 could do 2,000 places in about five minutes and is, therefore, comparable in power to mainframe computers of the early 1960s.

Finally it is a salutary thought that the main task of the Eniac, between 1945 and 1949, was numerical calculations to help design more-efficient atomic bombs.

OPEN SUNDAYS 10-1

more unbeatable deals from Data star Systems!



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

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!!!

Just a few of the many features:—

- Easily switchable between Epson & IBM graphic/print models.
- Will print all ASCII codes from computers that can only send 7 bits on their parallel interface such as Amstrad CPC 464/664 and Apple II.
- Compatible with all word processing programs.
- 50 CPS NLQ mode available from switch on.
- 120 CPS draft mode.
- 2K print buffer — expandable to 10K

AMAZING AMSTRAD WORD PROCESSING OFFER ONLY FROM DataStar Systems

Amstrad CPC 6128 green screen computer ...	£299.00
Tasword/Mailmerge W/P package	£24.95
Star SG-10 printer package as above	£347.85
Next day doorstep delivery service	£20.00
Total value:	£691.80

DataStar's Superdeal Price £619.95!!!

Many more package deals available on other Amstrad models and makes of micro's.

All goods despatched FREE OF CHARGE by next day doorstep courier service

01
482
1711

24 Hour Credit Card Hot Line

Post your cheques to

Datastar Systems UK

Unicom House, 182 Royal College Street,
London NW1 9NN.
Telephone: 01-482 1711 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*

A significant bit of sound programming...

NIGEL PETERS explains the numbers behind the notes in – Part VIII of his series on Amstrad sounds

WE'VE come a long way in our exploration of the channel parameter of the SOUND command. We've seen how values of 1, 2 and 4 can be used to produce notes on channels A, B and C respectively.

Also we've learnt how they can be combined so that:

```
SOUND 7,100,100,7
```

produces the same note on all three channels.

We've found that adding 8, 16 or 32 causes the sounds to rendezvous with notes on channels A, B and C. This means that when we try to produce a note on channel A with:

```
SOUND 33,100,200,6
```

we won't hear anything until we give it a rendezvous note on channel C using:

```
SOUND 12,200,200,5
```

You'll remember that both notes have to be tagged with the rendezvous parameter. Using a note such as:

```
SOUND 4,200,200,5
```

wouldn't have freed the note on the channel A queue. Try it and see.

Talking of the channel queues, each of which can hold four notes in addition to the one that's playing, leads us to the hold parameter.

By adding 64 to the channel parameter of a SOUND command we are able to ensure that when that note reaches the head of the queue it waits until it's released.

In practice this means that when we enter something like:

```
SOUND 1,300,100,7:
SOUND 65,400,100,7
```

we only hear the first note. The second note stays in the channel A queue until it's conjured up with:

```
RELEASE 1
```

Finally we've seen how we can join all these values into one combined parameter so that:

```
SOUND 81,200,100,7
```

produces a note on channel A that is not only held but also waiting for a rendezvous note on channel B. To bring the timid thing out of hiding we have to free it with a:

```
RELEASE 1
```

and then whistle it up with a note on channel B such as:

```
SOUND 10,1000,100,3
```

I hope that you've been messing around with the channel parameters of the SOUND command. There's no better way to learn. With the Amstrad's sound chip, practice is a lot better than theory. No doubt you've managed to get your channels in a twist more than once. The small Enter key, set up with:

```
KEY 139,"SOUND 135,0,0,0"+CHR$(13)
```

has probably taken a lot of hammer as you tried to escape from a cacophony or a wall of silence.

However, while you've been using it, have you noticed that there's something strange about the channel parameter of the SOUND command? It's 135, a lot higher than any we've used previously.

As you might have guessed, it's yet another number that can be added to the channel parameter. This time it's one that causes the relevant channel or channels to be flushed or cleared of notes.

Try a sound such as:

```
SOUND 1,400,4000,7
```

Monotonous isn't it? Suppose now that you wanted to have another note playing on channel A. In the middle of a program you couldn't just hit the small Enter key. How would you do it?

Using:

```
SOUND 1,200,100,7
```

has no effect, it just gets stuck on the channel A queue and the first note carries on until it's finished.

Why not use the flush parameter, 128? We want to clear channel A, which has parameter 1, so the channel parameter we need is 129 (1+128). This means that:

```
SOUND 129,200,100,7
```

will do the job, stopping the first note dead in its tracks. Now there's a note of pitch 200, loudness 7 playing for one second.

In other words the flush parameter gives that note priority over any other note that's playing or waiting in the queue. A channel parameter of 129 clears all the other notes on channel A out of the way while 130 (2+128) and 132 (4+128) do the same for channels B and C. This is the case even if the notes on the queue are held or rendezvoused.

Entering:

```
SOUND 66,100,100,7
```

holds a note on the channel B queue. However:

```
SOUND 130,400,200,7
```

gets rid of the first note – and any others that might be on the channel B queue – and plays one with pitch 400, loudness 7 for two seconds.

Similarly:

```
SOUND 12,100,100,7
```

is waiting for a rendezvous with a

note on channel A. Any other notes on channel C will just have to take their places behind it in the queue until its date turns up. That is, of course, unless it is the queue-jumping:

```
SOUND 132,1000,300.6
```

which gets rid of all the other notes and plays for three seconds.

As with our other parameters we can mix and match them. A value of 131 – 1+2+128 – clearing the channel A and B queues, while 133 – 1+4+128 – does the job for A and C. As you might guess:

```
SOUND 135,0,0,0
```

sweeps the notes from all the channels. Playing, held, rendezvoused or merely waiting, they all go. However since this last note has a pitch, volume and duration of zero, no other note is played. In effect:

```
SOUND 135.0,0,0
```

gets rid of all previous SOUND commands. As you've found when you've had recourse to the small Enter key, this can be a blessed relief.

And the flush parameter, you'll be glad to know, is the final one we'll meet in our treatment of the channel parameter.

Table I sums them up.

number	bit set	result
1	0	uses channel A
2	1	uses channel B
4	2	uses channel C
8	3	rendezvous with A
16	4	rendezvous with B
32	5	rendezvous with C
64	6	hold until RELEASed
128	7	flush the channel

Table I: Channel parameter values and actions.

Notice that in the table there are some values labelled "bit set", ranging from 0 to 7. The reason that these are shown is that the channel parameter of the SOUND command is what is known as "bit significant".

This means that when the number, which is usually in decimal, is translated into binary, the 0s and 1s of the binary number are used as flags to switch different aspects of the channel parameter on and off.

If you can't remember too much about binary, I refer you to Mike Bibby's Bits and Bytes in the first issue of *Computing with the Amstrad*. However, you don't have to know too much about it to understand the following.

To show what I mean, let's just take the simple case where the channel parameter of a SOUND command is 1. Now the binary equivalent of 1 is, not surprisingly, 1 or rather 00000001 when we pad it to eight figures. If you don't believe me, maybe you'll believe your micro when:

```
PRINT BIN$(1,8)
```

produces:

```
00000001
```

Now each of these eight figures makes up one bit of what's known as a binary byte. The bit furthest to the right – in this case a 1 – is known as bit 0, the one to its left – here a 0 – as bit 1, the next as bit 2 and so on until the final bit, bit 7. Figure I shows the bit numbers in a byte.

bit number	7	6	5	4	3	2	1	0
binary byte	0	0	0	0	0	0	0	1

Figure I. Bit positions of binary 1.

Let's take the binary byte for 2 which, as we'll find from:

```
PRINT BIN$(2,8)
```

is:

```
00000010
```

Here bit 0 is 0, while bit 1 is a 1, the remaining bits (2 to 7) all being zero. Figure II shows this.

bit number	7	6	5	4	3	2	1	0
binary byte	0	0	0	0	0	0	1	0

Figure II: Bit positions of binary 2.

I don't know about you, but I find talking about bit 0 as 0 and bit 1 as 1 a bit confusing! I prefer to say that a bit is "set" if it's a 1 and "clear" if it's a 0. So, in this last example, only bit 1 is set – 1 – all the others are clear – 0.

Going on to our last channel parameter, 4, a quick:

```
PRINT BIN$(4,8)
```

gives:

```
00000100
```

Bit 2 is set, the rest are clear.

Now compare these with the "bit set" column in Table I. With a channel parameter of 1, we've seen that bit 0 is set. Looking at the table it says that the note uses channel A. So if that bit is set channel A is used.

Similarly, if bit 1 is set, B is used and if bit 2 is set, C is the selected channel. As you can see if the bit is set, then the result next to it occurs. This is what is meant by saying that the channel parameter is bit significant. Different actions occur according to which bit of the binary byte is set.

Now suppose we have a note playing on all channels with a parameter of 7. In binary this is 00000111. Try:

```
PRINT BIN$(7,8)
```

if you doubt me.

As you can see, bits 0, 1 and 2 are set. Looking at the table we can see that this means that notes will be produced on channels A, B and C.

It's the same for all the other parameter values up to 255. If you translate them into eight bit binary numbers you can see which functions are switched on and off by the individual bits.

Let's look at one last example. We've seen that a channel parameter of 135 clears all the channels. Now:

```
PRINT BIN$(135,8)
```

gives us the byte:

```
10000111
```

Looking at this bit by bit shows that the flush action started by bit 7 being set will affect channels A, B and C – as bits 0-2 are also set.

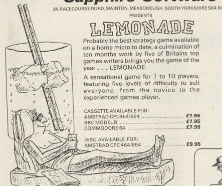
Try out other parameter values and examine their binary equivalents to see what will happen. It's fascinating to see how the action of a channel parameter of, say, 97 can be read from its binary equivalent – 0110001. Here bits 6, 5 and 1 are set, so it's obvious from Table I that we have a note held on channel A waiting for a rendezvous with one on channel C. Until you think about it, it seems almost uncanny that the binary version of a decimal number so closely relates to the channel parameter.

And that's it for this month. By the time you understand the significance of bit significance you can leave channel parameters and go on to the SQ command. Your knowledge of the SOUND command will have increased a significant bit.

Sapphire Software

86 RACECOURSE ROAD, SWINTON, MEXBOROUGH, SOUTH YORKSHIRE S64 8DR.

PRESENTS



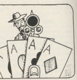
LEMONADE

Probably the best strategy game available on a home micro to date, a culmination of ten months work by five of Britains top games writers brings you the game of the year . . . LEMONADE.

A sensational game for 1 to 10 players, featuring five levels of difficulty to suit everyone, from the novice to the experienced games player.

CASSETTE AVAILABLE FOR:	
AMSTRAD CPC464/664	£7.95
BBC MODEL B	£7.95
COMMODORE 64	£7.95

DISC AVAILABLE FOR:	
AMSTRAD CPC 464/664	£9.95



KLONDIKE GOLD

Howdy partner . . . in this adventure you're after gold, but it's not buried in the desert, no siree bob, no, it's in the bank!

First you have to find the hidden combination to the safe and then execute a daring robbery on the bank, avoiding jail and the noose.

MYSTERY MANSION

Dare you enter the mysterious mansion, 'Nebula', on a quest for a fabulous treasure. Your hunger for wealth and to succeed where others have failed urges you deeper into the heart of the mansion, where danger lurks around every corner, only your skill and ingenuity can save you now . . .

CASSETTE AVAILABLE FOR:		DISC AVAILABLE FOR:	
AMSTRAD CPC 464/664	£6.95	AMSTRAD CPC 464/664	£9.95
BBC MODEL B	£6.95		

THESE GAMES ARE AVAILABLE FROM ALL GOOD RETAILERS OR DIRECT FROM SAPPHIRE SOFTWARE

PLEASE SEND ME

□ LEMONADE	□ CASSETTE	□ 3" DISK	NAME ....
□ MYSTERY MANSION	□ CASSETTE	□ 3" DISK	ADDRESS ....
□ KLONDIKE GOLD	□ CASSETTE	□ 3" DISK	....

I Enclose a Cheque/Postal Order for £..

Made payable to Sapphire Software

My Micro is ....

IF YOU DO NOT WISH TO DEFACE THIS MAGAZINE, ORDERS MAY BE PLACED ON PAPER

ALL PRICES INCLUDE
POSTAGE, PACKING AND VAT

Amstrad has considerably strengthened its hold on the micro market with the release of two new machines – the PCW8256 pictured here and the CPC6128 (overleaf). MIKE BIBBY and MIKE COWLEY report on what the new Amstrads offer the user and forecast their likely impact on the micro scene.

Enter the PCW 8256

Word processing at price everyone can afford

AIMED directly at the small business user, the new Amstrad PCW8256 seems set for a runaway success.

Perhaps the most astonishing feature is the price – just £399 for a system that includes a 256k computer/word processor plus green screen monitor, disc drive and printer.

There hasn't been anything like it since Sinclair ushered in the age of cheap home computing with the ZX80.

The system has been designed around one simple concept, to produce a no-nonsense, powerful word processor for business use at the lowest possible price.

First impressions are of an extremely neat looking package. Computer, green screen monitor and printer are all powered from the one plug, avoiding the morass of cables associated with other systems.

The disc drive, with 180k of storage on each side, is built into the monitor, with the option of a dealer-installed second drive which adds a further 720k.

The machine hardware has been designed around the word processor package, LocoScript, and the 82 keys of the keyboard reflect this, with such keys as Cut, Paste and Word Char.

One interesting deviation from the industry norm is the screen size – a roomy 90 characters by 32 lines rather than the standard 80 by 25.

LocoScript itself is driven by pull down menus accessed via dedicated function keys. Facilities include pagination, automatic paragraph alignment and re-alignment, headers, footers, search and replace and a printer buffer that allows simultaneous printing and editing. In other words, the sort of facilities you'd expect of a high quality word processor.

The printer, which handles both single sheet and tractor feed, is driven from Amstrad's own interface.

Although accepting the industry standard Epson control codes, the interface is not standard Centronics, though such an add-on is promised, as is a serial interface.

The printer supports the usual features like italics, boldface and underline, and can operate from 90 cps down to 20 cps in correspondence mode. Though not of the quality of a dedicated word processor, the correspondence mode does give near letter quality script.

While LocoScript itself is not a CP/M program, Amstrad has nonetheless included CP/M Plus as standard, giving the machine potential access to a vast number of existing programs.

Under CP/M the printer is configured to use standard Epson codes, and the screen emulates a standard industry terminal, so installation of existing CP/M programs should prove no problem to software houses.

As well as CP/M Plus and LocoScript, the PCW8256 comes with Mallard Basic, a CP/M-based version from Locomotive similar to the professional Basic found on the BBC Micro's Z80 second processor. Unlike Amstrad's other machines, there is no Basic in ROM.

To complete the package there's the usual Dr Logo and also the GSX Graphics System Extension as featured on the new CPC6128.

The PCW8256 has 256k of RAM based on a Z80A processor, and cunning use is made of the large memory available – 112k of it is used as a RAM disc. That is, to all intents and purposes the memory acts as another disc drive which you can save to or load from in the same way as with a disc – but with far faster access times.

To sum up, the PCW8256 is neat, easy to use and extremely good value for money. Not only does it provide a full word processor, it will, with the advent of customised software, prove to be a powerful business machine.

CP/M Plus is an upgraded version of CP/M 2.2 – the CP/M issued with the CPC464 and CPC664 – designed to take advantage of bank-switched machines.

The memory is divided into three separate banks, one for BDOS and BIOS, one for the TPA – the area your programs occupy – and one for the CCP and some workspace.

The nice thing is that once you've booted CP/M once, providing you don't press Reset the system files stay in memory. This means you can swap discs without having to worry about whether the new one has the system files on it.

All in all, the TPA now gives you about 61k of memory, which means that most CP/M programs should now run on the CPC6128.

Also included in the package is a version of CP/M2.2 to allow you to run the odd program that won't work under CP/M Plus.

Alan Sugar at the keyboard of his new micro, the PCW8256

OFF TO

AMSTRAD unveiled its latest world-beating machines to the sound of raunchy beat music in a plush theatre setting.

Billed none too modestly as "the single most important computer launch of '85", it turned out to be just that:

The actual event had been designed as an all-dancing, all-singing Las Vegas style stage spectacular, though in the end the production owed more to its venue in Aldgate in the East End of London than America's fun capital.

But for the 300 media men and industry representatives it attracted, the somewhat banal performance didn't matter. For they were able to witness not one but two

"The company has now come a long way since Alan Sugar was making Perspex lids for hi-fi equipment" – *Newscaster Richard Whitmore compereing the launch of the PCW8256 and the CPC6186.*

Now 664

AMSTRAD has now replaced the CPC664 with the CPC6128. As the name implies, it has 128k of RAM and comes with CP/M Plus and a bank manager which allows Basic to access the extra 64k.

Software written for the CPC664 should run on the new CPC6128 (unless it uses some illegal calls to the firmware) and a fair proportion of CPC464 tape software should be compatible.

In appearance the CPC6128 is quite different to the 664. It's much slimmer and narrower and the keyboard layout has been redesigned.

The 464 and 664 both had a separate numeric/function keypad and cursor key block. These have now been incorporated within the

O A GREAT START

"This will certainly screw up the electronic typewriter market. There will be a lot of people out there wished we hadn't brought out this machine" *– Mark Simon, Amstrad's sales manager on the PCW8256.*

brilliant new stars being born – the PCW8256 and the CPC6186.

The audience was transfixed as specification after specification was revealed – first by newcaster Richard Whitmore, then by Alan Sugar, Amstrad's chairman and himself a master showman.

It was if they were watching a stripper remove her clothes.

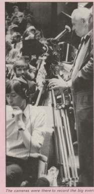
When Alan Sugar said of the PCW8256, "This machine will blow the lid off the personal computer and word processing market", there was not one person there who disbelieved him.

Gasps of astonishment were heard from usually unimpressable computer journalists when the price of £399 plus VAT was announced.

Amstrad was on to yet another winner.

"The show was lousy and probably wouldn't have run for another performance", observed one media man, "but we've seen the arrival of a couple of stars – the PCW8256 and the CPC6186 – who will be knocking them in the aisles for years to come".

"We feel we have not just one but two winners here for Amstrad" *– Malcolm Miller, Amstrad's marketing boss, on the new machines.*



The cameras were there to record the big event

4 gives way to powerful 6128

main keyboard. The cursor keys are at the bottom right and the Copy key at the bottom left of the keyboard. It feels a bit strange at first, but is easy to use after a bit of practice.

The function keypad has moved left so that it sits alongside the main keyboard.

Strangely the CPC6128 has both an Enter and Return key. The rather elongated Enter key is the same as the old Enter on the function/numeric keypad. Return corresponds to the large Enter key on the old keyboard.

The on/off switch and volume control have been moved to the rear of the case. The joystick, cassette and stereo sockets are more conveniently placed on the side of the micro. The familiar Centronics printer port, expansion port and socket for a second disc drive are at the back of the machine.

Actually the whole of the 128k is not available for Basic programs. There's still only 41k free. A utility program called Bankman has been included to enable the 64k to be used as storage space for up to four screens or as a random access filing system.

The new 6128 is capable of running both CP/M 2.2 and CP/M Plus (or CP/M 3.1). Although CP/M Plus is compatible with CP/M 2.2, the older version is provided so that applications developed using firmware features in the Amstrad CP/M 2.2 system will operate without requiring modification.

CP/M Plus is designed to take advantage of Z80 systems with an extra 64k of banked RAM. All but 3k of the main 64k of RAM is free, which means that the CPC6128 should be able to run most CP/M 80 software with none of the memory restrictions of the earlier machines.

The Graphics System Extension for CP/M Plus is provided to enable applications programs to access printer/plotters and the screen. Software will be able to take advantage of GSX to provide hard copies of graphs and diagrams.

Digital Research's Logo contains a few extensions to take advantage of the additional memory. Many new commands have been added and this is now the full version of Dr Logo.

The new CPC6128 flanked by the original CPC464 and CPC664

It's the micro with the built-in bank manager

THE CPC6128 has 128k of RAM. Unfortunately, though, not all of this is available for Basic programs. The problem is that the 8 bit Z80 processor at the heart of the machine can only access 64k of memory at a time.

On power up about 41k of RAM is free and the micro acts pretty much like the old CPC664. The other 64k is there, but can only be accessed from machine code.

Not everyone is a machine code expert so on the utilities disc supplied with the CPC6128 is a program called Bankman. This provides Basic with several additional commands in the form of RSXs which enable it to use the other 64k of RAM.

The RAM is in banks of 16k which can be switched in and out in the same way as the upper and lower ROMs, using a similar overlaying technique. Each 16k bank occupies memory addresses from &4000 to &7FFF when switched in.

As the screen memory occupies 16k of RAM each of the 16k blocks of RAM in the additional 64k can be used to store a copy of the screen. SCREENCOPY will copy one screen to another and |SCREENSWAP will swap two screens. The screens are numbered 1 to 5.

Although the commands are not fast enough for animation purposes they are reasonably quick and there are several possible uses for five screens.

It would be very easy to set up a couple of Help screens for a particular package. These would be instantly available and could be displayed at any time without corrupting the main working screen.

A business program could use separate screens to display data, spreadsheets, graphs and charts. Again each could be permanently set up and instantly recallable.

Amsdos, unfortunately does not support random access filing. However the extra 64k can be treated as a filing system for string variables which does support random access filing.

BANKOPEN sets the length of each record in the file – they must all be the same. BANKREAD is used to read a record from the filing system and BANKWRITE is used to write a record.

A pointer shows which record is to be read or written to and this can be altered to point to any position. There's no need to access a file serially – the pointer can be moved forwards and backwards without reading each record.

|BANKFIND will search through the file from any point to any point searching for a string. Wild cards are acceptable, and the position of the pointer at the first match found is returned.

The additional RAM has great potential. The commands supplied by Amstrad are just two examples of how it can be used. Once programmers get their hands on the CPC6128 some amazing software is going to be produced.

INCLUDED on the CP/M Plus utilities disc is the Graphics System Extension, which enables a CP/M program to display graphics as well as text by providing a graphic interface.

Once a call is received from a GSX application program it is converted to a call that can be understood by CP/M.

GSX cannot produce graphics displays itself. It is simply a device-independent interface which enables a graphic image to be sent to the monitor, printer or plotter without having to change the applications program for each piece of equipment.

A device driver unique to that equipment links with the application program to actually produce the display.

A selection of drivers have been included on the utilities disc. These are capable of driving Epson and Shinwa-compatible printers, Amstrad's own DMP1 printer and Hewlett Packard pen plotters in all screen modes.

Machine code can be easier than you think

FEW come to machine code easily. Most are dragged to the brink, kicking, screaming, and grasping at every available book on the subject.

But if necessity forces you to the plunge, be sure that you have **The Code Machine** by Picturesque, tucked firmly under your arm.

The tabulated assembler layout is easy on the eye, with unlimited six-letter labels.

Automatic line numbering is available and you can renumber, edit with a full copy cursor, and insert or delete lines.

It accepts all the Z80 mnemonics – plus a few unpublished ones – with extra facilities to define constants, variables, messages, and set up clear spaces or data tables.

Addition/Subtraction within operands is no problem and negative numbers are OK.

Before assembling, any lines with obvious errors are displayed, with an explanatory error message, ready for editing.

Assembly can be displayed to screen or printer, and both source and object code can be saved, loaded and – glory be – actually verified.

But, removing obvious errors and verification is no guarantee. With machine code, the best you can hope for is that it won't crash completely. Which is where the monitor comes in.

By inserting "breakpoints" – something like STOP in Basic – at appropriate points and running your program, you can then use any of the monitor's facilities, to trace problems, before continuing from the breakpoint.

For instance, the TRACE function single steps through your program, with a full display of the CPU register contents, a disassembly of the current and next instruction, the contents of the last five stack locations, and specified memory location contents.

Other facilities include displaying any memory location's contents in hex or Ascii, with the ability to alter the contents, inserting or deleting up to 255 bytes, moving memory contents to a new location or filling an area with a given value, and a hex/graphics dump to screen or printer.

A decimal/hex and hex/decimal converter makes life easier for non-mathematicians.

You can also disassemble any memory location, enable or disable the Upper or Lower ROM, and search for hex or Ascii values.

These features can also be used to examine the Amstrad's ROM or someone else's machine code.

The 70-page manual has some areas of "Huh?" but experimentation usually clarifies and the program itself is highly recommended.

Picturesque patiently answers the most stupid questions with a smile, offers updated programs to old customers at nominal prices, and its programs have the ability to make back-ups to tape or disc, built in for customers' convenience.

In other words, Picturesque are nice people – and you don't reward that sort of service by using the built-in facility to pirate copies. Or if you do, I don't think I'd care to know you.

Dorene Cox

Roland does it

JUST to prove that a superhero's work is never done, Roland has been sent on yet another exciting adventure. This time he's going boldly where no man has gone before.

Roland in Space is the latest title in the Roland saga from Gem Software, under the Amsoft label.

It's a Jet Set Willy clone on

This way, novice adventurers

Mountain Place Adventure, from Duckworth, is written using the techniques described by Peter Gerrard in his book Exploring Adventures on the Amstrad CPC464.

Since this book is available for quite a number of different computers, you'll appreciate that the techniques used are of necessity fairly basic and this is evident in the programming of the adventure.

Your task is to explore the nearby palace and collect as many of the treasures in there as possible.

You begin your quest on a ledge outside the palace. You are soon inside – a source of ignition had better have been obtained or it's back to the beginning.

The curtain will reveal a shovel if you can visualise a caveman's mating habits . . .

To get up the cliff you will need a rope. Obvious really.

The top will help you with the pit and you should now be able to map out about 60 locations.

You'll soon find a maze that is not too difficult to map and at the heart of it you will find the wizard's laboratory. What you find could be considered dirty – likes attract.

Later you'll find an avaricious guard, a greedy dog and a blacksmith.

Most of the solutions to these problems are easy ones and I doubt that the experienced adventurer will have too much trouble with them.

One annoying feature is a ghost that turns up randomly and then follows you.

Sooner or later it'll start to steal your treasures unless you kill (!) it first.

The adventure has about 101 locations and recognises about a hundred verbs and nouns.

Quite a lot of these are redundant and might have been better left out.

I must confess to being unsure quite how to classify this adventure. It has very few problems for its size, but on the other hand this is obviously an ideal feature for beginners.

Plenty of room to roam about in and get the feel of.

Overall, it is a reasonable Basic adventure that might prove ideal for the complete novice.

Paul Gardner

a grand scale with the scenario spread over the surface of seven planets.

The task before you is quite straighforward – merely collect the 158 separate parts of a super weapon and you win the game.

As you start off, you are presented with a view of the seven planets, one of which will be flashing. Using your left and right keys you may select which you wish to visit.

Your mode of transport is your trusty telephone box, as used in Roland in Time.

Having arrived at your destination, the screen provides you with a window on to the planet's surface.

You are allowed to take a peek at your surroundings by scrolling the screen a short distance in any of the four directions via the cursor keys.

Scattered around the screen are several flashing blue balls which are presumably the super weapon parts.

As is the norm in this type of game, your progress is hindered by numerous alien forms desperately trying to protect these flashing blue balls.

These chaps take various forms ranging from shoals of fish to miniature ETs, all of which are deadly to the touch.

Each of the planets appears to be divided into two distinct areas – those that can be walked and jumped on, such as solid sufaces, and those which cannot, such as sky and water.

However don't be lulled into believing that you cannot cross a space without first trying it.

On one occasion I made several fatal attempts to reach one particular ball before discovering that Roly could swim!

There are no doubt other such sneaky aspects to this game, so even if something looks impossible have a go. If you die, you have always got another nine lives to fall back on.

Roland in Space is Roland's grandest escapade yet. The excellent use of colour and the devious nature of the game combines to make this program yet another winner.

Carol Barrow

A touch of panic

Digger Barnes, by Cable, brings you face to face with Blugits, slimy animals resembling giant blueberries with teeth, but a lot more dangerous.

They hate humans, and are currently running riot in a local factory, despatching people with one deft movement of their poisonous jaws.

It's your job – as Digger Barnes – to enter the factory and clear it of Blugits.

Unfortunately, the official extermination equipment is late in arriving and you have to enter the factory armed only with a pickaxe, dig holes in the floor, wait for a Blugit to fall in, and then fill in the hole again.

Despite the original-sounding scenario, this is another version of the old arcade game Panic.

It is played out over five blue platforms, connected by green ladders of varying length.

The only innovation, apart from replacing the more-usual spade with a pickaxe, is a lift at either side of the screen.

If you are careful, you can use these lifts to move either up or down to another platform.

Blugits never use lifts, probably because they realise they are too dangerous ... Being crushed by the factory roof was a common occurrence when I played the game.

The basic format of the screen stays the same throughout, although the ladders alter when you lose a life.

You are up against two Bugits at the start, and an extra one appears for each screen you clear.

Games of this nature depend on good coordination and timing, but with Digger Barnes you are faced with two problems.

Firstly, once a hole has been started you cannot move until it has been dug, even if a creature is approaching.

Secondly, I sometimes lost one of my five lives when a Blugit was still some way away and not touching Barnes.

Both of these annoying problems reduce the program's playability.

A more serious problem occurred when I tried to use the keyboard controls. The key for up – A – did not work, although there were no problems when I used a joystick.

The program comes in a sturdy cassette box with a colourful picture on the front and clear instructions on the back. The game itself fails to live up to this excellent presentation, though.

Sound is limited and the graphics uninspired, and there is not enough variation to maintain interest for long.

A time limit for each screen and different types of mon-

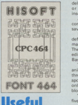
REVIEWED SO FAR

Adventure Quest	Level 9
American Football	APS
Amstradraw	B.S. McAlley
Android One	Vortex
Arnold goes to somewhere else	Nemesis
Artist & Sprite Designer	C.R.L.
Artwork	Kuma
Astro Attack	Amsoft
Amstrad Tutorial	Amsoft
Atomsmasher	Romik
Beach Head	U.S. Gold
Blagger	Alligata
Brawn Free	Nemesis
Bridge Player	C.P.
Centre Court	Epicsoft
Chopper Squad	Interceptor
Code Machine	Picturesque
Colossal Adventure	Level 9
Confuzion	Incentive
Country Cottages	Sterling
Crystal Theft	Wiccasoft
Cubit	Mr Micro
Deep Thought Chess	C.P.
Defend or Die	Alligata
Detective	Argus Press
DFM Database + Labels	Dialog
Digger Barnes	Cable
Dragon's Gold	Romik
Dungeon Adventure	Level 9
'Er*bert	Microbyte
Fighter Pilot	Digital Integration
Flexi-friend	Camel
Flight Path 737	Anirog
Flight Simulation	Myrrdin
Font 464	Hisoft
Football Manager	Addictive
Frank N. Stein	P.S.S.
Fruit Machine	Amsoft
Fruity Frank	Kuma
Forest at World's End	Interceptor
Galaxia	Kuma
Gate Crasher	Exopal
Gems of Stradus	Kuma
Ghouls	Micro Power
Happy Letters	B.E.S.
Harrier Attack	Durell
Happy Writing	B.E.S.
Holdfast	Kuma
Home Accounts Manager	Dialog
Home Budget	Kuma
Hunchback	Ocean
Hunter Killer	Protek
Jet Boot Jack	English
Jet Set Willy	Software Projects
Killer Gorilla/Gauntlet	Micro Power
Lords of Time	Level 9
Manic Miner	Software Projects
Map Rally	B.E.S.
Masterfile	Campbell
Message from Andromeda	Interceptor
Money Manager	Connect Systems
Mountain Palace Adventure	Duckworth
Mr Freeze	Firebird
Mr Wong's Loopy Laundry	Artic Computing
Music Composer	Kuma
Mutant Monty	Artic Computing
Mystery of the Java Star	Shards
Number Painter	ASK
Osprey	B.E.S.
Pinball Wizard	C.P.
Project Planner	Amsoft
Punchy	Mr Micro
Pyjamarama	Microgen
Return to Eden	Level 9
Roland in Space	GEM
Roland in the Caves	Amsoft
Roland Goes Square Bashing	Durell
Rollaball	Timeslip
Royal Quest	Timeslip
Screen Designer	DJL
Sir Lancelot	Melbourne House
Smuggler's Cove	Camel
Snooker	CDS
Snooker	Gem
Snowball	Level 9
Software Star	Addictive
Sorcery	Virgin
Splat	Incentive
Star Avenger	Kuma
Star Watcher	Amsoft
Stockmarket	Argos Press
Strip Poker	Knightsoft
Superchess	C.P.
Survivor	Anirog
Tasword	Tasman
Test Match	CRL
The Moors Challenge	Timeslip
The Prize	Arcade
The Wild Bunch	Firebird
Timebomb	Black Knight
Timeman One	B.E.S.
Timeman Two	B.E.S.
Trial of Arnold Blackwood	Nemesis
War Zone	C.C.S.
Wordhand	B.E.S.
Wordwise	B.E.S.
World Cup Football	Artic
Xanagrams	Postern
Z80 Assembler, Disassembler and Editor	Arnor

sters would have helped.

Digger Barnes could have been good, but in its present form it cannot be recommended. True, the game is fast and furious, but compulsive it is not.

Brian Finnerty



Useful toolkit

Font 464, from Hisoft, is a utility to enable you to design, edit and save your own character set. These characters, or fonts, can be loaded and used quite easily from Basic.

The powerful editor has some unusual, but extremely useful commands. Despite this it's quite straightforward and simple to use.

Instead of working on one character at a time, a block of four is displayed on a 16 x 16 grid.

These four are chosen from the full character set. Any or all the block of four characters can be redefined and manipulated.

The results can be stored either temporarily or permanently and another block of four selected.

This feature is very useful for designing large characters which may be constructed from several ordinary characters.

All the commands you would expect to be present in a character designer have been included. You can invert, rotate and mirror at the touch of a key.

Additional commands, not normally found in a character designer, are scroll, add and delete.

Either single characters or the whole block can be scrolled in any direction using the cursor keys.

A row of pixels can be deleted and the rest moved up, or a blank row of pixels inserted at any position.

At the end of a session the complete character set can be saved.

Five fonts have been defined for you and follow the main program on the tape. These can be loaded into the editor or loaded and used by a Basic program.

There's an option to animate a block of three by three characters. This is rather spoilt by the fact that you can only design a two by two block.

You can always fill the other five character positions with spaces, though.

An added bonus is the two screen dump utilities on side B of the tape. One is for the Amstrad DMP1 and the other for Epson printers. The Epson dump includes shading for the colours.

To sum up, Hisoft Font 464 is useful if you need to design large numbers of characters. With its five pre-defined fonts and two effective screen dumps it's quite a useful toolkit.

Roland Waddilove

A body too many

PSS Software's **Frank N. Stein** is a game in the style of Manic Miner.

The idea is that you guide Professor Stein around the screen picking up pieces of the monster's body as you go.

Needless to say, there are all sorts of beasties and other hazards intent on stopping you from achieving this objective, so you have to carefully plan your route and make sure that you get all the bits in the correct order.

The screen display is quite good, with smooth animation and an option of colour or monochrome monitor. The sound is adequate and unobtrusive.

Getting the hang of the key response took me a few tries until I realised that jumping up and sliding down the various hazards is best done on the move.

I struggled manfully with the first screen and after many attempts and corresponding failures made it on to the second.

Then by pure chance it seems I was into the third screen. My luck then deserted me and pow! back to screen one again.

All I can tell you about the number of screens is that there are at least three. I'm sure there are many more if you have the staying power to find them.

As I said in the introduction, this game is in the style of Manic Miner. And frankly if, like me, you have tired of leaping around the screen collecting things in the guise of Miner Willy, I rather suspect that you'll tire quickly of it.

I know that my "game freak" other half certainly did.

The instructions are restricted to telling you how to load and get started. They also say that "the program will work with any Amstrad compatible joystick".

Unfortunately my joystick appears to be inadequate for the task and I had to use the keyboard, although I don't think that this was any great disadvantage.

This is a well-written game. If you have not tired of Manic Miner-type games it is an adequate alternative.

Allan Mitchell

Not lost for words

WHENEVER I purchase a new computer I always look around for some "sensible" item of software to help justify the purchase to my wife.

Inevitably I purchase a word processing package and have therefore used quite a few on both the Amstrad and BBC computers.

I have always ranked **Tasword 464,** from Tasman Software, as one of the better packages. But with the release of their new disc-based package they are now well ahead of the competition.

Essentially, Tasword 464-D is the same word processor as the tape-based program.

The program uses Mode 0 which is the 80-column mode.

This allows the user to format the text on the screen and be sure that it will appear with the same layout when printed.

The only drawback with using this high-resolution mode is the poor quality of the Amstrad monitors.

I have always had the feeling that the program was based upon that classic word processor of the business world, Wordstar – all of the commands used to manupulate the text being given by a combination of Shift or Control key plus one other.

The new program can now accommodate more than 20,000 characters of text in a single document, a 50 per cent increase.

This is all made possible by the use of disc overlays. Quite simply, this means that the different parts of the program are stored on disc. They are then automatically loaded and run when required.

The increase in space available to text is not the only enhancement.

Should 20,000 characters be insufficient to hold the piece of text that you are working on, then you can save the piece as several files.

It is possible to create a short file which contains the names of the files to be printed, called a "print file".

When Tasword is called upon to print this file it will load and print each of the files mentioned, in sequence.

Another new facility provided is the data merge option.

This allows the user to store data, such as names and addresses, in a Tasword file.

These names and addresses can then be merged into a standard letter or circular.

The original Tasword program was an excellent program for the hobbyist. This new disc version should satisfy the needs of even the most demanding of business users.

James Riddell

Take a hand

ALTHOUGH millions play contract bridge, there must be many people who give up because the scoring and bidding seem so complex.

If this applies to you, buy **Bridge Player** from CP Software – because it does all the scoring *plus* allowing you to rebid *and* replay a hand.

The reason the game is so baffling is that many new players understand the jargon and mechanics of whist (trump, ruff, finesse, singleton, void etc), but books rarely explain how bridge developed from Captain Hornblower's favourite game.

Quite simply, Harold Vanderbilt (circa 1926 on a Pacific voyage) invented a system whereby the points scored depended not just on the number of tricks taken, but also on the contract and the game situation.

Briefly you score most if you make your contracted tricks in no trumps, next in major suit (spades, hearts) and least in a minor suit (diamonds, clubs) *plus* you can be vulnerable (that is you have won one game out of the two required to win a rubber or set).

The program uses Acol-Stayman with Gerber (on request) so there is little excuse for missing the right contract.

However, actual play is much more difficult for it – I suspect it occasionally peeks at my hand for a finesse – for example, West leads a 2, table plays 3 and East plays a 5? to my singleton 4??.

It can blunder badly in ten-ace situations and has a hard time keeping cards to make me trump back in – situations well know to any whist player.

The important thing (for the learner) is that if you blunder you can replay the hand and thus practise the skill of entry to and exit from dummy, plus cheat on the finesses yourself – not useful or ethical but interesting.

Although written in Basic, the program plays a fair game at reasonable speed and will get the novice (with book) started.

Aleatoire

Lift a curse

Smuggler's Cove is an excellent graphical adventure from Camel Micros.

You play the part of an adventurer who has had a curse laid on him by a long-dead pirate. To lift the curse you must recover the pirate's treasure.

You start the adventure in an underground network of caves.

A quick exploration of your surroundings isn't very rewarding, though definitely worth a second look.

A trench prevents exploration on one side and a collapsed tunnel on the other.

Your only implements are a rusty lamp, a hammer and a monster bag of munchies tied to a lifebelt.

Your lamp is soon lit and after quite a lot of self-examination you will find the barrel.

You'll have a smashing time and, yes, I will tell you, the rope can be thrown to give access to the ladder. This proves to be one of the innovations of this adventure.

If you do something, such as throwing the rope, the picture changes to show you the rope in position.

There are only about 30 locations in this adventure, yet the lack of rooms seems to make the game harder instead of easier.

I haven't managed to finish it, but it is one adventure that I will keep going back to until I do finish it.

The graphics during the course of the game are excellent and easily the best I have yet seen in an adventure.

Overall, I cannot help but to recommend this adventure. The graphics and the problems are all excellent. Highly recommended.

Paul Gardner

Gatecrashers need luck

I OFTEN feel quite guilty reviewing a game that I cannot master, and **Gatecrasher,** from Exopal, is no exception. Mind you, the only games I keep playing are the ones I haven't mastered . . .

It's a game of skill and strategy – and quite a lot of luck.

You control the lateral movement of a man above the entrances to a series of underground interconnecting tunnels.

After studying the maze, you must select a particular chute down which to drop a barrel.

The idea is to send the barrel into one of nine empty boxes at the base of the screen.

Put one barrel in each box and you survive the screen.

Put a barrel in a box which is already full and you lose its contents.

Fail to put a barrel in each compartment and your game is over.

The direction in which the barrel rolls is governed by a series of gates.

Once a barrel is moving, it continues until one of these gates changes its direction. As it does so, the gate swivels ready to send the next barrel the opposite way.

If you decide that there is no path to the box of your choice, you have two options.

The first is to scroll the screen either up or down, and this has the effect of changing the layout of the maze.

You can scroll six times in any one direction before the layout reverts to its original pattern.

Your alternative is to cause an earthquake.

At the press of a key, the whole screen shakes and a random number of gates flip over.

However, this costs you one barrel, and, as you only have 20, and score a bonus relative to the number of barrels not used, this option should not be used except as a last resort.

There are seven screens which increase in difficulty and tested my strategy to its limits. However I still only managed to reach screen 4.

The graphics are exceptional, and the man's animation is extremely realistic. Sound is used to good effect and the pre-selected keys are quite acceptable, although there is a joystick option.

It won't take the country by storm, but if I'd had to buy it, I would have had my money's worth already.

Carole Whitehead

ACCESS
ORDERS
NOW TAKEN

Haystack Peripherals CPC464 Software Haystack Peripherals

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

	RRP	OUR PRICE
ADDICTIVE		
Football Manager	7.95	6.95
ADVENTURE INTERNATIONAL		
Gremlins	8.95	7.95
Robin of Sherwood	9.95	8.95
ALLIGATA		
Blagger	8.95	7.95
Defend or Die	7.95	6.95
ACTIVISION		
Ghostbusters	10.99	9.99
Master of the Lamps	9.99	8.99
AMSOFT		
Airwolf	8.95	7.95
Alien	8.95	7.95
American Football	9.95	8.95
Answord (Advanced)	19.95	17.95
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
Grand Prix Rally II	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
Masterfile	24.95	22.95
Pitman Typing Tutor	9.25	8.25
Project Planner	24.95	22.95
Screen Designer	£14.95	12.95
Spreadsheet *	19.95	17.95
Starwatcher	17.45	15.45
Stock Aid	29.95	26.95
Transact	29.95	26.95
ANIROG		
Jump Jet	9.95	8.95
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
CCS		
Arnhem	9.95	8.95
Warzone	6.95	5.95
CDS		
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
3D Voice Chess	12.95	11.95
CRL		
Amstrad Artist	9.95	8.95
Handicap Golf	6.95	5.95
Rocky Horror Show	9.95	8.95
Smuggler's Cove	5.95	5.25
Test Match	6.95	5.95
CABLE		
Digger Barnes	7.95	6.95
CAMEL		
FlexiFrend	7.50	6.50
Grasp	8.50	7.50
DATABASE		
Mini Office	5.95	5.25
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

	RRP	OUR PRICE
DOMARK		
A View to a Kill	10.99	9.99
Codename Mat II*	8.95	7.95
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	8.85
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		
The Magician's Ball	7.95	6.95
HEWSON		
Dragontorc	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
KOSMOS		
French Mistress 'A'	8.95	7.95
KUMA		
Artwork	9.95	8.95
Database	14.95	12.95
Forth	19.95	17.95
Fruity Frank	6.95	5.95
Galaxia	5.95	5.25
Holdfast	5.95	5.25
Home Budget	19.95	17.95
Music Composer	9.95	8.95
Star Avenger	6.95	5.95
LEISURE GENIUS		
Scrabble	9.95	8.95
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
Mordon's Quest	6.95	5.95
Starion	9.95	8.95
Way of the Exploding Fist	9.95	8.95
MICROGEN		
Everyone's a Wally	9.95	8.95
Herberts Dummy Run*	9.95	8.05
Pyjamarama	8.95	7.95

	RRP	OUR PRICE
MICROPOWER		
Ghouls	6.95	6.15
Killer Gorilla/Gauntlet	9.95	8.95
MIRRORSOFT		
Boulderdash	9.95	8.95
Caesar's Travels*	8.95	7.95
First Steps/Mr. Men	8.95	7.95
Here + There/Mr. Men	8.95	7.95
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
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
SYSTEM 3		
International Karate*	6.50	5.50
TASKSET		
Souls of Darkon*	8.90	7.90
Superpipeline 2	8.95	7.95
TASMAN		
TASCOPY	9.90	8.90
TASPRINT	9.90	8.90
TASWORD	19.95	17.95
THE EDGE		
Brian Bloodaxe	8.95	7.95
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
Raid	9.95	8.95
VIRGIN		
Sorcery	8.95	7.95
VORTEX		
Android One	7.95	6.95
Android Two	7.95	6.95
Tornado Low Level	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 (Please write card no. on reverse)/PO's to:

HAYSTACK PERIPHERALS
8 Midgrove, Delph, Oldham OL3 5EJ
Tel: Saddleworth (04577) 5229

NAME ....

ADDRESS ....

....

Phone No. ....

Access No. ....

Still carrying the flag

Part IX of MIKE BIBBY's introduction to machine code

LAST month we spent a lot of time looking at the Carry flag – particularly how it could be used together with CP to create loops, filter inputs and so on.

However, the Carry flag isn't the only flag on the Z80: There are six altogether. The one we're interested in at the moment, though, is the Zero flag.

Actually, the Zero flag's name virtually tells you what it does – the Zero flag is set when the "answer" to some machine code operation comes to zero.

For example:

```
LD A,&FF
ADD A,&01
RET
```

Program I

will not only leave 0 in the A register, it will also set the Zero flag to tell us the result of the operation was zero.

Similarly:

```
LD A,&DF
SUB &DF
RET
```

Program II

will both leave zero in A and set the Zero flag.

So, if an ADD or SUB leaves zero in the A register, the Zero flag is set. If the result is non-zero, the Zero flag is cleared.

If you think about it, it's rather topsy-turvy: When the result *is* zero you set the Zero flag – to one. And when the result *is not* zero, you clear the Zero flag – to zero. One when it's zero, zero when it's not!

Don't worry about it – the actual values of flags don't really concern us. All we need to know is what conditions set and clear the flag. In this case the Zero flag is set when the result of an operation is zero, and cleared when not.

Not every instruction affects the Zero flag, but you'll soon get used to the ones that do – they're fairly obvious.

You'll probably remember that we had a pair of jump instructions based around the Carry flag – JP C and JP NC.

Well, we have a corresponding pair of jump instructions involving the Zero flag.

```
     JP Z     ( opcode &CA )
```

performs the jump if the Zero flag has been set, and

```
     JP NZ     ( opcode &C2 )
```

takes the jump if the Zero flag is not set – that is, it's clear.

Take a look at Program III. It shouldn't present you with any problems, we've met it twice before: It was our first loop.

```
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 III

You'll remember that we keep on incrementing the A register and jumping back to print out the corresponding Ascii code until we go round the clock – and in so doing set the Carry flag.

However, at the same time, the Zero flag will be set since the last ADD A,1 adds 1 to the 255 (&FF) in the A register, giving us zero.

This means we could rewrite Program III with a JP NZ, &3002 in place of the JP NC, &3002 – and this is what we do in Program IV.

```
address  hex code  anemonics
 3000    3E 20     LD A,&20
 3002    CD 5A BB  CALL CharOut
 3005    C6 01     ADD A,1
 3007    C2 02 30  JP NZ,&3002
 300A    C9        RET
```

Program IV

Here we keep on performing the loop until the Zero flag is set by the ADD A,1 when A contains &FF. We then drop through the loop and RET.

Don't get the idea that the Carry and Zero flags are linked in any way. In the above example it's just coincidence that both are set at the same time – it isn't always so.

To prove it, take a look at Program V. We're using INC A (&3C) to increase A by 1 instead of the ADD A,1. Otherwise it's identical to Program IV, but one byte shorter.

```
address  hex code  anemonics
 3000    3E 20     LD A,&20
 3002    CD 5A BB  CALL CharOut
 3005    3C        INC A
 3006    C2 02 30  JP NZ, &3002
 3009    C9        RET
```

Program V

As you'll see, it works fine.

However, if, as in Program VI, we replace the JP NZ with JP NC – which we might do if we thought the Zero and Carry flags were linked – disaster occurs.

The point is that INC A sets or clears the Zero flag depending on the result of increasing A by one. However, INC A doesn't affect the Carry flag at all.

This means the Carry flag is not set when A is increased from 255 to 0. The loop therefore keeps going, effectively printing out CHR$(1), CHR$(2) and so on.

However the Ascii codes under 32 (&20) are control codes, which, as you'll see from the User Guide, have some pretty powerful effects.

```
address  hex code  mnemonics
 3000    3E 20     LD A,&20
 3002    CD 5A BB  CALL CharOut
 3005    3C        INC A
 3006    D2 02 30  JP NC, &3002
 3009    C9        RET
```

Program VI

In practice, your screen will lock up: your micro has crashed. Don't worry, you haven't hurt it – it's just that you'll have to reset the machine to regain control. Unfortunately, in doing so, you'll lose whatever program is in memory – in my case Hexer, my hexadecimal loader.

My advice is that you don't run Program VI until the end of the session, when you're ready to switch off anyway . . . It certainly demonstrates the fact that the Carry and Zero flags are independent – in a way you're not likely to forget.

Actually, despite the obvious error in Program VI, there's a subtler one. We've assumed that the Carry flag is clear when we enter the program.

Think about it. If, when we run the program – either from Hexer or directly by CALL &3000 – the Carry flag were already set, it would remain so throughout the first three instructions.

As we've said, LD instructions don't affect the Carry flag, so LD A,&20 will leave Carry set, as will the call to CharOut. And, as we've already mentioned, INC A doesn't affect it.

This means that when we reach the JP NC, we don't take the jump, since Carry has been set from the beginning. We would then exit our routine via the final RET, having just printed out a space (Ascii &20).

In practice this doesn't happen. When we enter the routine, Carry is clear, and remains so since none of the instructions affects it. Hence the JP NC is always taken and we crash.

Still, in machine code, it's wrong to assume what the state of a particular flag will be. Doing so often causes elusive bugs in programs.

So in Program VI, we shouldn't just hope that Carry will be clear, we should force the issue. It would be nice if we could do this with a single Z80 instruction. Unfortunately, no such instruction exists.

However we can clear Carry with:

```
ADD A,0    ( C6 00 )
```

After all, you can't go round the clock by adding zero to anything, can you? And ADD *does* affect the Carry flag, so in this case Carry must be cleared. The added bonus (no pun intended) is that the contents of the A register remain unchanged – think about it.

Thus the effect of ADD A,0 is to clear the Carry flag for us, without affecting the contents of the A register.

Right, we know how to clear the Carry flag when we want to – so how do we set it?

Well, this time there is an instruction that fits the bill – SCF (opcode &37). SCF stands for Set Carry Flag.

Program VII shows a trivial example of the use of these ideas.

```
address  hex code  mnemonics
 3000    3E 07     LD A,7
 3002    C6 00     ADD A,0
 3004    DA 00 30  JP C,&3000
 3007    37        SCF
 3008    D2 00 30  JP NC,&3000
 300B    CD 5A BB  CALL CharOut
 300E    C9        RET
```

Program VII

As you can see, there are two conditional jumps which take you back to the beginning of the program. I've used ADD A,0 before the first and SCF before the second to rig things so that the jumps are not taken.

We go straight through to CALL CharOut, which gives us a bleep. If we get any of our conditional jumps the wrong way round, or if we set when we should clear, we'll get stuck in an infinite loop.

Program VIII shows another rather contrived example. If you follow through its rather tortuous path you'll see that, because of the way we've manipulated the Carry flag, nothing gets printed out.

```
address  hex code  mnemonics
 3000    3E 2A     LD A,&2A
 3002    C6 00     ADD A,0
 3004    D2 0A 30  JP NC, &300A
 3007    CD 5A BB  CALL CharOut
 300A    37        SCF
 300B    DA 11 30  JP C,&3011
 300E    CD 5A BB  CALL CharOut
 3011    C9        RET
```

Program VIII

Can you alter things so that you do indeed get two asterisks (Ascii code &2A) appearing on the screen? Hint: Have a good look at the conditions attached to those JP instructions.

Right, back to the Zero flag. Last month we saw that the CP n instruction, while not altering the A register, set and cleared the Carry flag as if a SUB n had been performed:

Carry was set if A < n

Carry was clear if A >= n

You won't be surprised to learn that CP n also affects the Zero flag as well:

If the byte in the A register is identical to n, the zero flag is set.

If the two bytes differ, the zero flag is cleared.

This is only what you'd expect. If the two bytes are the same, when the CP n does its simulated A – n, the answer must be zero, so the Zero flag's set.

If, on the other hand, the bytes vary, the answer is non-zero and the flag is cleared.

We can use this to "demand" that a certain character be input from the keyboard, as in Program IX, where we wait for an asterisk:

```
address  hex code  mnemonics
 3000    CD 18 BB  CALL CharIn
 3003    FE 2A     CP &2A
 3005    C2 00 30  JP NZ,&3000
 300B    CD 5A BB  CALL CharOut
 300B    C9        RET
```

Program IX

The program works by comparing the input character with the Ascii for an asterisk. If it is an asterisk, the zero flag is set and, ignoring the jump, we print it out and return.

If it isn't an asterisk, the codes will differ and the Zero flag will be cleared. We then jump back to the beginning, to get another character from the keyboard.

So far, when using CP, we've compared the A register with a specific number. We can, however, compare the A register with any other of our single registers, as Table I shows. The notation for this form of instruction is CP r, where r is an eight bit register.

CP B, for example, compares the byte in the A register with that in the B by doing a dummy A-B and setting the flags accordingly – the contents of both registers remain unchanged. Note, the register specified is subtracted from the A register.

Again, if A is equal to or greater than B, Carry is clear. If A is less than B, Carry is set. Also, if the bytes in A and B are equal, the Zero flag is set, otherwise it's cleared.

We put this ability to compare our eight bit registers to use in Program X. Here our task is to print out a fixed number of asterisks – eight in this case.

```
address  hex code  anemonics
 3000    06 00     LD B,0
 3002    3E 2A     LD A,&2A
 3004    CD 5A BB  CALL CharOut
 3007    04        INC B
 3008    3E 08     LD A,&08
 300A    BB        CP B
 300B    C2 02 30  JP NZ,&3002
 300E    C9        RET
```

Program X

We use the B register as a counter. Initially we set it to zero with LD B,0 and then print out an asterisk by loading A with &2A and calling CharOut. We next increase B, thus keeping track of the number of asterisks printed. We then load A with 8, the number of asterisks, then compare the B register with A.

If the two registers aren't equal – that is, if the Zero flag is not set – we haven't printed eight asterisks so we jump back to &3002, reloading A with &2A, calling CharOut and so on.

If, on the other hand, we've reached our limit, the Zero flag is set and the jump isn't taken, so we simply return.

```
address  hex code  anemonics
 3000    06 08     LD B,8
 3002    3E 2A     LD A,&2A
 3004    CD 5A BB  CALL CharOut
 3007    05        DEC B
 3008    C2 04 30  JP NZ,&3004
 300B    C9        RET
```

Program XI

Actually, the program is rather tortuous. There are more efficient ways to print out a number of asterisks. I just wanted to introduce the CP r instruction. Program XI shows an alternative way of doing it. The trick is to count down from the number you want by using DEC B.

When we reach zero, the Zero flag is set, otherwise we jump back and print another asterisk. In effect, B acts as a primitive loop variable or counter.

You can use this idea to create nested loops – using the B register for the inner loop and another for the outer loop.

Program XII uses this idea to print out a triangle of asterisks:

```
address  hex code  anemonics
 3000    0E 08     LD C,&08
 3002    41        LD B,C
 3003    3E 2A     LD A,&2A
 3005    CD 5A BB  CALL CharOut
 3008    05        DEC B
 3009    C2 05 30  JP NZ,&3005
 300C    3E 0A     LD A,&0A
 300E    CD 5A BB  CALL CharOut
 3011    3E 0D     LD A,&0D
 3013    CD BB 5A  CALL CharOut
 3016    0D        DEC C
 3017    C2 02 30  JP NZ,&3002
 301A    C9        RET
```

Program XII

We're going to have eight lines, eight asterisks in the first, seven in the second and so on. So each time round the outer loop we'll print a line of asterisks – the inner loop printing out the required number each time.

We use C to count the number of lines – hence the initial LD C,&08 – and B the number of asterisks. When you think about it, since we're using it as a counter, C goes down by one each time round the outer loop which is also what we want to happen to B. Our second instruction, the beginning of the outer loop, is therefore LD B,C.

If you look towards the end of the program, you'll see the end of the first loop:

```
3016    0D        DEC C
3017    C2 02 30  JP NZ,&3002
```

To return to the start of the outer loop, after LD B,C we load the A register with the Ascii for asterisk (LD A,&2A). We then arrive at the inner loop:

```
3005    CD 5A BB  CALL CharOut
3008    05        DEC B
3009    C2 05 30  JP NZ,&3005
```

As you'll soon see, this prints out "B asterisks". Now the value of B goes down by one each time through the outer loop since we get it via the C register (LD B,C) and C is decreased by the DEC C at the end of the outer loop each time. This means that each line has one less asterisk than the previous line – giving us our triangle effect.

When C gets to zero – after we've printed eight lines – the program drops through the JP NZ,&3002 of the outer loop and RETs. Incidentally the lines between the JP NZs of the inner and outer loops,

```
300C    3E 0A     LD A,&0A
300E    CD 5A BB  CALL CharOut
3011    3E 0D     LD A,&0D
3013    CD BB 5A  CALL CharOut
```

are just to print out a line feed followed by a carriage return, ensuring our lines are separate. Try leaving either or both out and see what happens. (Remember that your jumps will have changed.)

Right, that's plenty for one month. I'll leave you with two things to ponder, though.

If we're simply obtaining the value of B from C, why bother with C at all? Actually, that's easy, but it's the sort of daft thinking that can occur to you after your mind's been numbed by several hours of machine code programming.

My final poser is a little more demanding. Try turning the triangle upside down, with one asterisk at the top, and eight at the bottom.

CP A	BF
CP B	BB
CP C	B9
CP D	BA
CP E	BB
CP H	BC
CP L	BD

Table I: CP r

pride utilities LTD.

Some programs are sold on the understanding that they must not be used to infringe copyright.

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, 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
Cassette Headerless Backup

Achieve the ultimate in **tape to tape backup** with this no nonsense program which handles practically everything, even headerless, usually in one go. If you require to load and list software etc. then you require SYCLONE, the versatile tape to tape utility. If all you need are backups 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. ***"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

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

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
FREE with every order for a cassette 50
Worth over £10.00
EXPRESS DELIVERY- ORDER NOW! and get your FREE calculator watch with every 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. Evasive Action
12. Noughts and Crosses
13. King's Orb
14. Play Your Cards Right
15. Creepy Crawley
16. Hangman
17. Old Bones
18. Thin Ice
19. 3D Maze
20. Colony 9
21. Force Field
22. Nim
23. Tunnel Escape
24. Barrel Jump
25. Tracking
26. Rally 3000
27. Sitting Targ
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

Names and games may vary for each type of computer.

Name ______________________
Address ______________________
____
______________ Post Code ______________
Country ______________________ CA 10/85

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

VISA 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*

Stereo makes your Amstrad twice as good

By GABRIEL JACOBS

THERE'S no reason in principle why the Amstrad should not be connected to a conventional hi-fi stereo system. But for most people this is going to involve constantly carrying pieces of equipment around the house and always having to reposition the micro between the two speakers.

So if you've installed your machine in some remote corner to keep it away from uninvited fingers – or yourself away from the rest of mankind – and if you also want to take advantage of the full potential of its stereo capability, then the C-Lect 5000 kit of stereo amplifier, speakers and headphones, designed to sit permanently on your desk, may be just what you need.

The amplifier, purpose-built for the Amstrad, is housed in a sturdy black box measuring 7 by 5 by $3\frac{1}{2}$ inches. It plugs into the stereo output port, taking its power from the mains via a transformer supplied.

It has standard 3.5mm jack sockets for input and output, two separate volume controls, a selector switch for speakers or headphones and comes with all necessary leads.

It will drive any speakers of at least 4 ohms impedance, but the full C-Lect kit includes a pair of Japanese 3 inch double cone, air suspension units. These are capable of handling up to 30 watts each, and so I was surprised to find that the amplifier delivers only $1\frac{1}{2}$ watts per channel – plenty to spare, to say the least.

In practice, however, the amplifier and speakers proved to be well matched. Volume levels will certainly be adequate for all except the very hard-of-hearing, or those determined to damage their eardrums.

Stereo separation is excellent, and the frequency response is fairly flat, giving uniformly good sound across the range. Bass response in particular is impressive for such a small system.

The relatively cheap Korean headphones supplied with it, however, give rather poor results in the bass.

Once you have set up the C-Lect system, or any stereo amplification system, the Amstrad's own unremarkable mono speaker is bypassed, and any sound generated is diverted to the stereo port.

This means that currently available software written with stereo sound, such as Protek's Hunter/Killer, will be automatically implemented as its programmers intended it to be. And if you haven't yet heard some of your games with good stereo sound, you've a real surprise waiting for you.

A tape of a gavotte by Bach, which was sent with the review kit for demo purposes, gave a fair idea of what can be achieved musically with the Amstrad's three sound channels, given the proper equipment.

But we're still waiting for the appearance of a good range of music software comparable with that available for older-established micros.

When Island Logic, for example, produce a version of their Music System for the Amstrad, the stereo results, output through an external amplifier, ought to put the competition in the shade. In the meantime, the C-Lect system only serves to emphasise the present lack of this kind of software.

At the time of writing I know of only one other Amstrad-dedicated stereo amplifier actually available, the one which comes with Dk'tronic's Speech Synthesiser.

This will give you the same basic capabilities as the C-Lect 5000 – and, of course, you get a versatile speech synthesiser thrown in – but it will not give you the same quality of sound.

So if you're less interested in digital speech than in good quality digital music, not to mention good-quality crashes, bleeps and sirens, then the C-Lect system is worth buying.

Cost is £35 for the full kit, and £18.50 for the amplifier and headphones only. Not cheap, but affordable and quite good value for money.

"YOUR PROGRAMS ALWAYS REACH ME BEFORE ONES ORDERED LOCALLY"

(I. Sinclair, N.S.W., Australia)

WE ALWAYS TRY TO SEND YOUR PROGRAMS ON THE SAME DAY WE GET YOUR ORDER, THAT'S WHY IT'S CALLED SPEEDYSOFT! All programs normally in stock before we advertise them! FOREIGN ORDERS WELCOME, satisfied customers in 63 countries for 3 years. UK prices include VAT: export prices are the same, plus p&p, because of the extra work involved. Pay by sterling cheque, bank draft or postal order. It's faster to order by phone from any country with your VISA, EUROCARD, MASTERCARD, or ACCESS. Call 01-846 9353, 24hrs, 7 days a week. BIG FREE ILLUSTRATED CATALOGUE WITH EVERY ORDER WITH DETAILED REVIEWS, LOAD TIMES, ETC. Send £1.00 (refundable) for catalogue only.

AMSTRAD 6128 OWNERS

Welcome to Speedysoft. Programs listed here that will run on your machine are indicated. We shall be looking out for the first programs specially for your machine. Meanwhile send for our catalogue and find out just how wide your choice is!

JUMP JET

"Combines not only the challenge of coping with a flight simulation, but also the thrill of combat ... it's fairly easy to get airborne, but the rest of the mission is very challenging." (C&VG) This is your chance what even very few pilots do, fly your own jump jet in combat. Good tunes, great graphics, intelligent use of split-screens. Sticks OK. (Anirog) **AMSTRAD 464/664/6128 Cassette £9.95 Disk £13.95**

'C' COMPILER

"The manual claims that 'this is one of the most important products yet released for the Amstrad.' They may not be far wrong. The chance to gain experience of the 'C' Language in your own home is something no aspiring 'serious' programmer should miss. Of all the languages I have seen on the Amstrad, none has impressed me as much as this." (PCW) This program produces very fast machine-code by a one-step operation. You can sell your own machine-code software without royalties! Over 120 functions. 148-page manual. If you can write BASIC programs, you can run them a lot faster with this. (HiSoft) **AMSTRAD 464/664/6128 2 Cassettes £34.95 Disk £39.95**

RALLY 2

DESIGN AND SAVE YOUR OWN GRAND PRIX CIRCUITS! *"AT LAST! ... this game is very playable and equally addictive. Rally 2 really scores when compared with many other games ... editing is done purely on the joystick and could not be simpler." (AmUsr)* Now you can race your friends around a circuit you designed. The initial circuit has ten stages to complete in order. Different weather and scenery and plenty of competitors. Then you start to build your own races! (Amsoft) **AMSTRAD 464/664/6128 Cassette £ 8.95 Disk £12.95**

BRAXX BLUFF

"Graphics, sound and player facilities are all of extra quality here. It's really quite outstanding, with an intriguing plot ... the graphics are superb ...the 3D runway is quite something." (HCW) Spectacular cockpit views, great music. (Amsoft/Micromega) **AMSTRAD 464/66/6128 Cassette £8.95 Disk £12.95**

RED MOON

"The appearance of a new program from Level 9 is a flag day for all aspiring adventurers!" (PCW) 200 locations, plus pictures. You must use many weapons and 12 spells to collect the 9 treasures and find the Red Moon crystal. Cluesheet if you get stuck! (Level 9) **AMSTRAD 464/664/6128 Cassette £6.95**

MASTERCALC

"The easiest program of all to use ... the manual is by far the best." (PCW) The perfect Amstrad spreadsheet. Up to 3,000 cells. 0-7 decimal places. Variable column width. Insert/delete column/row. GRAPHIC BAR CHARTS. Hi-res screen copy with Epson compatibles. (Amsoft) **AMSTRAD 464/664/6128 Cassette £24.95 Disk £29.95**

JET SET WILLY 2

"The new rooms are, in many cases, inventive and fiendish ... the deluxe, souped-up version of the original JSW, addicts should be delighted." (PCW) Over 100 rooms: the perfected version of one of the greatest games of all. (Software Projects) **AMSTRAD 464 Cassette £8.95**

Frank Bruno's BOXING

"In the battle of the boxers, it's a knockout to Frank ...stands head, shoulders and boxing gloves above the competition." (YrSp) Meet 8 successive heavyweights (each fighter loads separately as you qualify to fight him. Each fights differently. Sticks Ok. (Elite) **Amstrad 464 Cassette £8.95**

EXPLODING FIST

"Karate should be a sure-fire winner when portrayed as realistically as in this superb simulation. The movement of the fighters is very realistic ... virtually beyond criticism, technically brilliant and visually absorbing." (PersCompWorld) "Each animated figure is a joy to see. Movement is fast, smooth and realistic ... just wonderful." (C&VG) For 1 or 2 players as they progress from novice to Tenth Dan. Over 700 sprites used for each fighter! 18 manoeuvres to learn. Instructions in French and German too! Sticks OK. (Melbourne House) **AMSTRAD 464 Cassette 9.95**

RED ARROWS

"ONE OF THE BEST FLIGHT SIMULATORS TO BE RELEASED IN A LONG TIME." (HCW) "IT'S TREMENDOUSLY DIFFICULT TO DO AND ONLY ABSOLUTE ADDICTS SHOULD TRY IT." (PCW) Faithful 3D graphic reconstruction of the Red Arrows aerobatics as you train and eventually join the display team. Monthly competiton to watch a real flypast ends 31/1/86. Sticks OK. (Database) **AMSTRAD 464 Cassette £8.95 Disk £12.95**

DUN DARACH

"A superb game ... definitely a strong contender for game of the year ... the animation is fantastic, one of the best-looking games around." (Crash) Direct the young Cuchulainn around the mediaeval city of Dun Darach to seek Loeg the charioteer. Encounter 11 brilliantly animated characters. Demo. 9 command keys. Great graphic adventure. No Stick. (Gargoyle) **AMSTRAD464. Cassette £9.95**

BEACH HEAD

"Spectacular sound and graphics bring amazing realism ... the most realistic shoot-em-up I've ever come across." (PCG) "A multi-screen arcade game that is very close to being perfect ... graphically, it is astonishing." (PCW) A whole war on one tape! Demo. 6 different battles. 3 skill levels. SAVE Hall of Fame. Stick OK. (US Gold) **AMSTRAD 464/664/6128. Cassette £9.95**

FIGHTER PILOT

"THE FLYING IS THE BEST I'VE COME ACROSS ... it wins too on graphics and instrumentation." (PCG) "The game with the most playability and the longest-lasting appeal I have played on the Amstrad . . . months of flying here." (CVG) "A superb flight simulation with full instrumentation and marvellous moving horizon cockpit views." (CompCh) Stick OK. (Digital) **AMSTRAD 464/664/6128. CASSETTE £8.95**

The CODE MACHINE

"This is a first-class product ... In a straight comparison with the Amsoft DEVPAC program The Code Machine is easier to use and has extra features." (HCW) The editor and the assembler can be used separately, or both loaded into memory together, giving direct access to each other. Both programs are relocatable to allow external ROMS or RSXs to reserve memory space. You can back up the Program to DISK or TAPE. If you are writing machine-code, you need this. It's simply the best. No Stick. (Picturesque) **AMSTRAD 464/664. CASSETTE £19.95**

VISA
ACCESS

SPEEDYSOFT

01-846 9353 (24 HRS)

EUROCARD
MASTERCARD

POST TO: SPEEDYSOFT (CWA1)
37 CHURCH ROAD, LONDON SW13 9HQ, ENGLAND.
For CATALOGUE ONLY, send £1 cash. Refunded with your first order.
I own an AMSTRAD . . . . . . I enclose a cheque/PO payable to Speedysoft
OR charge my VISA/ACCESS/EUROCARD/MASTERCARD

No. □□□□ □□□□ □□□□ □□□□

Signature: ______________ Expiry Date ______________
Please write clearly. If we can't read it, you won't get it.
Name: ______________
Address: ______________
____
______________ Postcode: ______________
PHONE NO: if any, in case of query ______________

Program Name	Cass/Disk	Price
Postage & Packing	UK add 75p per order	
	Europe ADD £1.00 per program	
	Outside Europe ADD £1.50 per program	
	Total Order	

WIN the fabulous tale of...

"Well Mr Baggins, all is ready for adventure and I must say things are looking very hopeful. Shall we be off then?"

SO begins what has become one of the all-time classic games for micros – The Hobbit. What follows is a superb recreation of a quest through Tolkien's fantasy world of Middle Earth – a world peopled by trolls, elves, dwarves and many other mysterious creatures.

Based on the plot of Tolkien's famous book, The Hobbit, the task is to steal treasure from the evil dragon Smaug, with you in the role of Bilbo Baggins, gentleman hobbit.

Your journey through Middle Earth will be anything but easy, though. The characters you meet aren't always friendly, to say the least – and there's no guarantee your faithful companions are!

In this unique adventure each character makes his own decisions, so that the game is slightly different every time it is played.

The exciting news is that The Hobbit is now available for the Amstrad. So for the first time you can explore the magical world of Middle Earth.

To celebrate this fact, we are giving away 50 cassette versions of The Hobbit as prizes for this month's contest.

The competition shouldn't give you any problems at all – it's simply a word square. Hidden in it are the names of 18 characters, creatures and locations mentioned in the book.

All you have to do is to ring them on the coupon below, and send them to us not later than October 30, 1985. The first 50 correct solutions out of the bag will be the winners.

Hobbit Contest entry form

Name______________________________

Address____________________________

____

____

____

B	E	E	L	D	F	L	E	R	B	Y	N	E	D
A	I	R	B	I	R	D	O	O	W	K	R	I	M
G	N	N	L	G	E	L	L	I	R	R	O	P	A
G	N	I	W	H	T	O	R	A	G	S	E	P	S
I	D	R	O	L	C	G	U	N	D	A	B	A	D
N	U	U	F	T	D	K	C	A	R	R	O	C	K
S	M	D	I	R	A	V	E	N	H	I	L	L	T
R	H	U	X	N	D	N	O	R	L	E	E	V	H
E	K	B	L	B	I	L	B	O	E	W	S	D	O
X	L	S	S	L	A	T	T	E	R	C	O	P	R
G	F	A	L	R	O	G	U	A	M	S	O	G	I
F	L	A	D	N	A	G	T	D	W	A	L	I	N

Send to: Hobbit Contest, Computing with the Amstrad,
Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Be a race ace . . . with ROLAND WADDILOVE's

ALL the thrills, the spills and glamour of the racetrack in your own home. The famous Grand Prix has been transferred to the Amstrad to test your skill and daring.

And you'll need nimble fingers and fast reactions if you're to win the race.

After a poor start your turbo charged formula three racer is rapidly catching up with the back markers. You're on the long straight now so put your foot down and pass as many as you can while you have the chance.

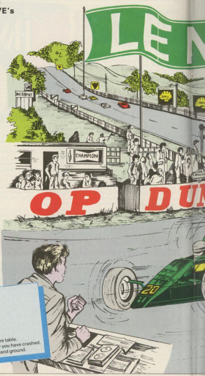
It's easy at first, but as you progress through the field it becomes more difficult. The leaders will try to block your path by weaving and dodging, so watch out.

Le Mans is a gruelling 24 hour race pushing both car and driver to the limit. Daylight should pass fairly smoothly, but it becomes more hairy as you drive into the night.

During the following day you may experience car trouble as the strain begins to show. Keep your fingers crossed!

This is a fast, 3D race game with multi-colour graphics and smooth scrolling landscape. It contains brief instructions and you can use either joystick or the keyboard. There is a table of best drivers and a suitably high speed tune to accompany it and the instructions.

The program is well structured and each subroutine has been given a title to aid debugging. Be careful with the data statements, there are rather a lot – make sure that you've entered each exactly as listed.



VARIABLES

a,b	Coordinates of car 1.
c,d	Coordinates of car 2.
x	Your x coordinate.
dif	Difficulty
score	Score.
lives	Lives left.
gear	What gear you're in.
name$(10)	Names in the high score table.
ok	Flag showing whether you have crashed.
f,g,h,	Flags for moving cars and ground.

```
10 REM Le Mans
20 REM By R.A.Waddilove
30 REM(c)Computing With The Amstrad
40 GOSUB 1290:REM Initialise
50 GOSUB 2020:REM Instructions
60 WHILE dead.interested
70 GOSUB 2300:REM Starting grid
80 GOSUB 920:REM Screen
90 AFTER 2000,0 GOSUB 600
100 WHILE lives
110 WHILE ok
120 GOSUB 320:REM Ground
130 GOSUB 260:REM Your car
140 GOSUB 360:REM Car 1
150 GOSUB 450:REM Car 2
160 GOSUB 540:REM Gears
170 GOSUB 320:REM Ground
180 GOSUB 260:REM Your car
190 GOSUB 540:REM Gears
200 WEND
210 GOSUB 750:REM Crashed
220 WEND
230 IF score>hiscore(10) THEN GOSUB 2
420:REM Hi score
240 WEND
250 END
260 REM --------- Move car ---------
270 SOUND 129,960,500,8+gear,0,1:SOUN
D 130,1205,500,8+gear,0,1:ax=x+(INKEY
(left)>-1)-(INKEY(right)>-1):IF ax<4
THEN ax=4 ELSE IF ax>16 THEN ax=16
280 IF RND*dif<0.6 THEN RETURN ELSE P
OKE &A034,16:CALL &BD19:CALL &A000,x,
22,ax,22,&A100:x=ax
290 IF b>21 THEN IF ABS(x-a)<3 THEN o
k=NOT ok:RETURN
300 IF d>21 THEN IF ABS(x-c)<3 THEN o
k=NOT ok
310 RETURN
320 REM --------- Ground ---------
330 g=(g+1)MOD 3:IF g=0 THEN CALL &BD
19:INK 0,9:INK 1,19:INK 2,21:INK 4,13
:INK 5,13:INK 6,26:RETURN
340 IF g=1 THEN CALL &BD19:INK 0,21:I
NK 1,9:INK 2,19:INK 4,26:INK 5,13:INK
 6,13:RETURN
350 CALL &BD19:INK 0,19:INK 1,21:INK
2,9:INK 4,13:INK 5,26:INK 6,13:RETURN
360 REM ----- Move car 1 ----------
370 IF b=23 THEN LOCATE a,b:PRINT "
":a=0:b=0:score=score+gear:LOCATE 8,2
5:PRINT MID$(STR$(score),2):RETURN
380 f=(f+1)MOD (dif+1):POKE &A034,8
390 IF a=0 THEN a=10:b=12:CALL &A000,
a,b,a,b,gp(0):SOUND 132,900,500,10,0,
1:RETURN
400 IF f=0 THEN aa=a+(x<a)-(x>a) ELSE
```

```
 aa=a-(b>20)*(x<a)+(b>20)*(x>a)
410 size=(b-12)\4
420 CALL &A000,a,b,aa,b+1,gp(size):SO
UND 132,900,500,10+size*2,0,1
430 a=aa:b=b+1
440 RETURN
450 REM ----- Move car 2 ----------
460 IF d=23 THEN LOCATE c,d:PRINT "
":c=0:d=0:score=score+gear:LOCATE 8,2
5:PRINT MID$(STR$(score),2):RETURN
470 h=(h+1)MOD dif:POKE &A034,8
480 IF c=0 THEN c=10:d=12:CALL &A000,
c,d,c,d,gp(0):SOUND 132,900,500,10,0,
1:RETURN
490 IF h=0 THEN cc=c+(x<c)-(x>c) ELSE
 cc=c-(d>20)*(x<c)+(d>20)*(x>c)
500 size=(d-12)\4
510 CALL &A000,c,d,cc,d+1,gp(size):SO
UND 132,900,500,10+size*2,0,1
520 c=cc:d=d+1
530 RETURN
540 REM ------ Change gear ---------
550 IF gear<5 THEN gear=gear-(INKEY(u
p)>-1)
560 IF gear>1 THEN gear=gear+(INKEY(d
own)>-1)
570 LOCATE 19,25:PRINT MID$(STR$(gear
),2,1)
580 FOR i=1 TO 550-gear*100:NEXT
590 RETURN
600 REM -------- night ------------
610 INK 15,13:INK 3,11:dif=dif-1
620 AFTER 2000,0 GOSUB 640
630 RETURN
640 REM --------------------------
650 INK 15,0:INK 3,1:dif=dif-1
660 AFTER 2000,0 GOSUB 680
670 RETURN
680 REM --------------------------
690 INK 15,13:INK 3,11:dif=dif-1
700 AFTER 2000,0 GOSUB 720
710 RETURN
720 REM --------------------------
730 INK 15,11:INK 3,20:dif=dif-1
740 RETURN
750 REM --------- Crashed --------
760 PRINT CHR$(30);CHR$(22);CHR$(1)
770 FOR i=0 TO 30
780 SOUND 129,10+INT(RND*1000),10,15,
0,0,15
790 PEN 7+INT(RND*4):LOCATE x+INT(RND
*2),23-INT(RND*2):PRINT CHR$(216+INT(
RND*8))
800 FOR delay=1 TO 100:NEXT
810 NEXT
820 PRINT CHR$(22);CHR$(0)
830 SOUND 130,0,0,0:SOUND 132,0,0,0
840 lives=lives-1
850 IF lives=0 THEN RETURN
860 FOR i=30 TO 610 STEP 4
870 MOVE i,20:DRAW 320,250,14
880 NEXT
890 f=0:a=10:b=12:c=10:d=16:h=0:gear=
1:ok=-1
900 PEN 7
910 RETURN
920 REM -------- Screen ----------
930 PRINT CHR$(23);CHR$(0):INK 8,15:I
NK 9,6:INK 10,12:INK 11,3
940 MODE 0:INK 14,0:INK 3,20:INK 7,2:
BORDER 1
950 INK 15,11:PAPER 15:CLS
960 ORIGIN 1000,1000,1000,2000,3000,4
000:ORIGIN 0,0,0,640,0,400:CLG 15:PLO
T -10,-10,3:FOR j=1 TO 10:x=RND*640:y
=252+RND*150:TAG:FOR i=1 TO 10:MOVE x
+RND*32,y+RND*16:PRINT CHR$(231);:NEX
T:TAGOFF:NEXT
970 INK 12,4:INK 13,8
980 y=254:j=0
990 PLOT -10,-10,j MOD 3
1000 FOR i=y TO y-j STEP -2
1010 MOVE 0,i:DRAW 638,i
1020 NEXT
1030 j=j+1:y=y-j
1040 IF y>0 THEN 990
1050 FOR i=1 TO 10
1060 x=640*RND:y=280+RND*20:PLOT -10,
-10,13:r=0-4*INT(RND*3)
1070 FOR j=-20-4*(INT(RND*4)) TO 20+4
*(INT(RND*4))
1080 MOVE x,y:DRAW x+j,252
1090 IF j=r THEN PLOT -10,-10,12
1100 NEXT
1110 NEXT
1120 x=320:PLOT -10,-10,14
1130 FOR i=0 TO 640 STEP 4
1140 MOVE i,0:DRAW x,252
1150 IF i=320 THEN x=x+4
1160 NEXT
1170 x=316:PRINT CHR$(23);CHR$(3)
1180 FOR i=-32 TO 0 STEP 4
1190 MOVE i,0:DRAW x,252,4
1200 IF i=320 THEN x=x+8
1210 NEXT
1220 FOR i=640 TO 672 STEP 4
1230 MOVE i,0:DRAW 324,252
1240 NEXT
1250 PRINT CHR$(23);CHR$(0):ORIGIN 10
00,1000,1000,2000,3000,4000:CLG 14:OR
IGIN 0,0,0,640,0,400:PLOT -10,-10,11
1260 PEN 7:PAPER 14:LOCATE 2,25:PRINT
 "Score:0     Gear:1"
1270 f=0:a=10:b=12:c=14:d=18:h=0:gear
=1:ok=-1:x=8:score=0:lives=3:dif=6
1280 RETURN
1290 REM ------- Initialise ------
1300 MEMORY &9FFF
1310 DEFINT a-z
1320 dead.interested=-1
1330 RESTORE
1340 ENT 1,5,2,1,5,-1,1
1350 DIM gp(3),name$(10),hiscore(10):
gp(0)=&A200:gp(1)=&A1C0:gp(2)=&A180
1360 FOR i=1 TO 10:name$(i)="The A Te
am":hiscore(i)=600-i*50:NEXT
1370 a=&A000
1380 FOR i=1 TO 11
1390 sum=0:READ a$,b$
1400 FOR j=1 TO 21 STEP 2
1410 k=VAL("&"+MID$(a$,j,2)):POKE a,k
1420 sum=sum+k:a=a+1
1430 NEXT
1440 IF sum<>VAL("&"+b$) THEN PRINT "
Error in data...":STOP
1450 NEXT
1460 DATA DD7E003257A0DD7E013258,46A
1470 DATA A026C0DD7E043D87876F11,480
1480 DATA 5000DD4602051910FDE526,3A8
1490 DATA C0DD7E083D87876F115000,43E
1500 DATA DD4606051910FD111000ED,36A
1510 DATA 5379A043F3C53A7AA047E5,5E7
1520 DATA 363F2310FBE17CC6086730,465
1530 DATA 040150C009C110E7E11100,3C8
1540 DATA 003A79A047C53A7AA047E5,4DF
1550 DATA 1A77231310FAE17CC60867,463
1560 DATA 30040150C009C110E6FBC9,4C9
1570 FOR i=0 TO 319:READ j:POKE &A100
+i,j:NEXT
1580 RETURN
1590 REM Racing car
1600 REM Rows=16/Columns=8
1610 DATA 63,3,63,195,195,63,3,63,63,
3,107,207,207,151,3,63,63,3,75,139,71
,135
1620 DATA 3,63,3,3,107,139,71,151,3,3
,23,3,199,199,203,203,3,43,23,23,195,
203
1630 DATA 199,195,43,43,23,23,199,75,
135,203,43,43,23,23,199,143,79,203,43
,43
1640 DATA 23,67,15,15,15,15,131,43,23
,7,207,143,79,207,11,43,23,67,199,79,
143
1650 DATA 203,131,43,23,23,195,195,19
5,195,43,43,23,23,107,195,195,151,43,
43
1660 DATA 23,23,63,63,63,63,43,43,23,
23,63,63,63,63,43,43,3,23,63,63,63,63
,43
1670 DATA 3
1680 REM Big racer
1690 REM Rows=8/Columns=8
1700 DATA 63,3,63,131,67,63,3,63,3,3,
67,195,195,131,3,3,23,3,199,79,143,20
3
1710 DATA 3,43,71,23,199,143,79,203,4
3,139,23,3,15,15,15,15,3,43,71,23,199
,143
1720 DATA 79,203,43,139,23,23,107,195
,195,151,43,43,3,23,63,63,63,63,43,3
1730 REM Medium racer
1740 REM Rows=8/Columns=8
1750 DATA 63,63,63,63,63,63,63,63,63,
43,23,131,67,43,23,63,63,3,3,195,195,
3
1760 DATA 3,63,63,3,107,15,15,151,3,6
3,63,3,71,143,79,139,3,63,63,3,107,19
5
1770 DATA 195,151,3,63,63,3,63,63,63,
63,3,63,63,63,63,63,63,63,63,63
1780 REM Small racer
1790 REM Rows=8/Columns=8
1800 DATA 63,63,63,63,63,63,63,63,63,
63,63,63,63,63,63,63,63,63,43,43,23,2
3
1810 DATA 63,63,63,63,23,195,195,43,6
3,63,63,63,7,15,15,11,63,63,63,63,23,
63
1820 DATA 63,43,63,63,63,63,63,63,63,
63,63,63,63,63,63,63,63,63,63,63
1830 REM -------- tune --------
1840 WHILE INKEY(47)<0 AND INKEY(76)<
0
1850 RESTORE 1960:READ pitch:SOUND 12
9,213,200,12:SOUND 130,253,2000,12:SO
UND 1,127,2000,12
1860 WHILE pitch>0 AND INKEY(47)<0 AN
D INKEY(76)<0
1870 SOUND 4,pitch,10,14:SOUND 4,159,
10,14:READ pitch
1880 WEND
1890 READ pitch:SOUND 129,319,200,13:
SOUND 1,213,200,13:SOUND 1,159,200,13
1900 WHILE pitch>0 AND INKEY(47)<0 AN
D INKEY(76)<0
1910 SOUND 4,pitch,12,15:READ pitch
1920 WEND
1930 WEND
1940 IF INKEY(47)>-1 THEN up=0:down=2
:left=8:right=1 ELSE up=72:down=73:le
ft=74:right=75
1950 RETURN
1960 DATA 142,127,119,106,119,127,142
,127,119,106,119,127,127,119,106,95,1
06,119,127,119,106,95,106,119,119,106
,95,84,95,106,119,106,95,84,95,106
1970 DATA 0
1980 DATA 80,84,95, 84,95,106, 95,106
,119, 106,119,127
1990 DATA 80,84,95, 84,95,106, 95,106
,119, 106,119,127
2000 DATA 142,127,119, 127,119,106, 1
19,106,95, 106,95,84,80
2010 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
2020 REM ----- instructions ------
2030 MODE 1:CALL &BB4E:CALL &BC02
2040 INK 0,13:BORDER 13:INK 1,18:INK
2,1:INK 3,13
2050 WINDOW #1,11,31,1,3:PAPER #1,2:C
LS #1
2060 WINDOW #1,10,30,2,4:PAPER #1,1:C
LS #1
2070 PEN 3:LOCATE 1,25:PRINT "Le  Man
s":PLOT -10,-10,3
2080 FOR x=0 TO 128 STEP 2:FOR y=2 TO
 14 STEP 2
2090 IF TEST(x,y) THEN PLOT x+230,y+y
+344:PLOT x+230,y+y+346
2100 NEXT:NEXT:PEN #1,2:INK 3,6
2110 WINDOW #1,2,40,5,24:PAPER #1,3:C
LS #1
2120 WINDOW #1,1,39,6,25:PAPER #1,1:C
LS #1
2130 PRINT #1
2140 PRINT #1," After a poor start yo
ur turbo charged"
2150 PRINT #1," formula 3 car is rapi
dly catching up"
2160 PRINT #1," with the back markers
."
2170 PRINT #1
2180 PRINT #1," You'll need skill, da
ring and quick"
2190 PRINT #1," reactions if you are
to win the Grand"
2200 PRINT #1," Prix."
2210 PRINT #1
2220 PRINT #1," Use the left and righ
t cursor keys to"
2230 PRINT #1," move your car left an
d right. Cursor"
2240 PRINT #1," up and down change ge
ar."
2250 PRINT #1:PRINT #1:PRINT #1:PRINT
 #1,SPC(11)"Or use Joystick"
2260 PEN #1,1:PAPER #1,2:LOCATE #1,5,
19:PRINT #1," Press SPACE or FIRE to
start ":PAPER #1,0
2270 WHILE INKEY$<>"":WEND
2280 GOSUB 1830
2290 RETURN
2300 REM ------- Starting grid -----
2310 i=REMAIN(0)
2320 MODE 1:BORDER 1:INK 0,13:INK 1,2
0:INK 2,6:INK 3,26
2330 DRAW 0,398,1:DRAW 638,398:DRAW 6
38,0:DRAW 0,0
2340 PAPER 2:PEN 3:LOCATE 11,2:PRINT
"   Starting  Grid   ":PAPER 0
2350 FOR i=1 TO 10
2360 LOCATE 11+10*(i AND 1),2+i*2:PEN
 1:PRINT name$(i)
2370 NEXT
2380 PAPER 3:PEN 2:LOCATE 6,24:PRINT
" Press SPACE or FIRE to start "
2390 WHILE INKEY$<>"":WEND:GOSUB 1830
2400 SOUND 129,0:SOUND 130,0:SOUND 13
2,0
2410 RETURN
2420 REM ------- high score -------
2430 MODE 1:INK 0,0:INK 1,24:INK 2,20
:INK 3,6
2440 DRAW 0,398,1:DRAW 638,398:DRAW 6
38,0:DRAW 0,0
2450 PEN 1:PAPER 0:LOCATE 7,5:PRINT "
This  is  a  new  high  score"
2460 LOCATE 10,10:PEN 3:PRINT "Please
 type in your name"
2470 LOCATE 14,20:PAPER 2:PEN 3:PRINT
 " ?.......... "
2480 LOCATE 16,20:k$="":name$(10)=""
2490 WHILE INKEY$<>"":WEND
2500 WHILE k$<>CHR$(13)
2510 IF k$>CHR$(31) AND k$<"z" AND LE
N(name$(10))<10 THEN name$(10)=name$(
10)+k$:PRINT k$;
2520 IF k$=CHR$(127) AND LEN(name$(10
)) THEN name$(10)=LEFT$(name$(10),LEN
(name$(10))-1):PRINT CHR$(8);CHR$(16)
;
2530 k$=INKEY$
2540 WEND
2550 hiscore(10)=score
2560 FOR i=10 TO 2 STEP -1
2570 IF hiscore(i)>hiscore(i-1) THEN
k$=name$(i):name$(i)=name$(i-1):name$
(i-1)=k$:score=hiscore(i):hiscore(i)=
hiscore(i-1):hiscore(i-1)=score
2580 NEXT
2590 RETURN
```

SUPERCHARGE

SUPERPOWER SIDEWAYS ROM CARD (Ref A101)

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....SEPTEMBER
Spreadsheet; Graphics/Statistics....OCTOBER

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 DISK USER'S UTILITIES (Ref A103)

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, menmonics 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 A101)

SUPERPOWER MAILING LIST AND CLUB MEMBERSHIP PROGRAM (Ref A102)	£39.95	SUPERPOWER DISK USER'S UTILITIES ROM (Ref A103)	£39.95
SUPERPOWER PROGRAMMER'S TOOLBOX (Ref A104)	£39.95	SUPERPOWER ASSEMBLER, DISASSEMBLER & MACHINE CODE MONITOR (Ref A105)	£39.95
SUPERPOWER WORD PROCESSOR (Ref A106)	£39.95	SUPERPOWER DATABASE (Ref A107)	£39.95

SUPERPOWER MAILING LIST AND CLUB MEMBERSHIP PROGRAM (Ref A102)

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).

SUPERPOWER ASSEMBLER, DISASSEMBLER & MACHINE-CODE MONITOR (Ref A105)

This suite of routines represents the complete Development Package for the Amstrad Z80 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 Z80 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.

SUPERPOWER PROGRAMMER'S TOOLBOX for the CPC 464 (Ref A104)

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 charter. ECHO ON/ECHO OFF: Printer output on and off. PAGE ON/PAGEOFF: 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.

ROM-BASED SOFTWARE FULFILS THE PROM

YOUR AMSTRAD.

SUPERPOWER WORD PROCESSOR (Ref A106)

This program incorporates the most useful facilities offered by the best Word Processors currently available on the BBC Micro and other up-market Home Computers.

Text Entry/Editing. User choice of 40 or 80 column mode. Document formatting achieved through use of embedded commands. Text can be formatted and justified on screen, if required. Find <and Replace>with 'wild-card' option. Word count, character count and page numbering reported.

Formatting. Multiple rulers are provided, allowing sophisticated layout of documents e.g. different combinations of TAB stops at different points in the text. All the normal controls are available e.g. page lengths & width, margins, indents, tab stops, justification, headers, footers etc., together with output of control characters to the printer.

Virtually UNLIMITED Document Length. The disk routines in the program handle multiple data files representing parts of a total document, with loading and saving carried out automatically.

Simple Calculator. A calculator window can be called to carry out simple arithmetic calculations. It is also possible to embed a calculation in the text, with the result being calculated and printed on output.

Data Exchange. As part of an integrated suite, the program will be able to handle ASCII files originating from the Spreadsheet, Database and Graphics/Statistics Packages.

Disk/External Commands. Direct Access available.

MANY MORE FEATURES. Ask your dealer for information.

SUPERPOWER DATABASE (Ref A107)

This menu-driven program has been designed to be the most comprehensive and flexible Database which can be achieved in 16K of machine code. Of special note are the PROCEDURES which have been implemented, enabling the more advanced user to write simple structured programs to manipulate the database information in order to meet his more specialized requirements. Databases need not be limited in size since the program contains routines for multiple file handling.

Field types — alphanumeric, integer, currency, floating point, logic, calculated and date.

MAIN FILE COMMANDS

— Create/Modify database structures, reports and procedures.

— Create ASCII file from a Database, enabling data to be passed to the Word Processor, Spreadsheet and Graphic/Statistics programs.

— Copy data to new Database.

RECORD EDITING

— Add, amend, delete, insert, view and duplicate.

— Search and Sort. Due to intelligent file structure and the indexing system adopted, both SEARCH and SORT are extremely fast.

REPORTS

Standard — selected fields may be output, together with totalling on those nominated. (Value fields only). Screen or printer output.

Labels — user-selected fields for printing. User-defined label size, number on sheet etc.

User-defined — user created report-format, combining entered text and designated fields. Editing and storing facilities.

SUPERPOWER ADVICE CENTRES

AVON

K & K Computers,
Weston Super Mare
0934 419324.

BEDFORDSHIRE

Chiltern Computers
Luton 0582 455684.

Hobbyte Limited
Luton 0582 457195.

Zedem Computers
Bedford 0234 213645.

DJ Computers
Leyton Buzzard
0525 383929.

BERKSHIRE

Gadgets
Windsor 07535 67211.

CHESHIRE

Northern Computers
Frodsham 0928 35110.

Pyramid Micros
Warrington 0925 35713.

CLEVELAND

Multicoloured Micro Shop
Redcar 0642 486643.

DEVON

Computerbase
Plymouth 0752 672128.

Computer Systems (Torbay)
Paignton 0803 524284.

Georges Booksellers
Exeter 0392 71072.

Georges Booksellers
Plymouth 0752 661011.

Computerware
Exeter 0392 210507.

Syntax Limited
Plymouth 0752 28705.

Seven Counties
Exeter 0392 211211.

DORSET

Seven Counties
Dorchester 0305 66022.

EIRE

Bit by Bit
Bray 001 868 541.

ESSEX

Clarke Coyston Computers
Southend-on-Sea
0702 337161.

Estuary Personal Computers
Southend-on-Sea 0702 614131.

Computerama
Southend-on-Sea 0702 335443.

Colchester Computers
Colchester 0206 47242.

Mirage Microcomputers
Braintree 0376 48321.

Computer Centre
Hornchurch 04024 75613

GLOUCESTERSHIRE

Screen Scene
Cheltenham 0242 528979.

HAMPSHIRE

Electronequip
Fareham 0329 230671.

Alton Computers
Alton 0420 87998.

Timatic Systems
Fareham 0329 236727.

HERTFORDSHIRE

DJ Computers
Stevenage 0438 65501

DJ Computers
Welwyn Garden City
0707 328435.

Hobbyte Ltd.
St. Albans
0727 41396

HUMBERSIDE

Computing World
Beverley 0482 881831.

Ashby Computer Centre
Scunthorpe 0724 871756.

ISLE OF WIGHT

V M S Computer Centre
Newport 0983 525450.

KENT

The Village House of
Computers
Bromley 01 460 7122.

Microway Computers
Rainham 0634 376702.

Marlowe Micros
Canterbury 0227 453531.

Adventurer's Guild
Gravesend 0474 334008.

Numbus Computing
West Wickham
01 777 3160.

MD Micros
Crayford 0322 524123.

LANCASHIRE

Computer World
Bolton 0204 494304.

I M O Computers
Burnley 0282 54299.

LEICESTERSHIRE

Mays Hi-Fi Ltd
Leicester 0533 22212.

LONDON

Adams World of Software
London N12 01 446 7170.

Vic Oddens
London SE1 01 403 1988.

Computers Inc.
London NW11
01 209 0401.

G & B Computers
London W1 01 580 3702.

Ramco Electronics
London NW7 3TG
01 906 3363.

MANCHESTER

Mighty Micro
Fallowfield 061 224 8117.

Celcom Systems
Manchester
061 834 2808.

MERSEYSIDE

MicroByte
Wallasey 051 6306933.

Hargreaves Computers
Liverpool 051 525 1782.

MIDDLESEX

J K L Computers
Uxbridge 0895 51815.

Screens Microcomputers
Northwood
09274 20664.

Trionic Limited
Harrow 01 861 0036.

NORFOLK

Abacus
Norwich 0603 614441.

Thetford Micros
Thetford 0842 61645.

NORTHERN IRELAND

Botanic Computer Centre
Belfast 0232 240798.

Fun-cational Computers
Lurgan 07622 28543.

NORTHUMBERLAND

Alnwick Computers
Alnwick 0665 604816.

NOTTINGHAM

G A Computers
Ruddington 0602 211225.

Computer Market
Nottingham
0602 586454.

S P Electronics
Nottingham 0602 640377

SCOTLAND

Micro Mart
Kirkcaldy 0592 206251.

SOMERSET

The Forum
Chard 04606 5009.

STAFFORDSHIRE

Leek Computer Centre
Leek 0538 387859.

Computer Cabin
Newcastle under Lyne
0782 636911.

Town Computer Centre
Stoke on Trent
0782 287540.

SURREY

Computerama
Burgh Heath
07373 54717.

Computersolve
Surbiton 01 390 5135.

Micro Workshop
Epsom 03727 21533.

SUSSEX

Gamer
Brighton 0273 698424.

Computermania
Chichester 0243 779263.

Computermania
Littlehampton
0903 716187.

TYNE AND WEAR

Sunderland Computer
Centre
Sunderland
0783 655711/655736.

WALES

Gwynedd Computers
Caernarfen 0286 2610.

Cardigan Electronics
Cardigan 0239 614483.

Bucon Limited
Swansea 0792 467980.

Teleview
Neath 0639 53282.

Cwmbran Computer
Centre
Cwmbran 06333 60996.

WARWICKSHIRE

O E M Computer Systems
Rugby 0788 70522.

Micro City
Nuneaton 0203 382049.

WILTSHIRE

Barbury Computers
Swindon 0793 611487.

Trowbridge Computer
Shack
Trowbridge 02214 67299

YORKSHIRE

Micro Power
Leeds LS7 0532 458800.

Danum Computer
Systems
Doncaster 0302 20355.

The Computer Room
Barnsley 0226 204000.

Just Micro
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 Office, Micro Power Ltd.,
Northwood House, North Street, Leeds LS7 2AA. TELEPHONE (0532) 434006.

SUPER POWER

SERIOUS SOFTWARE
FROM THE MICROPOWER
ORGANISATION

ISE OF YOUR AMSTRAD.

AND now you see it– OR now you don't...

GEOFF TURNER and MICHAEL NOELS continue their Amstrad graphics series by delving deeper into logical colours . . . and squaring up to some problems

WE saw in the September issue how multiplane images can be created by careful use of the logical OR operator. Using OR and a bit of thought, shapes can be laid on top of, below or even between other shapes.

Last time we were putting things on the screen. This month we'll be looking at what happens when we begin to remove shapes from the screen.

After all, if we can lay one drawing over another then we should be able to remove the top one and reveal what's underneath. The difference is between drawing one shape over another, obliterating what went before, and using more subtle, logical methods to keep track of the previous shape.

Before we get to these techniques, let's first see what happens when normal drawing techniques are used.

Program I uses the subroutine starting at line 160 to draw two squares. The first is drawn in pen 1, the second in pen 2. On entry to Mode 0 these are filled with bright yellow and bright cyan ink – ink numbers 24 and 20 respectively.

The squares, which are slightly overlapping, are drawn with the normal graphics pen. We haven't used the PRINT CHR$(23) that we came across last time.

The cyan square appears on top of the yellow one, simply because we plotted it last. The program now halts at the WHILE . . . WEND of line 110 until a key is pressed.

The program now attempts to remove the cyan square by redrawing it in the background colour using pen 0 – blue, ink number 1. It's less than successful. You'll see that a corner is taken out of the yellow square where the overlay occurred.

The point to grasp is that if we use the normal graphics pen the computer doesn't remember what colour was under the cyan square. When we satisfy the loop by pressing a key the entire area occupied by the second square is redrawn using pen 0. Hence the chunk out of the yellow square.

To provide the micro with what amounts to some form of memory, we need to use the logical OR function. Let's modify Program I by adding line:

```
30 PRINT CHR$(23);CHR$(3)
```

Now we have drawn the two squares again, but this time we've used the OR function. Notice when the squares overlap a third colour, red, appears.

If you've followed what we said last time you should have no difficulty in working out why the third colour is there.

The yellow square is drawn first, and since it is ORed with pen 0 (the blue background) it is unchanged. You'll remember that 1 OR 0 is 1. Similarly where the cyan square is ORed on to the background it remains cyan.

However where the squares

```
10 REM PROGRAM I
20 MODE 0
50 colour=1
60 x=200:y=300
70 GOSUB 160
80 colour=2
90 x=250:y=250
100 GOSUB 160
110 WHILE INKEY$="":WEND
120 colour=0
130 x=250:y=250
140 GOSUB 160
150 END
160 REM DRAW SQUARE
170 MOVE x,y
180 FOR strip=1 TO 25
190 MOVER 4,-100
200 DRAWR 0,100,colour
210 NEXT
220 RETURN
```

Program I

overlap it's a different kettle of fish. Here we are ORing the top left corner of the cyan square on to the bottom right corner of the yellow square.

This ORing of pens 1 and 2 results in the overlap being drawn using pen 3. This is because 1 ORed with 2 gives 3 as the result. As pen 3 is filled with bright red ink (ink number 6) the overlap of the squares is red.

Of course we don't really want this third colour. It's a nuisance. We get rid of it by changing the ink colour in pen 3. Try adding:

```
40 INK 3,20
```

to Program I.

Now when you run the program the overlapping portion of the squares will still be drawn with pen 3 but this time it's filled with bright cyan ink. This gives the impression that the cyan square is in front of the yellow one. You can reverse the process by filling pen 3 with yellow ink using:

```
INK 3,24
```

If you enter this in direct mode you'll see the yellow square suddenly leap in front of the blue one.

Observant readers may have noticed that after adding lines 30 and 40 to Program I pressing a key seems to have no effect. This is because we're now using OR logic.

After the delay at line 110 the program goes on to call the subroutine to draw the second square again. This time, however, it's using pen 0 and OR logic. Now anything ORed with 0 remains the same – 1 OR 0 is 1, 0 OR 0 is 0 – so the result is that as the same pens are used to draw the second square, the colours are the same. Hence there appears to be no change.

So far we haven't done anything using OR that we couldn't have achieved using the normal default logic. Let's get back to our problem of removing the cyan square to reveal the complete yellow one. We do this by taking a close look at which pens are being used to draw on which part of the screen. Figure I tells us what we need to know.

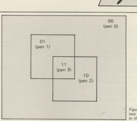


Figure I: ORing two squares on to the screen

We can see that the background is drawn in pen 0, 00 in binary. The yellow square is in pen 1, 01 in binary, while the cyan square is drawn in pen 2, 10 in binary. The overlap is drawn using pen 3 (11). We see this as cyan because of the change of ink in line 40.

A little thought should convince you that to erase the cyan square and leave the yellow intact needs two operations. The overlap has to become yellow to complete the yellow square, while the rest of the cyan square becomes blue, blending in with the background. There are two ways of doing this.

The first method is to change the inks in pens 2 and 3 to blue and yellow respectively. By now you should be capable of doing this using the INK command. However this can be a bit complicated and it seems rather wasteful having two pens filled with the same colour ink. It's better to use the logical colours to achieve the same end.

The method is exactly the same, patching in the colours as required, but using logic. We need to apply some logical function which will change the parts drawn using pen 2 (cyan) to pen 0 (blue) and the overlapping pen 3 (cyan) to pen 1 (yellow).

This is exactly the same as before except that now we don't mess around changing the inks in the pens, we use the logical drawing methods to pick a pen that already contains the colour of ink we need. In this case the AND function will do the job for us, as you'll see in a moment.

The question is, which number should we AND on to the screen to produce the correct result. If you examine September's AND logic chart you should find several numbers which will give the correct result. For example 2 AND 1 will produce 0 (10 AND 01 gives 0), while 3 AND 1 will produce 1 (11 AND 01 gives 1) which is exactly what we want.

From the table you'll see that the numbers 1, 5, 9 and 13 all produce the required results. Actually if you look at these numbers in binary it's not difficult to see why they all work. Table I shows the decimal and binary equivalents.

Notice that the last two bits – bits 0 and 1 measured from the rightmost bit – are always the same. We are ANDing these numbers with pen numbers 2 (0010) and 3 (0011), so we are really only using bits 0 and 1. Bits 2 and 3 may be effectively ignored as whatever value they hold will turn to 0 when ANDed with the 0s in bits 2 and 3 of the pen numbers 2 and 3. The result is that all the four numbers above will produce the same result.

Decimal	Binary
1	0001
5	0101
9	1001
13	1101

Table I: Decimal and binary equivalents

Let's see if it works correctly. Remember we need to change pen 2 (cyan) back to pen 0 (blue). If we AND together 1 and 2 then the result is that pen 0 is used. This will restore our blue background.

We also need to change pen 3, which is filled with cyan, back to pen 1 (bright yellow) to make sure that there's no chunk out of the yellow square. This is done by ANDing pen 1 (01) with pen 3 (11) to get pen 1 (01) which is, of course, bright yellow. Program II shows how it works.

After drawing the two squares it waits for a key to be pressed before removing the cyan square. You'll see the yellow square is restored to normal. Press a key again and the cyan square is redrawn. The sequence is repeated every time a key is pressed.

The program works by ORing the two squares on the screen using pen 1 for the yellow square and pen 2 for the cyan one. The ink in pen 3 is changed to cyan to allow for the overlap.

Now line 110 holds things up until a key is pressed. Once this is done the program uses the subroutine at 210 to draw over the second square. This time, however, it's using pen 1 with AND logic. The result of this is shown in Figure II.

As you can see, although pen 1 is the selected graphics pen, the AND logic ensures that pen 0 (blue) is used for drawing most of the square while pen 1 (yellow) is used for the overlap. Hence the second square is effectively erased.

```
10 REM PROGRAM II
20 MODE 0
30 INK 3,20
40 colour=1
50 code=3
60 x=200:y=300
70 GOSUB'210
80 colour=2
90 code=3
100 x=250:y=250
110 GOSUB 210
120 WHILE INKEY$="":WEND
130 colour=1
140 code=2
150 x=250:y=250
160 PRINT CHR$(23);CHR$(2);
170 GOSUB 210
180 WHILE INKEY$="":WEND
190 GOTO 80
200 END
210 REM DRAW SQUARE
220 PRINT CHR$(23);CHR$(code);
230 MOVE x,y
240 FOR strip=1 TO 25
250 MOVER 4,-100
260 DRAWR 0,100,colour
270 NEXT
280 RETURN
```

Program II

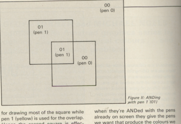


Figure II: ANDing with pen 1 (01)

The program then waits for a key to be pressed before the GOTO of line 190 sends the program back to line 80.

Now the bottom square is drawn again, this time using pen 2 with OR logic. The result is that the first square appears again. Try ORing 2 with the pen numbers shown in Figure II and you'll see why. The program now comes to line 120 and, when a key is pressed, goes on to erase the cyan square as before.

So we've used the OR function to put the shapes on to the screen and used the AND function to remove them again.

Try to grasp the fact that the AND function is being used as a mask (see Mike Bibby's excellent Bits and Bytes series for more on masking.) This is done by selecting a value which will mask out, or set to zero, the bits we do not want from the pens that are on the screen.

Notice that we used the number 1 as an AND mask in the last example. Pen 1 is in fact bright yellow, but, remember, we weren't actually using it to put yellow on the screen – although by coincidence our first square was in ink 1.

It was simply being used as a suitable mask to remove the bits we didn't want from the pen numbers already used on screen and so leave the pen numbers we did.

You'll remember that we said that 5, 9 and 13 would do just as well. Try using them instead of the 1 in line 130 and you'll see that the action of the program is exactly the same. We're not trying to draw in pens 5, 9 or 13, we're just using them because when they're ANDed with the pens already on screen they give the pens we want that produce the colours we need.

Although we have only used two overlapping squares up to now, it is possible to expand on the same idea by using three or even more squares. However it does become more complex, and it leaves less choice of colours. Remember that although we have 16 colours initially in Mode 0, every time we use an INK command to change a colour it reduces our number of colours by one.

In Program I we used two pens both filled with bright cyan, so that cut down on the number of pens available to use for other colours.

We have seen that when using only two squares we need to change one ink colour, so that in Mode 1 (four pens) we can't progress any further because of the colour limitation. The first pen is used for background, the second for one square, the third for the other and the fourth for the overlap.

However in Mode 0 with its 16 pens we can demonstrate how to place three overlapping coloured squares on the screen and then remove any of them while still leaving the others intact.

Program III shows how it's done. When you first run the program leave out lines 40 to 70 so that the shapes are drawn without any ink.changes.

Press keys 1, 2 and 3 to make the squares appear. As the three overlapping squares are drawn you'll see that there are now several different colours on the screen. The three squares have been ORed onto the screen using pens 1, 2 and 4. There are eight colours on the screen as a result of all the ORing.

Now add lines 40 to 70 to the

```
10 REM PROGRAM III
20 MODE 0
30 REM CHANGE INKS
40 INK 3,20
50 INK 5,26
60 INK 6,26
70 INK 7,26
80 REM DETECT KEYS
90 WHILE -1
100 IF INKEY(64)=0 THEN GOSUB 190
110 IF INKEY(65)=0 THEN GOSUB 250
120 IF INKEY(57)=0 THEN GOSUB 310
130 IF INKEY(64)=32 THEN GOSUB 370
140 IF INKEY(65)=32 THEN GOSUB 430

150 IF INKEY(57)=32 THEN GOSUB 490
160 WEND
170 END
180 REM DRAW SQUARE 1
190 COLOUR=1
200 CODE=3
210 x=200:y=300
220 GOSUB 540
230 RETURN
240 REM DRAW SQUARE 2
250 COLOUR=2
260 CODE=3
270 x=250:y=275
280 GOSUB 540
290 RETURN
300 REM DRAW SQUARE 3
310 COLOUR=4
320 CODE=3
330 x=225:y=225
340 GOSUB 540
350 RETURN
360 REM ERASE SQUARE 1
370 colour=6
380 x=200:y=300
390 code=2
400 GOSUB 540
410 RETURN
420 REM ERASE SQUARE 2
430 colour=5
440 x=250:y=275
450 code=2
460 GOSUB 540
470 RETURN
480 REM ERASE SQUARE 3
490 colour=3
500 x=225:y=225
510 code=2
520 GOSUB 540
530 RETURN
540 REM DRAW /.ERASE SELECTED SQUARE
550 PRINT CHR$(23);CHR$(code);
560 MOVE x,y
570 FOR strip=1 TO 25
580 MOVER 4,-100
590 DRAWR 0,100,colour
600 NEXT
610 RETURN
```

Program III

program, and when we run it we end up with our three squares in the correct colours. We may now remove or replace any one of the squares by pressing keys 1, 2 or 3 in conjunction with the Shift key.

Pressing one of the numbers will draw the appropriate square, while holding down the Shift key and pressing 1, 2 or 3 will erase that square. Notice that at all times the correct colours are maintained on the screen. There are no overlaps.

Figure III shows the pens being used to draw each section of the squares. They are all the result of ORing the three graphics pens, 1, 2 and 4. When all three squares overlap we need to OR all three numbers to find the resulting ink number. The logic functions can be applied to more than two numbers.

In this program we're using the OR function on the numbers 1, 2 and 4. The resulting pen number at the centre is then binary 111 or 7 in decimal as shown by:

```
001 OR 010 OR 100 = 111
```

Each number is ORed in turn to get the final result.

You may have noticed that each square is represented by a separate bit in the above numbers. Bit zero, the rightmost bit remember, can be used to determine whether square 1 is on the screen, while bit one identifies the second square and finally bit two refers to the third square.

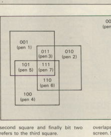


Figure III: ORing three squares on to the screen

If the appropriate bit is set to one the square should be visible. Otherwise if the bit is set to zero the square is erased.

If you find this a little difficult to understand try reconstructing Figure III with pencil and paper, noting down the results in binary.

First OR the yellow square, pen 1 on to the blue background, pen 0. Then OR the cyan square, pen 2, on to these and notice the results. Finally OR the white square, pen 4, on to the screen and see the results. You should see how bit one, the second from the right, isn't set in any of the overlaps until the second square is on screen. Similarly, bit two isn't set until the third square makes its appearance.

To erase any square all we need to do is make the appropriate bit zero by using a suitable mask and applying the AND logic. In Program III we've used 6 (110) as the mask to erase the yellow square, 5 (101) as the mask to erase the cyan square and 3 (011) as the mask to erase the white square.

If you've made your own pencil and paper version of Figure III try working backwards, erasing successive squares with their AND masks. You'll soon see how they work.

It's possible to draw a fourth ▶

square and still make the program work. We would use bit 3 to represent the fourth square, so a suitable ink to draw the square would be 8 (binary 1000 – cyan) although you may like to change the actual colour of ink 8 to contrast with the bright cyan we've used until now.

We'll leave you to work out how to modify Program III to work with four squares. Remember to OR the squares on to the screen, then AND them off again with a suitable mask.

Up to now we have only used the OR and AND functions. It seems that we have been neglecting the EOR function. However EOR has a very useful property, as we shall see with Program IV.

```
10 REM PROGRAM IV
20 MODE 0
30 PRINT CHR$(23);CHR$(1);
40 x=250:y=250
50 WHILE -1
60 MOVE x,y
70 FOR strip=1 TO 25
80 MOVER 4,-100
90 DRAWR 0,100
100 NEXT
110 WHILE INKEY$="":WEND
120 WEND
```

Program IV

This draws one square in pen 1 (bright yellow). After drawing it the program waits for a key to be pressed, when it then erases the square. Press a key again and the square is redrawn. The process is repeated every time a key is pressed.

This demonstrates how the EOR function works. You see, if we EOR a first number with a second one, then we EOR the second number again with the result, we end up with the original number again.

A look at some binary numbers will make this clearer. If we use 0 as our first number, and 1 as our second number then in binary:

```
01 EOR 00  =  01
00 EOR 01  =  01
```

Notice that we have ended up with the original number again. This rule applies to any numbers, and the sequence will continue to repeat itself over and over again.

You might like to try this using the following as a direct command:

```
PRINT n1 XOR n2 XOR n2
```

substituting numbers for *n1* and *n2*. The computer will always respond with the answer *n1*.

If we apply this to pen numbers then we will always end up with our original pen colour. This is exactly what is happening in Program IV. We are simply EORing the same pen number on to the screen every time a key is pressed, which results in our square appearing and disappearing over and over again.

In Program IV we have only drawn the square over the background pen, but the EOR logic will work equally well over a number of colours. Program V demonstrates this by drawing 16 coloured strips.

When a key is pressed we EOR another 16 strips horizontally across the vertical strips. Press a key again and we EOR these horizontal strips off again, leaving all the original colours intact.

The multicolour pattern produced by the horizontal and vertical strips is a replica of the EOR logic table which we produced in the last article, but this time the table consists of colours rather than numbers.

```
10 REM PROGRAM V
20 MODE 0
30 FOR colour=0 TO 15
40 FOR strip=1 TO 40 STEP 4
50 MOVE 40*colour+strip,16
60 DRAW 40*colour+strip,399,colour
70 NEXT
80 NEXT
90 PRINT CHR$(23);CHR$(1);
100 WHILE -1
110 FOR colour=0 TO 15
120 FOR row=1 TO 24 STEP 2
130 MOVE 0,399-(24*colour+row)
140 DRAW 639,399-(24*colour+row),colo
ur
150 NEXT
160 NEXT
170 WHILE INKEY$="":WEND
180 WEND
```

Program V

Of course when we EOR a single colour across several other colours the line becomes multicoloured. To produce a single colour strip we would need to use several ink changes. This is shown in Program VI where a bright green horizontal strip is drawn across four coloured vertical strips.

The program uses five ink changes, so reducing the number of colours available by five. So although relatively easy to use, you can see that the EOR function can be rather extravagant in its use of inks.

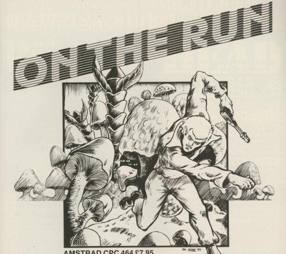
And that's where we finish for this month. Don't worry if you didn't take it all in at first reading. Logical colours take a bit of getting used to. The secret is to try them out for yourself.

Once you think you understand how the programs work try changing them. How about the yellow square coming between the cyan and the white in Program III?

● *That should keep you busy until next month, when we look at rubber banding techniques.*

```
10 REM PROGRAM VI
20 MODE 0
30 FOR colour=1 TO 5
40 FOR strip=1 TO 40 STEP 4
50 MOVE 40*colour+strip+150,200
60 DRAW 40*colour+strip+150,300,colou
r
70 NEXT
80 NEXT
90 PRINT CHR$(23);CHR$(1);
100 INK 13,18
110 INK 14,18
120 INK 15,18
130 INK 8,18
140 INK 9,18
150 colour=12
160 WHILE -1
170 FOR strip=1 TO 200 STEP 4
180 MOVE 150+strip,225
190 DRAW 150+strip,275,colour
200 NEXT
210 WEND
```

Program VI



AMSTRAD CPC 464 £7.95

Other Amstrad titles include:–

DARK STAR • TANK BUSTERS

Selected Design Design titles are available from:–

Boots WOOLWORTH John Menzies
COMET WHSMITH

and all leading software stockists, or direct from:–

Design Design

Design Design Software,
125 Smedley Road, Cheetham Hill, Manchester M8 7RS
Trade enquiries: – 061 205 6603
Price includes p & p

Computing with the AMSTRAD

is looking for more programmers and writers to meet the ever-increasing demand from our readers for more and more top-quality features.

So if you can create exciting games or useful utilities, or can write about aspects of using the Amstrad in the easy-to-understand way that has made this magazine such an outstanding success, then please get in touch with us.

Send your programs – or simply your ideas – to:

Contributions Editor
Computing with the Amstrad
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY

● *For a copy of our Programmers' guidelines send a stamped addressed envelope to the above address.*

One of America's most popular games

THE ULTIMATE IN BATTLE ACTION...

BEACH-HEAD™

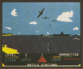
AMSTRAD 64

- Incredible 3-D Graphics
- Unbelievable Sound Effects
- Unique Games Concept
- Multiple Screens
- High Resolution Scenario
- 100% Machine Language

Voted by U.S. Billboard magazine as the best game for sound and graphics on the Commodore 64

It's a unique arcade experience in sound and vision and a stunning display of Amstrad capabilities.

Another quality product from *ACCESS* Software.



Available on CASSETTE £9·95 for AMSTRAD

THE ULTIMATE IN AMERICAN SOFTWARE FOR YOU ON U.S. GOLD

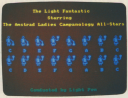
U.S. Gold is stocked by all leading computer stores including:

BOOTS WHSMITH JOHN MENZIES WILDINGS WOOLWORTH

DEALERS! For information on how to become a U.S. Gold Stockist write to: CentreSoft Ltd., Unit 10, The Parkway Industrial Centre, Heneage Street, Birmingham B7 4LY. Telephone: 021-359 3020. Telex: 337268. Overseas enquiries welcome.

U.S. GOLD

All American Software



```
The Light Fantastic
Starring
The Amstrad Ladies Campanology All-Stars

  C    D    E    F    G    A    B    C

  C    D    E    F    G    A    B    C

Conducted by Light Pen
```

Pointing t to some L

. . . and more serious uses too The investigated here by ROLAND WA

THE dk'tronics light pen comes with a superb graphics package enabling you to draw diagrams and paint pictures with ease. This is great fun, but what else can it do and how can the light pen be incorporated into your own programs?

There's a short demonstration program in the manual which should give you a few ideas – it's a simple reaction tester.

But our own favourite light pen program is The Light Fantastic, which was originally written for the BBC Micro by Mike Cook. We thought it so good we've now converted it for the Amstrad.

When you run it, first the light pen is calibrated. All you need to do is place the pen on the blue square in the centre of the screen and press Space. The main program then starts.

You'll see two rows of ladies holding bells. These are The Amstrad Ladies Campanology All-Stars. Simply point the light pen at one of the ladies and she will respond with a ring of her bell. This must be one of the easiest ways to play music on the Amstrad . . .

The pitch of the bell held by each of the 16 ladies is printed below them as a guide. There are two complete octaves and you can either play by ear or use these notes to help you.

The actual routines to read and calibrate the light pen are quite straightforward. The subroutine at line 570 reads the pen position. It's taken from the light pen manual.

To find where the pen is, GOSUB 570. It returns with x set to the x position and y set to the y position. The x co-ordinate ranges from 1 to 40 and the y from 1 to 25. This is irrespective of the current mode and is just right for our program, which runs in Mode 1.

It's not important how it works – just copy it into your own program. It reads and writes to addresses on the expansion bus at the rear of the Amstrad if you're interested.

To calibrate the pen it's placed at a known position on the screen. Then the variable *cal* is adjusted until x and y are correct. Lines 260 and 300 check x and y and alter as necessary. You'll need this if you want to use the light pen in your own program.

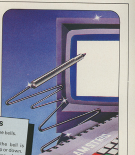
An important point to note is that a light pen needs light to work. The more light there is the better it works, so use bright colours.

The rest of the program is just simple Basic. Two rows of bellringers are printed with the pitch of the bells below. Then there's a loop which checks if you are pointing the pen at

```
10 REM The Light Fantastic
20 REM By Mike Cook
30 REM Coverted by The A Team
40 REM(c)Computing With The Amstrad
50 MODE 1
60 INK 0,3
70 INK 3,18
80 BORDER 3
90 ENV 1,5,2,1,15,-1,3
100 REM ---------------------------
110 REM          Calibrate pen
120 REM ---------------------------
130 LOCATE 15,5
140 PEN 1
150 PRINT "Calibrating..."
160 LOCATE 13,15
170 PRINT "Place the pen on"
180 LOCATE 8,17
190 PRINT "the square and press SPACE
"
200 LOCATE 20,10
210 PAPER 2
220 PRINT " "
230 PAPER 0
240 WHILE INKEY$<>" ":WEND
250 cal=12292
260 GOSUB 570
270 IF y>10 THEN cal=cal+40:GOTO 260
280 IF y<10 THEN cal=cal-40:GOTO 260
290 IF x>20 THEN cal=cal+1:GOTO 260
300 IF x<20 THEN cal=cal-1:GOTO 260
310 REM ---------------------------
320 REM          Main program
330 REM ---------------------------
340 CLS
350 PRINT SPC(10)"The Light Fantastic
"
360 PRINT
370 PRINT SPC(15)"Starring"
380 PRINT
390 PRINT "The Amstrad Ladies Campano
logy All-Stars"
400 LOCATE 10,25
410 PEN 3
420 PRINT "Conducted by Light Pen"
430 PEN 2
440 GOSUB 700:REM girl
450 GOSUB 570:IF x<0 OR x>39 THEN 450
460 IF y=10 OR y=16 THEN 470 ELSE 450
470 ring=x\5
480 IF y=10 THEN p=1 ELSE p=9
490 LOCATE ring*5+2,y-1
500 PRINT USING "&";bell$(updown(ring
+p-1))
510 updown(ring+p-1)=updown(ring+p-1)
 XOR 1
```

g the way e LIGHT fun

too. The light pen's potential is
AND WADDILOVE



one of the ladies and if you are, rings the bell.

I hope this has given you a few ideas and shown how easy it is to use a light pen.

It's especially suitable for young children who haven't learned to read yet. They can simply point the pen at what they want.

Can any reader come up with an educational program for young children using a light pen?

VARIABLES

nval(16)	Pitch of the bells.
bell$(1)	The bells.
updown(16)	Whether the bell is pointing up or down.
x,y	Co-ordinates of the pen.
girl$	A bellringer.

```
520 SOUND 129,nval(ring+p),30,5,1
530 FOR delay=0 TO 25:NEXT
540 GOTO 450
550 END
560 REM ----------------------------
570 REM   Get light pen position
580 REM ----------------------------
590 OUT &1C00,17
600 l=INP(&1F00)
610 OUT &1C00,16
620 h=INP(&1F00)
630 l=h*256+l-cal
640 y=l\40
650 x=l-y*40
660 x=x+1
670 y=y+1
680 RETURN
690 REM ----------------------------
700 REM            Girl
710 REM ----------------------------
720 DIM bell$(1),updown(16),nval(16)
730 SYMBOL AFTER 240
740 SYMBOL 244,1,2,4,4,2,1,1,31
750 SYMBOL 245,128,64,32,32,64,128,12
8,128
760 SYMBOL 246,39,39,39,39,39,39,7,4
770 SYMBOL 247,228,226,225,224,224,22
4,224,16
780 SYMBOL 248,8,16,16,31,4,4,4,6
790 SYMBOL 249,8,4,4,252,64,64,64,96
800 SYMBOL 250,24,126,126,60,60,24,24
,24
810 SYMBOL 251,24,24,240,0,0,0,0,0
820 SYMBOL 252,0,0,0,128,112,24,24,24
830 SYMBOL 253,24,24,60,60,126,126,24
,0
840 girl$=""
850 bs$=CHR$(10)+CHR$(8)
860 FOR a=244 TO 249 STEP 2
870 girl$=girl$+CHR$(a)+CHR$(a+1)+bs$
+CHR$(8)
880 NEXT
890 bell$(0)=CHR$(9)+CHR$(9)+CHR$(250
)+bs$+CHR$(251)+bs$+CHR$(32)
900 bell$(1)=CHR$(9)+CHR$(9)+CHR$(32)
+bs$+CHR$(252)+bs$+CHR$(253)
910 inote=1
920 j=9:GOSUB 960
930 j=15:GOSUB 960
940 RETURN
950 REM ----------------------------
960 REM        Print girl
970 REM ----------------------------
980 PEN 2
990 vnote=67
1000 FOR a=2 TO 38 STEP 5
1010 LOCATE a,j
1020 PRINT USING "&";girl$
1030 LOCATE a,j
1040 PRINT USING "&";bell$(0)
1050 LOCATE a+1,j+4
1060 PRINT CHR$(vnote)
1070 updown(inote)=1
1080 READ nval(inote)
1090 vnote=vnote+1
1100 IF vnote>71 THEN vnote=65
1110 inote=inote+1
1120 NEXT
1130 RETURN
1140 REM ----------------------------
1150 REM            Notes
1160 REM ----------------------------
1170 DATA 119,106,95,89,80,71,63,60
1180 DATA 60,53,47,45,40,36,32,30
```

Give your fingers a rest . . .

All the listings from this month's issue are available on cassette.

See our special offer on Page 85.

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.

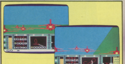
Every ounce of power contained in the micro, and its enhanced sound and graphics capabilities, is used to give the utmost realism to re-creating the most spectacular aeronautical displays ever seen in the skies of Britain.

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!



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 ____________________

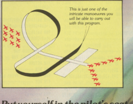
Signed ____________________

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.*

CA 10



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

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

WIN THE POOLS?

NEW! AMSTRA-DRAW 2 – THE LATEST DISC BASED VERSION OF THE ORIGINAL AND BEST POOLS PREDICTION PROGRAM FOR THE AMSTRAD CPC464 (WITH DISC DRIVE) AND CPC664 COMPUTERS!!

"THE COMPUTER CAME UP WITH ONE FIRST DIVIDEND PLUS MINOR DIVIDENDS MAKING A TOTAL OF £6000!"
– J. R. Beckett, Sheffield.

- Supplied with Database containing data on over 12,000 matches since 1980!
- You update the Database each week – but no tedious typing,as team and division names already in program!
- Errors easily corrected – the program even checks your entries!
- Comprehensive instruction manual and menu driven program easy to use, even for a newcomer to computing!
- Will forecast the least likely draws for those who prefer to bet on fixed odds!
- Built in perm generator – complete your coupon direct from the screen!
- Compatible with the DK'tronics speech synthesiser – the only pools prediction program to read you its predictions!
- Each copy is individually updated with all results up to the date of dispatch – no typing in enormous lists of previous matches before using the program!
- Full after sales service, including Database updates and end of season upgrades!

AMSTRA-DRAW 2 (ON DISK ONLY) £12·95
ORIGINAL AMSTRA-DRAW ON TAPE (NO SPEECH) £9·95
(Cheques/P.O.s payable to B. S. McALLEY)
AMSTRA-DRAW (Dept CU), 1 Cowleaze, Chinnor, Oxford OX9 4TD
(Tel: 0844-52426)

Elidon from Orpheus

AMSTRAD CPC 464

ORPHEUS
NEVER·LOOK·BACK...

ELIDON

With the awakening of Spring, in the secret forest of Elidon comes the seven magical flowers of Finvarra which will grace the crown of our Queen. I have been chosen to search for the ethereal flasks of potion needed to make the flowers bloom. Yet the forest has dark, evil places where jealous tree spirits whisper and hide, hungry for the dust that sustains me. I have only you to guide me, in my hazardous task. A stunningly animated arcade adventure for the Amstrad CPC 464

£8.95

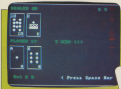
Please send me ______ copies of Elidon @ £8.95 P&P FREE (UK only) £2.00 P&P overseas.

TO:
Name: ....
....
Address: ....
....
....
....
....

Please make cheques or money orders payable to Orpheus Ltd and send together with your order form to:
Orpheus Ltd, The Smithy, Unit 1 Church Farm, Hatley St. George Nr. Sandy, Beds. SG19 3HP.
Please allow approximately two weeks for delivery.

Orpheus Ltd., The Smithy, Unit 1, Church Farm,
Hatley St. George, Nr. Sandy, Beds. SG19 3HP
Tel. Gamlingay (0767) 51481/91 Telex 817117 ETHGEN G

ORPHEUS
NEVER·LOOK·BACK...



```
DEALER 20                    £ 5

PLAYER 17       I WIN !!!

Bet £ 5          < Press Space Bar
```

Bet you'll lose your shirt in SURJIT RANDHAWA's

PONTOON

PONTOON is a simulation of the well known card game, the object being to hold cards of a higher face count than the bank – in this case the computer.

The computer deals both you and itself two cards. You can see your cards but one of the computer's cards remains concealed.

After placing a bet on the opening strength of your hand, you can increase its total face value by "twisting" further cards until you think you have a total to beat the bank's.

You must do this without exceeding a total of 21 or else you "bust" and lose. When adding up the cards values, 10s to king count 10, ace counts 1 or 11, and all other cards count their face value.

The bank will continue to twist cards to you until you decide to "stick" – when you consider your count is high enough. Once you have chosen to stick the bank will take cards until it either beats your total or busts itself.

You start the game with £10 and must bet a minimum of one pound each time you play a hand.

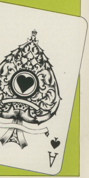
A self-destruct option at the start of the game adds a little extra spice, because should you be brave enough to select it and run out of money not only have you lost your shirt, you've also lost your program and must load it again.

PROGRAM STRUCTURE

10-60	Read card values and set amount.
70-135	Introductory screen.
140	Start of main loop.
220	Start of dealer's routine.
260	Start of player's routine.
470	Sorts out winner.
570	Draws cards.
870	Computer's voice.
940	Pontoon!
1030	Out of money.

MAJOR VARIABLES

x,y	Coordinates of cards.
c	Card number (1-13).
player	0=computer, 1=player.
s	Indicates suit.



```
5 REM(c)Computing with the Amstrad
10 DIM  card$(13) , pack(4,13)
20 MODE 1 : BORDER 0 : INK 0,0 : INK
1,26 : INK 2,18 : INK 3,22
30 REM read in card values
40 FOR n=1 TO 13 : READ card$(n) : NE
XT
50 DATA A,2,3,4,5,6,7,8,9,10,J,Q,K
60 money=10  : REM thats all you get
!
70 REM ******* start screen *******
80 PEN 2 : LOCATE 15,1 : a$="* PONTOO
N *" : GOSUB 870 : GOSUB 920
85 LOCATE 1,15 : PRINT "Would you lik
e some instructions ?"
86 a$=INKEY$ : IF a$="" THEN 86 ELSE
IF LEFT$(UPPER$(a$),1)<>"Y" THEN 120
90 CLS : PRINT : PRINT"The object  of
  the  game  is  to  add  together th
e face value of the cards    that are
 dealt to you and then compare  them
with the dealers total. "
95 PRINT "If your total is greater th
an the       dealers total then you w
in,but your     total must not exceed
 21 or you're BUST!"
100 PRINT "If you get 21 then you get
 a bonus." : PRINT"The keys to use ar
e :" : PRINT :PRINT"   S ... Stick if
 you are satisfied with         your
total" : PRINT "   T ... Twist if you
 want another card"
110 PRINT:PRINT"To make the game more
 interesting you   have the option of
 auto-destruct .      If you run out
of money the computer    resets!"
114 LOCATE 11,23:PRINT"< Press any ke
y >"
115 WHILE INKEY$="":WEND
120 CLS:LOCATE 7,10 : a$="Auto-destru
ct  ON/OFF " : GOSUB 870
130 INPUT autodestruct$ : autodestruc
t$=UPPER$(autodestruct$)
135 a$="Minimum bet is " : LOCATE 12,
18 : GOSUB 870 : PRINT CHR$(163);" 1"
 : GOSUB 920 : GOSUB 920
140 REM ******* main loop *******
150 playertotal=0 : dealertotal=0 : d
ealeraces=0 : playeraces=0
160 CLS : PEN 1 : INK 0,4 : BORDER 4
: GOSUB 920 :a$=" You have ":LOCATE 1
,10:GOSUB 870:PRINT CHR$(163);" ";mon
ey
170 GOSUB 920 : a$=" How much do you
want to bet " :LOCATE 1,14: GOSUB 870

180 INPUT bet : IF bet < 1 OR bet > m
oney THEN LOCATE 2,18 : a$="What ? Tr
y again.  Bet " : GOSUB 870 : LOCATE
25,18 : GOTO 180
190 CLS : INK 0,9 : BORDER 0 : LOCATE
 1,23 : PEN 2 : PRINT"  Bet";" ";CHR$
(163);bet
200 money=money-bet : LOCATE 31,2 : P
RINT CHR$(163);money
205 LOCATE 2,2 : PRINT "DEALER" : LOC
ATE 2,12 : PRINT "PLAYER"
210 MOVE 0,0 : DRAW 639,0 : DRAW 639,
399 : DRAW 0,399 : DRAW 0,0
220 player=0 : start=1 : x=15 : y=250
 : GOSUB 570 : REM dealers first card
230 x=105 : GOSUB 570     : REM deale
rs second card
240 player=1 : x=15 : y=90 : GOSUB 57
0 : x=110 : GOSUB 570
250 TAGOFF : LOCATE 20,23 : PEN 3 : P
RINT "<S>tick / <T>wist"
260 WHILE playertotal<22
270  reply$=UPPER$(INKEY$) : IF reply
$="" THEN 270
280  IF reply$<>"T" THEN  310
290  SOUND 1,100,5 : SOUND 2,110,10 :
 x=x+90 : GOSUB 570 : REM another car
d
300 WEND
310 WHILE playeraces<>0
320  IF playertotal+10>21 THEN player
aces=playeraces-1
330  IF playertotal+10<22 THEN player
aces=playeraces-1 : playertotal=playe
rtotal+10
340 WEND
345 LOCATE 8,12 : PEN 2 : PRINT playe
rtotal
350 IF playertotal>21 THEN LOCATE 17,
12 : PRINT "B U S T !     " : SOUND 1
,3000,40 : GOTO 510
360 IF playertotal=21 THEN  GOSUB 940
 : REM pontoon !
370 player=0 : x=15: y=250 : c=startc
ard : s=startsuite
380 TAG : GOSUB 700 : x=105 : REM dra
w dealers first card
390 WHILE dealertotal < playertotal
400  IF dealeraces=0 OR dealertotal=2
 THEN 430
410  IF dealertotal+10 > 21 THEN deal
eraces=dealeraces-1 : pace=pace+1
420  IF dealertotal+10 < 22 THEN deal
ertotal=dealertotal+10 : pace=pace+1
: dealeraces=dealeraces-1 : GOTO 450

430 GOSUB 920 : REM slow things down
a bit !
440  SOUND 1,90,5 : SOUND 2,70,10 : x
=x+90 : GOSUB 570 : REM another card
450 WEND
460 IF pace<>0 AND dealertotal >21 AN
D dealeraces<>0 THEN dealertotal=deal
ertotal-10 : pace=pace-1 : GOTO 390
465 LOCATE 8,2 : PEN 2 : PRINT dealer
total
470 REM ******* who has won ? *******
480 IF dealertotal > playertotal AND
dealertotal < 22 THEN LOCATE 17,12 :
PRINT"I WIN !!!    "
490 IF dealertotal < playertotal OR d
ealertotal >21 THEN LOCATE 12,12 : PE
N 3 : PRINT "YOU HAVE WON ";CHR$(163)
;bet*2 : money=money+bet*2
500 IF dealertotal = playertotal THEN
 LOCATE 15,12 : PRINT"* D R A W *   "
 : money=money+bet
510 IF money < 1 THEN GOTO 1030
520 FOR suite=0 TO 3 : FOR c=1 TO 13
530  pack(suite,c)=0        : REM new
 pack of cards
540 NEXT c : NEXT suite
550 LOCATE 18,23 : PEN 1 : PRINT " <
Press Space Bar >"
560 IF INKEY$="" THEN 560 ELSE CLS :
GOTO 140
570 REM ******* cards *******
580 PEN 1 : MOVE x,y-10 : DRAW x,y+10
0 : DRAW x+75,y+100 : DRAW x+75,y-10
: DRAW x,y-10 : TAG : MOVE x,y+90
590 suite=INT(RND*4) : c=INT(RND*13)+
1 : REM select random card
600 IF suite=0 THEN s=226 : REM clubs
610 IF suite=1 THEN s=227 : REM diamo
nds
620 IF suite=2 THEN s=228 : REM heart
s
630 IF suite=3 THEN s=229 : REM spade
s
640 IF pack(suite,c)=1 THEN 590 : REM
 card has been dealt
650 IF pack(suite,c)=0 THEN pack(suit
e,c)=1
660 IF c>10 THEN cardvalue=10 ELSE ca
rdvalue=c
670 IF player=1 THEN playertotal=play
ertotal+cardvalue : IF c=1 THEN playe
races=playeraces+1
680 IF player=0 THEN dealertotal=deal
ertotal+cardvalue : IF c=1 THEN deale
races=dealeraces+1
690 IF start=1 THEN startsuite=s : st
artcard=c : start=0 : RETURN
700 MOVE x+2,y+95 : PRINT card$(c);
710 IF c=10 THEN MOVE x+42,y+8 : PRIN
T card$(c);: ELSE MOVE x+58,y+8 : PRI
NT card$(c);
720 IF c=1 THEN MOVE x+32,y+50 : PRIN
T CHR$(s);: REM ace
730 IF c=2 THEN MOVE x+30,y+40 : PRIN
T CHR$(s);: MOVE x+30,y+60 : PRINT CH
```

```
R$(s);
740 IF c=3 THEN MOVE x+30,y+70 : PRIN
T CHR$(s);: MOVE x+30,y+50 : PRINT CH
R$(s);: MOVE x+30,y+30 : PRINT CHR$(s
);
750 IF c=4 THEN MOVE x+15,y+70 : GOSU
B 860 : MOVE x+15,y+30 : GOSUB 860
760 IF c=5 THEN MOVE x+15,y+70 : GOSU
B 860 : MOVE x+15,y+30 : GOSUB 860 :
MOVE x+32,y+50 : PRINT CHR$(s);
770 IF c=6 THEN MOVE x+15,y+70 : GOSU
B 860 : MOVE x+15,y+50 : GOSUB 860 :
MOVE x+15,y+30 : GOSUB 860
780 IF c=7 THEN MOVE x+15,y+70 : GOSU
B 860 : MOVE x+15,y+50 : GOSUB 860 :
MOVE x+15,y+30 : GOSUB 860 : MOVE x+3
2,y+50 : PRINT CHR$(s);
790 IF c=8 THEN MOVE x+15,y+70 : GOSU
B 860 : MOVE x+15,y+50 : GOSUB 860 :
MOVE x+15,y+32 : GOSUB 860 : MOVE x+3
2,y+40 : PRINT CHR$(s);: MOVE x+32,y+
60 : PRINT CHR$(s);
800 IF c=9 THEN MOVE x+15,y+75 : GOSU
B 860 : MOVE x+15,y+25 : GOSUB 860 :
MOVE x+15,y+58 : GOSUB 860 : MOVE x+1
5,y+42 : GOSUB 860 : MOVE x+32,y+50 :
 PRINT CHR$(s);
810 IF c=10 THEN MOVE x+15,y+75 : GOS
UB 860 : MOVE x+15,y+25 : GOSUB 860 :
 MOVE x+15,y+58 : GOSUB 860 : MOVE x+
15,y+42 : GOSUB 860 : MOVE x+32,y+40
: PRINT CHR$(s);: MOVE x+32,y+60 : PR
INT CHR$(s);
820 IF c=11 THEN MOVE x+15,y+45 : PRI
NT CHR$(141);CHR$(140);CHR$(142);: MO
VE x+20,y+45 : DRAW x+40,y+70 : DRAW
x+59,y+45 : MOVE x+32,y+55 : PRINT CH
R$(s);
830 IF c=12 THEN MOVE x+15,y+37 : PRI
NT CHR$(131);CHR$(131);CHR$(131);: MO
VE x+15,y+55 : PRINT CHR$(197);CHR$(2
38);CHR$(199);: MOVE x+32,y+70 : PRIN
T CHR$(s);
840 IF c=13 THEN MOVE x+15,y+35 : PRI
NT CHR$(131);CHR$(131);CHR$(131);: MO
VE x+15,y+50 : PRINT CHR$(203);CHR$(2
03);CHR$(203);: MOVE x+32,y+65 : PRIN
T CHR$(s);
850 TAGOFF : RETURN
860 PRINT CHR$(s);" ";CHR$(s);: RETUR
N
870 REM ***** computer voice ! *****
880 FOR t=1 TO LEN(a$) : PRINT MID$(a
$,t,1);
890  FOR delay=0 TO 30 : NEXT delay
900 SOUND 1,20+INT(RND*10),3
910 NEXT t : RETURN
920 REM ******* delay *******
930 FOR delay=0 TO 700 : NEXT delay :
 RETURN
940 REM ******* pontoon ! *******
950 LOCATE 13,12 : PEN 3 : PRINT "PON
TOON !!  BONUS ";CHR$(163);bet
960 money=money+bet
970 FOR loop=1 TO 3
980  FOR note=70 TO 30 STEP -10 : SOU
ND 1,note : SOUND 2,note+loop*5,7
990  NEXT note : SOUND 1,70
1000 NEXT loop
1010 FOR delay=0 TO 3000 : NEXT delay
1020 LOCATE 11,12 : PRINT SPC(25) : G
OSUB 920 : RETURN
1030 REM ***** run out of money *****
1040 FOR t=1 TO 3 : GOSUB 920 : NEXT
1050 CLS : LOCATE 5,12 : PRINT"You ha
ve run out of money !!"
1060 LOCATE 1,23 : GOSUB 920
1070 IF autodestruct$="ON" THEN CALL
0 : REM goodbye !!
1080 LOCATE 5,18 : PRINT"Another game
? Y/N";:INPUT answer$
1085 answer$=UPPER$(answer$)
1090 WHILE answer$<>"Y" AND answer$<>
 "N":WEND
1100 IF answer$="N" THEN END
1110 money = 10:INK 0,0: INK 2,18: GO
TO 120
```

Design and your sprint

In Part III of his series on machine code WADDILOVE presents a sprite routine an help you design your own images

SO far we have looked at how the Mode 0 screen memory is organised and by the end of the last article we had developed quite a powerful sprite print routine. This isn't the end of the story, however – far from it. There's still a long way to go.

This month I'll list the final sprite routine plus a sprite editor to enable you to design your own sprites.

The editor is in Basic, but for the machine code you'll need an assembler. RAW, the assembler listed in the July issue of *Computing with the Amstrad* is perfectly adequate for the short programs that we are writing.

If you typed in the examples last time, you'll have a simple print routine that is capable of printing any size multicoloured character at any screen address.

This routine simply stores the sprite data directly in the screen memory to display the character. This has the unfortunate side-effect of destroying whatever is already there, which is not at all sprite-like. We need to modify it slightly to overcome this.

First, though, let's have a look at this month's programs and see what they do. It will be easier to follow the explanation once you've seen the machine code running.

Program I is an example sprite to use with the machine code program. You'll need to enter and run this before you run Program II. It's a man walking his dog on the beach by the sea on a sunny day. If you don't believe me, just run it and see!

It was created using the sprite editor – Program III – and is two characters wide, three high, and uses 10 different pens and inks.

It's not the sort of sprite you would normally have, but it does show the speed and power of the sprite routine. When you've seen it, imagine trying to print it and move it around from Basic. I wouldn't even attempt it.

Save Program I after running it and enter Program II. To run the sprite demonstration enter:

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

and use the cursor keys to move the sprite round the screen. Pressing Escape at any time returns to Basic.

Take a look at the print routine and see how it differs from the one developed last time. It starts at &8087 in Program II and is in two parts.

The first erases the old sprite and starts at &8087. The second prints the sprite at the new position and is at &80AD. The second part is also used by the initialisation routine to put the character on the screen.

This final print routine uses the XOR method to print the sprite. Each item of sprite data is collected and eXclusively ORed with whatever is in the screen memory. loop2 between &8098 to &809D in Program II shows how this is done.

The big advantage of the XOR method is that when the sprite is moved the screen is restored to whatever it was before.

Mike Bibby's Bits and Bytes in the September issue explored the use of logical functions such as XOR. If you're not sure what it does, then it's well worth looking it up.

Basically, when two bytes are XORed they are compared bit for bit and the result set according to the following rules:

1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0

If the screen byte is &00 and the sprite data is &CC then XORing the two will give &CC. When the sprite is moved the data is again XORed with the screen memory and as the screen memory is now &CC, XORing it again with &CC results in &00, which is

```
10 REM PROGRAM I
20 MEMORY &7FFF
30 FOR i=0 TO 193
40 READ j:POKE &9000+i,j
50 NEXT
60 REM 1 man and his dog
70 DATA 24,8
80 DATA 16,48,48,240,176,48,192,192,4
8,48,112,240,240,48,48,192,48,48,240
90 DATA 207,218,176,96,96,48,48,229,7
1,71,176,48,96,48,48,101,207,207,48,4
8
100 DATA 48,48,48,101,206,207,48,48,4
8,48,48,48,207,154,48,48,48,48,48,48,
101
110 DATA 48,48,48,48,48,48,60,192,148
,56,48,48,48,52,60,104,60,60,48,48,15
120 DATA 60,60,60,60,60,45,15,15,60,3
0,44,60,30,45,15,15,60,30,40,60,30,45
130 DATA 10,15,60,30,44,60,30,45,15,1
5,60,30,40,60,30,45,15,5,207,30,44,60
140 DATA 79,143,135,5,252,30,40,60,27
,75,195,5,173,30,60,60,145,193,195,13
3
150 DATA 173,14,8,12,192,51,98,192,17
3,14,8,12,192,193,195,192,232,132,72,
12
160 DATA 192,193,193,192,232,132,72,1
2,192,193,193,192,232,132,72,12,192,1
93
170 DATA 193,64,232,196,200,204,192,1
93,193
```

Program I

d print g sprite

de gphics, ROLAND e anq sprite editor to

what it was before the sprite was printed.

Suppose that the screen byte is &0C and the data byte &CC. When these are XORed we get &C0 which is not what we want. All it means is that the colours will be upset if the sprite is placed on a background which isn't zero.

When &C0 is XORed again with &CC we get &0C back again so the background is restored.

If that's confused you, try printing a few characters on the screen before calling the sprite routine. Then move the sprite over them and you'll see what I mean.

The colour problem is easily solved by redefining the inks so that they produce the right colours. Look up the simple Basic sprites article in the April issue for more information.

Before calling *print*, the new address of the sprite is placed in *new* and the old address in *old*. Note that the instruction at &808B is LD HL,0 which is a three-byte instruction, and that *old* is set to &808C, the second byte of it, at the start of Program II.

This means that the two zero bytes in the instruction will be replaced by the old address of the sprite.

This saves time as LD HL,nn is faster than LD HL,(nn). The same technique is used for some of the other variables.

The size of the sprite in columns and rows is passed in H and L. The sprite designer will tell you the size.

There are also a couple of extra variables to be set. These are *olddata* and *newdata* which make it possible to animate characters by pointing *olddata* to sprite 1 and *newdata* to sprite 2. The sprite at the old position is XORed with *olddata* and *newdata* is placed at *new*.

We saw in the first two articles that the screen is made up of rows of pixels split into groups of eight – the lines that we LOCATE and PRINT on, and that the address of each row of pixels in a group is separated by &800.

The print routine holds the address of the sprite in the HL register pair so &800 is added to get the next row. This is accomplished by passing H to A, adding &8 and passing the result back to the H register. It's faster than loading the BC pair with &800 and adding HL and BC.

The sprite print routine is fairly straighforward. When writing it I decided to go for speed rather than compactness, so it may seem longwinded in places, but it's the fastest routine I've come up with so far.

> **‘. . . writing the sprite print routine, I went for speed rather than compactness – it may be longwinded but it's the fastest I've done so far’**

That's enough about the sprite print routine. Now we'll have a look at the routines to actually move the sprite.

There are four sections labelled *up*, *down*, *left* and *right*. Each section reads the keyboard and moves the sprite in the appropriate direction.

I'm not going to explain them all as they aren't that different from each other. I'll go through the *up* routine at &8039 which is the most complicated. If you can follow that, you can follow the others, hopefully.

One of the firmware calls to read the keyboard is at &BB1E. It expects the key number to be in the A register and returns with the Zero flag clear if the key was pressed.

The cursor up key is key 0 so the first instruction XOR A sets the A register to zero and is followed by a call to the firmware. The Zero flag is tested and if the key isn't being pressed control jumps to the next section.

The old address of the sprite is kept on the stack so this is POPped into HL. As there are two horizontal pixels per byte in Mode 0, the address is decremented by &1000, (2*&800). This is two pixel rows so that the sprite moves up the same amount it moves when travelling horizontally.

To subtract &1000 from HL all that is necessary is to subtract &10 from H by transferring it to the A register. A bonus is that a copy of the high byte of the address is left in A.

The screen memory starts at &C000 so &C0 is subtracted from A, the high byte of the sprite address.

If the Carry flag is set the new address must be off the top of the screen so a correction factor is added to HL.

This situation occurs between groups of eight pixels because of the way the Amstrad maps the screen memory.

The program jumps back to the start again to print the sprite at the new position. Note that there is a call to &BD19 just before the call to the print routine. This prevents flicker by waiting for frame flyback.

Program III is the sprite editor. It contains full instructions and is quite easy to use. Two sprites, up to two characters wide and three high, can be held by the designer at the same time to enable you to work out an animated sequence.

There are commands to mirror sprites vertically and horizontally and a sprite can also be rotated through 90 degrees.

When you have completed the design, the sprite data is printed out as data statements with line numbers. Either the data can be saved directly, or the program ended and the data statements entered using the copy key. Replace Program I with your own data.

● *That's all for this month. Have fun designing sprites while I get to work on some collision detection routines. See you soon . . .*

Program II

```
RAW Assembler V.3

Pass... 2               ORG &8000

8000:                   .new=&80AE
8000:                   .old=&808C
8000:                   .newdata=&80B1
8000:                   .olddata=&808F

                        initialise
8000:21 00 C0           LD HL,&C000
8003:E5                 PUSH HL
8004:22 8C 80           LD (old),HL
8007:22 AE 80           LD (new),HL
800A:2A 00 90           LD HL,(&9000)
800D:22 D4 80           LD (rows),HL
8010:21 02 90           LD HL,&9002
8013:22 B1 80           LD (newdata),HL
8016:CD AD 80           CALL put
8019:E1                 POP HL

801A:                   .start
801A:22 D6 80           LD (address),HL
801D:22 AE 80           LD (new),HL
8020:21 02 90           LD HL,&9002
8023:22 B1 80           LD (newdata),HL
8026:22 8F 80           LD (olddata),HL
8029:2A D4 80           LD HL,(rows)
802C:CD 19 BD           CALL &BD19
802F:CD 87 80           CALL print
8032:2A D6 80           LD HL,(address)
8035:22 8C 80           LD (old),HL
8038:E5                 PUSH HL

                        up
8039:AF                 XOR A
803A:CD 1E BB           CALL &BB1E
803D:28 10              JR Z,down
803F:E1                 POP HL
8040:7C                 LD A,H
8041:D6 10              SUB &10
8043:67                 LD H,A
8044:D6 C0              SUB &C0
8046:30 D2              JR NC,start
8048:01 B0 3F           LD BC,&3FB0
804B:09                 ADD HL,BC
804C:C3 1A 80           JP start

804F:                   .down
804F:3E 02              LD A,2
8051:CD 1E BB           CALL &BB1E
8054:28 0F              JR Z,left
8056:E1                 POP HL
8057:7C                 LD A,H
8058:C6 10              ADD A,&10
805A:67                 LD H,A
805B:D2 1A 80           JP NC,start
805E:01 50 C0           LD BC,&C050
8061:09                 ADD HL,BC
8062:C3 1A 80           JP start

8065:                   .left

8065:3E 08              LD A,8
8067:CD 1E BB           CALL &BB1E
806A:28 05              JR Z,right
806C:E1                 POP HL
806D:2B                 DEC HL
806E:C3 1A 80           JP start

8071:                   .right
8071:3E 01              LD A,1
8073:CD 1E BB           CALL &BB1E
8076:28 05              JR Z,escape
8078:E1                 POP HL
8079:23                 INC HL
807A:C3 1A 80           JP start

807D:                   .escape
807D:3E 42              LD A,66
807F:CD 1E BB           CALL &BB1E
8082:E1                 POP HL
8083:CA 1A 80           JP Z,start
8086:C9                 RET

8087:                   .print
8087:22 D4 80           LD (rows),HL
808A:45                 LD B,L
808B:21 00 00           LD HL,0
808E:11 00 00           LD DE,0
8091:F3                 DI
8092:                   .loop1
8092:C5                 PUSH BC
8093:3A D5 80           LD A,(columns)
8096:47                 LD B,A
8097:E5                 PUSH HL
8098:                   .loop2
8098:1A                 LD A,(DE)
8099:AE                 XOR (HL)
809A:77                 LD (HL),A
809B:23                 INC HL
809C:13                 INC DE
809D:10 F9              DJNZ loop2
809F:E1                 POP HL
80A0:7C                 LD A,H
80A1:C6 08              ADD A,&8
80A3:67                 LD H,A
80A4:30 04              JR NC,nextrow1
80A6:01 50 C0           LD BC,&C050
80A9:09                 ADD HL,BC
80AA:                   .nextrow1
80AA:C1                 POP BC
80AB:10 E5              DJNZ loop1
80AD:                   .put
80AD:21 00 00           LD HL,0
80B0:11 00 00           LD DE,0
80B3:3A D4 80           LD A,(rows)
80B6:47                 LD B,A
80B7:                   .loop3
80B7:C5                 PUSH BC
80B8:3A D5 80           LD A,(columns)
80BB:47                 LD B,A
80BC:E5                 PUSH HL
80BD:                   .loop4
80BD:1A                 LD A,(DE)
80BE:AE                 XOR (HL)
80BF:77                 LD (HL),A
80C0:23                 INC HL
80C1:13                 INC DE
80C2:10 F9              DJNZ loop4
80C4:E1                 POP HL
80C5:7C                 LD A,H
80C6:C6 08              ADD A,&8
80C8:67                 LD H,A
80C9:30 04              JR NC,nextrow2
80CB:01 50 C0           LD BC,&C050
80CE:09                 ADD HL,BC
80CF:                   .nextrow2
80CF:C1                 POP BC
80D0:10 E5              DJNZ loop3
80D2:FB                 EI
80D3:C9                 RET

80D4:                   .rows
80D4:00                 DEFB 0
80D5:                   .columns
80D5:00                 DEFB 0
80D6:                   .address
80D6:00 00              DEFW 0

80D8:                   END
```

Program III

```
10 REM  Sprite  Designer
20 REM  By R.A.Waddilove
30 REM(c)Computing With The Amstrad
40 MEMORY &8000
50 GOSUB 2080:REM instructions
60 GOSUB 140 :REM initialise
70 GOSUB 320:REM screen
80 GOSUB 570:REM design
90 MODE 1
100 CALL &BB4E:CALL &BC02:REM reset
110 SPEED KEY 25,2
120 END
130 REM --------------------------
140 REM Initialise variables+arrays
150 REM --------------------------
160 SPEED KEY 250,250
170 DEFINT a-z
180 FOR i=1 TO 6
190 READ a$
```

```
200 POKE &8000+i,VAL("&"+a$)
210 NEXT
220 gpaper=&8001
230 DATA DD,7E,00,C3,E4,BB
240 DIM sdata(15,23),sdata2(15,23),te
mp(15,23)
250 FOR i=0 TO 15
260 READ j:INK i,j
270 NEXT
280 DATA 0,18,6,24,2,8,20,26,15,16,7,
9,13,22,2,26
290 colour=1:sprite=1
300 RETURN
310 REM --------------------------
320 REM      Draw main screen
330 REM --------------------------
340 MODE 0:BORDER 4
350 MOVE 0,0:DRAW 0,398,14
360 DRAW 638,398:DRAW 638,0:DRAW 0,0
370 CALL gpaper,14:PLOT -10,-10,15
380 TAG:MOVE 40,385
390 PRINT " SPRITE DESIGNER ";
400 TAGOFF
410 CALL gpaper,0:PAPER 0:PEN 13
420 LOCATE 2,22:PRINT "Sprite:";
430 PRINT USING "#";sprite
440 LOCATE 2,24:PRINT "Pen:";
450 PEN colour
460 PRINT MID$(STR$(colour),2)
470 MOVE 32,350:DRAW 320,350,13
480 DRAW 320,140:DRAW 32,140
490 DRAW 32,350:MOVE 470,280
500 DRAW 555,280:DRAW 555,215
510 DRAW 470,215:DRAW 470,280
520 PLOT 36,342-16*8:PLOT 316,342-8*1
6
530 PLOT 36,342-8*8:PLOT 316,342-8*8
540 PLOT 51+8*16,348:PLOT 51+8*16,142
550 RETURN
560 REM --------------------------
570 REM    Sprite design
580 REM --------------------------
590 PEN 15

600 PRINT CHR$(30);CHR$(23);CHR$(1)
610 x=0:y=0:char$=""
620 WHILE INKEY$<>"":WEND
630 WHILE char$<>"E"
640 gpen=sdata(x,y):PLOT 50+x*16,338-
y*8,15:GOSUB 950:FOR i=0 TO 100:NEXT:
char$=UPPER$(INKEY$):GOSUB 950:PLOT 5
0+x*16,338-y*8,15
650 IF char$=CHR$(&E0) THEN GOSUB 102
0:sdata(x,y)=colour:gpen=colour:GOSUB
 950
660 IF char$="P" THEN colour=(colour+
1) MOD 16:PEN colour:LOCATE 6,24:PRIN
T MID$(STR$(colour),2);" ":PEN 15
670 IF char$="V" THEN GOSUB 1360
680 IF char$="H" THEN GOSUB 1520
690 IF char$="N" THEN GOSUB 1080
700 IF char$="R" THEN GOSUB 1690
710 IF char$="S" THEN GOSUB 1080
720 IF char$="I" THEN GOSUB 790
730 IF char$=CHR$(13) THEN GOSUB 2320
740 IF char$=CHR$(127) THEN IF sdata(
x,y) THEN gpen=sdata(x,y):GOSUB 950:s
data(x,y)=0
750 x=x-(INKEY(1)>-1 AND x<15)+(INKEY
(8)>-1 AND x>0):y=y-(INKEY(2)>-1 AND
y<23)+(INKEY(0)>-1 AND y>0)
760 WEND
770 RETURN
780 REM --------------------------
790 REM     Redefine pen
800 REM --------------------------
810 i=-1:j=-1
820 WHILE i=-1
830 LOCATE 2,20:INPUT "Pen ";i
840 IF i<0 OR i>15 THEN i=-1
850 WEND
860 LOCATE 2,20:PRINT SPC(18)
870 WHILE j=-1
880 LOCATE 2,20:LINE INPUT "Ink ?";a$
890 IF LEN(a$) THEN j=VAL(a$):IF j<0
OR j>26 THEN j=-1
900 LOCATE 2,20:PRINT SPC(18)
910 WEND
920 IF INSTR(a$,",") THEN k=VAL(MID$(
a$,INSTR(a$,",")+1)):INK i,j,k ELSE I
NK i,j
930 RETURN
940 REM --------------------------
950 REM    Plot a pixel
960 REM --------------------------
970 PLOT 480+x*4,271-y*2,gpen
980 TAG:MOVE 51+x*16,342-y*8
990 PRINT CHR$(129);:TAGOFF
1000 RETURN
1010 REM -------------------------
1020 REM  Check if pixel set
1030 REM -------------------------
1040 IF sdata(x,y)=0 THEN RETURN
1050 gpen=sdata(x,y):GOSUB 950
1060 RETURN
1070 REM -------------------------

1080 REM    New sprite
1090 REM -------------------------
1100 LOCATE 2,20:PRINT "New"
1110 FOR i=0 TO 15
1120 FOR j=0 TO 23
1130 sdata(i,j)=0
1140 NEXT
1150 NEXT
1160 GOSUB 1200
1170 LOCATE 2,20:PRINT SPC(18)
1180 RETURN
1190 REM -------------------------
1200 REM   Clear windows
1210 REM -------------------------
1220 ORIGIN 0,0,40,311,346,144:CLG 0
1230 ORIGIN 0,0,476,548,276,219:CLG
1240 ORIGIN 0,0,0,640,400,0
1250 RETURN
1260 REM -------------------------
1270 REM   Print sprite
1280 REM -------------------------
1290 FOR x=0 TO 15
1300 FOR y=0 TO 23
1310 IF sdata(x,y) THEN gpen=sdata(x,
y):GOSUB 950
1320 NEXT
1330 NEXT
1340 RETURN

1350 REM -------------------------
1360 REM    Vertical mirror
1370 REM -------------------------
1380 LOCATE 2,20:PRINT "Vertical mirr
or"
1390 i=x:j=y
1400 FOR x=0 TO 7
1410 FOR y=0 TO 23
1420 k=sdata(15-x,y)
1430 sdata(15-x,y)=sdata(x,y)
1440 sdata(x,y)=k
1450 NEXT
1460 NEXT
1470 GOSUB 1200:GOSUB 1270
1480 x=i:y=j
1490 LOCATE 2,20:PRINT SPC(18)
1500 RETURN

1510 REM -------------------------
1520 REM    Horizontal mirror
1530 REM -------------------------
1540 LOCATE 2,20
1550 PRINT "Horizontal mirror"
1560 i=x:j=y
1570 FOR x=0 TO 15
1580 FOR y=0 TO 11
1590 k=sdata(x,23-y)
1600 sdata(x,23-y)=sdata(x,y)
1610 sdata(x,y)=k
1620 NEXT
1630 NEXT
1640 GOSUB 1200:GOSUB 1270
1650 x=i:y=j
1660 LOCATE 2,20:PRINT SPC(18)
1670 RETURN
1680 REM -------------------------
1690 REM      Rotate
1700 REM -------------------------
1710 LOCATE 2,20:PRINT "Rotating..."
1720 i=x:j=y
1730 FOR y=0 TO 15
1740 FOR x=0 TO 15
```

```
1750 temp(15-y,x)=sdata(x,y)
1760 NEXT
1770 NEXT
1780 FOR x=0 TO 15
1790 FOR y=0 TO 23
1800 sdata(x,y)=temp(x,y)
1810 NEXT
1820 NEXT
1830 GOSUB 1200:GOSUB 1270
1840 x=i:y=j
1850 LOCATE 2,20:PRINT SPC(18)
1860 RETURN
1870 REM ------------------------
1880 REM      Swap
1890 REM ------------------------
1900 LOCATE 2,20
1910 PRINT "Swap sprites"
1920 i=x:j=y
1930 FOR x=0 TO 15
1940 FOR y=0 TO 23
1950 k=sdata(x,y)
1960 sdata(x,y)=sdata2(x,y)
1970 sdata2(x,y)=k
1980 NEXT
1990 NEXT
2000 IF sprite=1 THEN sprite=2 ELSE s
prite=1
2010 LOCATE 9,22
2020 PRINT USING "#";sprite
2030 GOSUB 1200:GOSUB 1270
2040 x=i:y=j
2050 LOCATE 2,20:PRINT SPC(18)
2060 RETURN
2070 REM ------------------------
2080 REM    Instructions
2090 REM ------------------------
2100 CALL &BC02:CALL &BB4E:REM reset
2110 MODE 1:BORDER 10:INK 0,4
2120 PAPER 3:PEN 1:LOCATE 12,2:PRINT
" SPRITE  DESIGNER "
2130 PAPER 0:PEN 2:PRINT:PRINT " Maxi
mum size=16x24 pixels (2x3 chars.)"
2140 PRINT:PRINT:PRINT " Use ";CHR$(2
4);" cursor keys ";CHR$(24);" to move
."
2150 PRINT:PRINT " ";CHR$(24);" COPY
";CHR$(24);" to plot a pixel."
2160 PRINT:PRINT " ";CHR$(24);" DEL "
;CHR$(24);" to erase."
2170 PRINT:PRINT " ";CHR$(24);" ENTER
 ";CHR$(24);" saves current sprite da
ta."
2180 PRINT:PRINT:PRINT " ";CHR$(24);"
P";CHR$(24);"...change pens"
2190 PRINT " ";CHR$(24);"S";CHR$(24);
"...swap sprites"
2200 PRINT " ";CHR$(24);"H";CHR$(24);
"...horizontal mirror"
2210 PRINT " ";CHR$(24);"V";CHR$(24);
"...vertical mirror"
2220 PRINT " ";CHR$(24);"R";CHR$(24);
"...rotate top 16x16 pixels"
2230 PRINT " ";CHR$(24);"I";CHR$(24);
"...to redefine ink"
2240 PRINT " ";CHR$(24);"E";CHR$(24);
"...end"
2250 PEN 1:LOCATE 14,24:PRINT "Press
space..."
2260 MOVE 0,0:DRAW 0,398,1:DRAW 638,3
98:DRAW 638,0:DRAW 0,0
2270 WHILE INKEY$<>"":WEND
2280 WHILE INKEY$<>" ":WEND
2290 CLS
2300 RETURN
2310 REM -------------------------
2320 REM Save
2330 k=0:FOR i=0 TO 15:FOR j=0 TO 15:
k=k+sdata(i,j):NEXT:NEXT
2340 IF k=0 THEN RETURN
2350 address=&C000+8*&50+15*4
2360 LOCATE 2,20:PRINT "Saving..."
2370 f=1:GOSUB 2770:y=j
2380 address=address+&800*(y MOD 8)+&
50*(y\8)
2390 f=-1:GOSUB 2770:rows=j-y
2400 f=1:GOSUB 2870:x=i-(i AND 1)
2410 address=address+(x\2)
2420 f=-1:GOSUB 2870:columns=(i-x)\2
2430 k=&9000
2440 POKE k,rows+1:POKE k+1,columns+1
:k=k+2
2450 FOR j=0 TO rows
2460 x=address+&800*(j MOD 8)+&50*(j\
8):IF x>&FFFF THEN x=x+&C050
2470 FOR i=0 TO columns
2480 POKE k,PEEK(x)
2490 k=k+1:x=x+1
2500 NEXT
2510 NEXT
2520 length=k-&9000
2530 WHILE INKEY$<>"":WEND:MODE 2:BOR
DER 0:PEN 1
2540 PRINT:INPUT "Sprite's name ";nam
e$
2550 IF name$="" THEN name$="char"
2560 l=9020:k=&9002:a$="9020 DATA "
2570 PRINT:PRINT "9000 REM ";name$
2580 PRINT "9010 REM Rows=";MID$(STR$
(rows+1),2);"/Columns=";MID$(STR$(col
umns+1),2)
2590 FOR y=0 TO rows
2600 FOR x=0 TO columns
2610 a$=a$+MID$(STR$(PEEK(k)),2)+",":
k=k+1
2620 IF LEN(a$)>75 OR (x=columns AND
y=rows) THEN PRINT LEFT$(a$,LEN(a$)-1
):l=l+10:a$=MID$(STR$(l),2)+" DATA "
2630 NEXT
2640 NEXT
2650 PRINT:PRINT "Save or End (S/E) ?
";
2660 a$="":WHILE a$<>"S" AND a$<>"E":
a$=UPPER$(INKEY$):WEND
2670 IF a$="E" THEN PRINT"End":SPEED
KEY 25,2:END ELSE PRINT CHR$(13);SPC(
20)CHR$(13);
2680 name$=LEFT$(name$,8)+".SPR"
2690 PRINT "Saving ";CHR$(34);name$;C
HR$(34);",b,&9000,&";HEX$(length):PRI
NT
2700 SAVE name$,b,&9000,length
2710 PRINT:PRINT "Press a key..."
2720 WHILE INKEY$<>"":WEND:WHILE INKE
Y$="":WEND
2730 MODE 0:GOSUB 320:GOSUB 1200:GOSU
B 1270
2740 x=0:y=0:PEN 15
2750 RETURN
2760 REM ---------------------
2770 REM get top/bottom
2780 j=-23*(f=-1):k=0
2790 WHILE k=0
2800 FOR i=0 TO 15
2810 k=k+sdata(i,j)
2820 NEXT
2830 j=j-f*(k=0)
2840 WEND
2850 RETURN
2860 REM ---------------------
2870 REM get left/right
2880 i=-15*(f=-1):k=0
2890 WHILE k=0
2900 FOR j=0 TO 23
2910 k=k+sdata(i,j)
2920 NEXT
2930 i=i-f*(k=0)
2940 WEND
2950 RETURN
```

SHEKHANA COMPUTER SERVICES

	RRP	Our Price		RRP	Our Price
Way of Exploding Fist	9.99	7.50	International Karate	6.50	4.99
Rocky Horror Show	8.95	6.75	Daley Toms Decathalon	8.95	6.75
Every Ones a Wally	9.95	7.50	World Series Baseball	8.95	6.75
A View to a Kill	10.99	8.50	Steve Davis Snooker	7.95	5.90
Frank Brunos Boxing	8.95	6.75	*Spy v Spy	9.95	7.50
Dark Star	7.95	6.40	Technician Ted	7.95	6.40
Jump Jet	9.95	7.95	Combat Lynx	8.95	6.95
Raid Over Moscow	9.95	7.95	Grand Prix Driver 2	8.95	6.75
Football Manager	7.95	6.40	Glen Hoddle Football	8.95	7.50
Sir Lancelot	6.95	5.50	Master of the Lamp	9.95	7.95
Fighter Pilot	8.95	6.95	Gremlins	9.95	7.50
Jet Set Willy	8.95	6.75	Arabian Knights	6.00	4.50
Dragontorc	8.95	6.95	Azimuth (Head Aligment)	8.95	7.50
The Hobbit	14.95	11.95	Hard Hat Mac	8.95	7.50
Handicap Golf	6.95	5.95	Warlords	6.00	4.50
Pyjamarama	8.95	6.75	Artist (CRL)	9.95	7.50
Robin of Sherwood	9.95	7.95	One On One	8.95	7.50
Beachhead	9.95	7.35	Starion	9.95	7.95
Rocco (Rocky)	8.95	7.50	Superpipeline II	8.95	7.50
Mordens Quest	6.95	5.95	Herbbys Dummy Run	9.95	7.50
Project Future	8.95	7.50	Realm of Impossibility	8.95	7.50
Sorcery	8.95	6.95	Red Moon	6.95	5.95
Alien 8	9.95	7.50	Android One	7.95	5.95
Airwolf	8.95	7.50	Android Two	7.95	5.95
Amehem	9.95	7.95	Tornado Low Level	7.95	5.95
Dun Darach	9.95	7.25	Highway Encounter	7.95	5.95
Hard Hat Mac	8.95	7.50	Forth (Interceptor)	12.99	9.99
Archon	10.95	9.99	International Basketball	5.99	4.99
Bridge Player (Kuma)	6.95	5.95	Elite		POA
*Shadowfire	9.95	7.50	3D Stunt Rider	8.95	7.50
3D Star Strike	6.95	5.50	Knight Lore	9.95	7.50
Hypersports	8.95	6.95	Red Arrows	9.95	7.50
Emerald Isle	6.95	5.75	Confuzion	6.95	5.95
Mini Office	5.95	4.60	Ashkeron	6.95	5.95
Scrabble	9.95	7.95	Battle for Midway	9.95	8.00
Monopoly	9.95	7.95	Tank Busters	7.95	6.50
*Lords of Midnight	8.95	6.95	Code Name Mat II	8.95	6.75
Hurricane Higgins Pool	8.95	7.50	Killer/Gorilla	9.95	7.99
Hurricane Higgins Snooker	8.95	7.50	Animated Strip Poker	8.95	7.25
H. Higgs Snooker (disk)	12.95	11.95	Any Level 9 Game	9.99	7.50
Fighter Pilot (disk)	12.95	11.95	Roland in Space	8.95	6.95
Roland in Space (disk)	12.95	11.95	Home Budget (Kuma)	19.95	18.00
Zen Assembler (Kuma)	19.95	18.00	Database (Kuma)	14.95	13.95
Master Chess	8.95	6.75	Quickshot II-Joystick	12.95	8.99

Cheques/P.O. To: **SCS (DEPT CAM3), 653 GREEN LANES N8 0QY LONDON (MAIL ORDER ADDRESS ONLY). Tel: 01-800 3156 (S.A.E. FOR LIST)**

Europe add £1.00 per tape. Elsewhere at cost.

Come and pick up your discount software direct from our shop at Marbles Shopping Centre Unit 11. 527-531 Oxford St. W1R 1DD. (1 min. from Marble Arch Tube Station). Open 7 days a week from 10am-7pm. Sunday 10am-6pm.

Please bring copy of advert to our shop for above discounts.

* Please ring to confirm availability.

Barclaycard/Access orders welcome

REPLICATING REALITY:

EXPLORING COMPUTER SIMULATIONS

A new, major work from top-selling author Tim Hartnell is now available, to give you power. Power to replicate reality in a way which can be quite uncanny. From microbes which fight it out in a closed environment for the right to survive, to a program which allows you to fly a particularly temperamental plane (FLIGHT SIMULATION), you'll find a wide variety of programs to allow you to experience the power of computer simulations in action.

Of course, there is much more to the book than just programs. As well as the ready-to-run material, you'll discover how easy it is to analyse real-life situations to create your own computer simulations. The book contains eighteen complete programs, including the massive FLIGHT SIMULATION one. Machine-specific listings are included for SPECTRUM+ and SPECTRUM, BBC MICRO, AMSTRAD, COMMODORE 64, all MSX machines, plus any machine furnished with *Microsoft* BASIC.

Author Tim Hartnell explains: "Now you can use your computer as a window on artificial reality. Fly a plane with FLIGHT SIMULATION. Control a ROBOT. Plan the STOCK MARKET. Take part in a GRAND PRIX classic race or participate in American GRIDIRON. And learn the secrets of feedback loops and casual diagrams to create genuine simulations of your own."

Replicating Reality: Exploring Computer Simulations by Tim Hartnell (ISBN 0 907563 80 5) is £7.95, and available from most book and computer stores, or buy directly from the publisher, post free:

Interface Publications Ltd., Dept CA,
9-11 Kensington High Street,
London W8 5NP
(Trade only: WHSD Leicester)

Interface

UNLOCK YOUR

AMSTRAD 464/664

AMSKEY

● Tape to tape backup utility (Must not be used to infringe copyright laws)

Multi colour display and so very user friendly. Gives full header details and choice of saving speed. Full on screen instructions and prompts. **FREE** Peep a code program, for finding the hidden words in adventure games, etc.

Tape £6.99

AMSHELP

● Transfer tape software to disc

Better than most, AMSHELP will even transfer ASCII files. Also automatically adds a relocator for those programs that overwrite the disc memory area. **FREE** Disc utility load and list protected programs etc.

Tape £6.99 Disc £10.99

AMSDISC1

● The most powerful Amstrad disc utility to date

Powerful CPM style utility with many features making easy work of disc file handling, including the unerasing of erased files. "No disc user should be without this utility" (Independent Amstrad review). **FREE** Disc utility load and list protected programs etc.

Disc £12.99

Prices include packing and postage
Overseas orders please add £1

INTERLOCK SERVICES LTD
123b Brent Street
London NW4
Tel: 01-202 8727

VISA

"...a jewel of Spectrum programming..."

(Popular Computing Weekly)

£8.50
INCL. VAT & P&P.

£7.50
INCL. VAT & P&P.

...now for AMSTRAD 464/664

- * Assembler & Monitor in one package
- * Many new features
- * Fully relocatable
- * Supplied on cassette — back up option to Disc
- * Fully Disc compatible
- * Plus much more

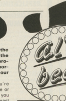
£19.95

PICTURESQUE

Fast Mail Order from:

Send SAE for full details

PICTURESQUE (Dept **C A**), 6 Corkscrew Hill, West Wickham, Kent



ALAN McLACHLAN lets his Smiley Hunt techniques loose to play havoc with the subs

MY little expedition over the last two months into the world of the games programmer has generated an enormous amount of interest from our readers.

It seems that whether you're young or old, an absolute novice or reasonably proficient, many of you want to have a go at writing at least one simple game in your computing career.

Smiley Hunt was very simple in concept and format – it had to be for me to write it – so I've decided to use the same basic ideas in another mega-production, and expand them slightly to introduce some different, perhaps more interesting routines.

The principle of the X, Y coordinate grid we'll retain, but in this game, so that we can incorporate some interesting graphics on-screen, we're going to create a variation called Sub Hunt.

I visualised a boat dropping depth charges to try and destroy submarines hidden from view beneath the sea's surface. We will move the boat left and right according to its X coordinate.

This is something we didn't do in Smiley Hunt, and will involve moving a graphics character across the screen. Not only will we incorporate a routine to' detect a key press to do this, we will also include one to detect a joystick as well.

The depth for the charge we will input from the keyboard just as we did with the Y coordinate in Smiley Hunt.

Before we start I want you to do something we didn't do last time that will make life a lot easier – set up the small Enter key to get you back into Mode 1 or Mode 2 for listing purposes. Type in, in direct mode, with no line number, substituting Mode 2 for Mode 1 if you prefer it:

```
KEY 139, "MODE 1:PAPER 0:PEN 1" +
CHR$(13) (Return)
```

All you have to do now is press the small Enter key after a trial run of the program and you'll be back in a decent mode for listing.

Right, let's get started. Once again the first thing to do is make a list of tasks that we want the program to carry out. I wrote them down on paper initially but to save time I've moved on from that and given each of these tasks a line number.

I've also inserted a GOSUB to the subroutine that will carry out the task, and supplied a short description of each.

Two WHILE . . . WEND loops have been added to control the game. The first creates a loop allowing you to play any number of games while a –1 or TRUE state exists – lines 40, 150. As everything is TRUE until we say it isn't, the game will go on until we choose to stop it – that is, the condition is FALSE.

The second loop, line 70, brings each game to an end if certain conditions are met. These conditions can be altered if you so desire, but in this version the end comes when either you have destroyed five submarines or used up 30 mines.

```
10 REM Al's sub hunt
20 GOSUB 160:REM title screen
30 GOSUB 200:REM initialise
40 WHILE -1
50 GOSUB 430:REM draw scene
60 GOSUB 540:REM hide subs
70 WHILE subs<5 AND mines<30
80 GOSUB 610:REM move boat
90 GOSUB 720:REM keyboard input
100 GOSUB 820:REM drop mine
110 GOSUB 950:REM check hits
120 WEND
130 GOSUB 1070:REM game over
140 GOSUB 1130:REM results
150 WEND
```

Listing I

Listing I is, in fact, the whole game in a nutshell. Granted the meat is missing, but all we have to do now is write each subroutine to carry out the task required of it.

We're going to have a title screen for this game and this is called at line 20. The subroutine consists simply of two FOR . . . NEXT loops to print out rows of asterisks, and two PRINT statements´ to put the title on the screen.

After "Al's" is printed the words "Sub Hunt" are put into the string *b$*. The MID$ command at line 220 then strips off each letter, one at a time, and prints it across the screen in a random colour accompanied by a simple scale of music. The notes for the scale are held in the DATA in line 1240.

I've also used this routine to dimension the array *sea(19,21)* for the invisible grid, and set the mode and various colours. The grid is to be used in exactly the same way as last time.

Type in Listings II and III then enter the dummy line 25:

```
25 GOTO 25
```

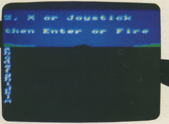
Run the program, and if you have typed in this routine correctly you

```
160 REM *** title ***
170 MODE 0:DIM sea(19,21):BORDER 0:PA
PER 5:CLS
180 PEN 3:FOR x=3 TO 17:LOCATE x,4:PR
INT "*":NEXT x
190 FOR x=3 TO 17:LOCATE x,17:PRINT "
*":NEXT x
200 LOCATE 9,8:PRINT "Al's"
210 a$="Sub Hunt"
220 FOR x= 1 TO LEN(a$):b$=MID$(a$,x,
1):PEN (RND)*3:LOCATE 2*x+1,12:PRINT
b$:READ pitch:SOUND 1,pitch
230 FOR delay =1 TO 500:NEXT delay
240 NEXT x
250 FOR delay = 1 TO 1500:NEXT delay
260 CLS:FOR delay = 1 TO 1500:NEXT de
lay
270 RETURN
```

Listing II

```
1230 REM *** music data ***
1240 DATA 119,106,95,89,80,71,63,60
```

Listing III



```
Z, X or Joystick
then Enter or Fire
a
b
c
d
e
f
g
h
i
j
k
l
m
```

The program so far: putting the subs in position

should see a title screen displaying "Al's Sub Hunt" between two rows of asterisks. The screen should clear to black after approximately three seconds.

You must now press Escape twice, then press the small Enter key. List the program, and if all's well we can add the next subroutine.

This is called by line 30 and initialises all the items that only need to be set up once in our game. In lines 290 to 340 I have defined some graphics characters to use in the screen display. These are the front and back of the ship, 240 and 241, joined together in line 350 as *boat$*, the depth charge, 242, the submarine 243, and the explosion, 244.

I have also used some of the Amstrad's existing character set to create a piece of land for each side of the screen – lines 360 and 370 – for obvious reasons called *rightland$* and *leftland$*.

You could experiment here and make up your own if you like, but be careful to use only four characters on each side otherwise the screen display will do some strange things and spoil the game.

The remainder of the routine sets up three sound envelopes and two text windows, WINDOW #0 cleared to a dark blue background for the sea, and WINDOW #1 cleared to light blue for the sky.

Type in Listing IV. As this subroutine only initialises certain items for later use there will be nothing to be seen by running the program at this moment.

The next routine, Listing V, will actually put something on the screen. That's why I've called it "draw scene". Line 440 holds the variables that need resetting each time the game is run. These are *mines*, the number of depth charges used, *subs*, the number of subs hit, and *oldboatx*, the start X coordinate of the ship.

Line 460 prints *boat$* at its first location *oldboatx* while lines 470 and 480 print *leftland$* then *rightland$* on their appropriate sides of the screen.

In Smiley Hunt we used the numbers 0-9 to indicate the 10 Y coordinates of the grid. Here we need more than 10, so to keep them to single characters we are going to use the letters a-m rather than numbers. This will avoid having to print two characters side by side.

Line 490 used a FOR ... NEXT loop to print these letters down the left hand side of the screen. It does

```
280 REM *** initialise ***
290 SYMBOL AFTER 240
300 SYMBOL 240,0,0,0,0,0,7,255,127
310 SYMBOL 241,0,0,0,0,200,240,255,25
2
320 SYMBOL 242,0,0,0,24,24,0,0,0
330 SYMBOL 243,0,0,16,24,126,255,126,
0
340 SYMBOL 244,17,128,4,80,8,34,40,13
2
350 boat$=CHR$(240)+CHR$(241)
360 rightland$=CHR$(95)+CHR$(210)+CHR
$(140)+CHR$(140)
370 leftland$=CHR$(140)+CHR$(210)+CHR
$(95)+CHR$(95)
380 ENT 1,200,2,1:ENV 1,15,-1,25:ENV
2,15,-1,10
390 BORDER 10
400 WINDOW #0,1,20,9,25:PAPER #0,0:CL
S #0
410 WINDOW #1,1,20,1,8:INK 8,11:PAPER
 #1,8:INK 15,24:CLS #1
420 RETURN
```

Listing IV

```
430 REM *** draw scene ***
440 mines=0:subs=0:oldboatx=10
450 CLS #0:CLS #1
460 PEN #1,5:LOCATE #1,oldboatx,8:PRI
NT #1,boat$
470 PEN #1,12:LOCATE #1,1,8:PRINT #1,
leftland$
480 LOCATE #1,17,8:PRINT #1,rightland
$;
490 FOR numy=1 TO 13:LOCATE #0,1,numy
:PEN #0,4:PRINT #0,CHR$(numy+96):NEXT
 numy
500 FOR locx=6 TO 10:FOR locy=1 TO 13
:sea(locx,locy)=0:NEXT locy:NEXT locx
510 PEN #1,15:LOCATE #1,1,2:PRINT #1,
"Z, X or Joystick"
520 LOCATE #1,1,5:PRINT #1,"then Ente
r or Fire"
530 RETURN
```

Listing V

this by adding 96 to the variable *numy* and printing the resultant Ascii character.

The first time through the loop *numy* is equal to 1, therefore we print CHR$(97) which is a lower case letter a. The next time through *numy* is 2, *numy* + 96 = 98, CHR$(98) is b, and so on to the letter m.

We initialise the grid so that every location contains a zero using the FOR . . . NEXT loop at line 500.

A couple of prompts on the screen at lines 510 and 520 will remind us at the start how to play the game, but the idea is they won't be seen again.

Now were ready to have a look at what's on the screen. Delete line 25 and enter dummy line 55:

```
55 GOTO 55
```

Run the program. If you have done everything right so far your screen should correspond exactly with the screen photograph. If it doesn't, you would be well advised to check your typing very carefully for mistakes before going any further.

Line 60 calls the next subroutine at line 540. This hides the submarines in exactly the same way that the Smileys were hidden last time.

A single FOR . . . NEXT loop selects 20 random numbers for both X and Y coordinates, then places a number 1 in the appropriate array element of *sea(sux,suby)*.

```
540 REM ### hide subs ###
550 FOR count=1 TO 20
560 subx=INT(RND*11)+5:suby=INT(RND*1
3)+1
570 IF sea(subx,suby)=1 THEN GOTO 560
580 sea(subx,suby)=1
590 NEXT count
600 RETURN
```

Listing VI

Line 560 sends the program back to the random number generator should an element already have a 1 in it. This ensures that there are always 20 separate submarines hidden away.

Now add line 585:

```
585 LOCATE #0,subx,suby:PRINT #0, "+"
```

remove line 55 and add dummy line 65:

```
65 GOTO 65
```

You will be able to see the locations of the submarines highlighted by the + sign. These can be left in until later as they will be extremely useful for testing out the detection routines.

This is an ideal place to break – it should give you time to ensure everything is working all right in time for next month.

If it isn't, don't be afraid to drop me a line at *Computing with the Amstrad.* But remember, if you want a personal reply – and this applies to anyone else writing in to us for whatever the reason – you must enclose a SAE.

We may not be able to help you, but at least we'll admit an honourable defeat.

LOOK

If you were a

COMPUTACLUB

member
you would be entitled to:

15% discount off all Amsoft Software
10% off KORES® Ribbons
10% off 1000 LABEL-TRACK® Computer Labels

3" MAXELL® Compact Disks **£39**

3M 5¼" Discs at **Trade Prices**

S/S D/D	**£14.45**	S/S Q/D	**£21.90**
D/S D/D	**£19.60**	D/S Q/D	**£25.35**

All prices include VAT and Postage

All orders despatched on day of receipt
SUBJECT TO STOCK AVAILABILITY

Plus SPECIAL OFFERS EVERY MONTH

Annual membership fee £11.50

For full details and an application form, write to

COMPUTATECH LTD
45 Plovers Mead, Wyatts Green, Essex CM15 0PS

PRINTERLAND

ORDERED TODAY - DELIVERED TOMORROW

UK's LOWEST PRINTER PRICES
FULL PRINTER SUPPORT FOR THE AMSTRAD
ORDERED TODAY – DELIVERED TOMORROW

		EX VAT	INC VAT
■ **DOT MATRIX**			
SHINWA CPA80		£165.00	£189.75
EPSON FX80		£314.00	£361.10
EPSON FX100		£425.00	£488.75
■ **NEAR LETTER QUALITY**			
KAGA KP-810		£235.00	£270.25
CANON 1080A	SPECIAL OFFER	£240.00	£276.00
KAGA P-910		£335.00	£385.25
CANON 1156A	SPECIAL OFFER	£335.00	£385.25
■ **DAISY WHEEL**			
DAISY STEP 2000		£225.00	£258.75
JUKI 6100		£325.00	£373.75
EPSON DX100		£356.00	£409.40
■ **AMSTRAD ADD ONS**			
TASWORD TWO		£15.60	£17.95
TASPRINT		£7.74	£8.90
TASCOPY		£7.74	£8.90
PRINTER CABLE		£11.00	£12.65
■ **COLOUR PRINTERS**			
EPSON JX-80	SPECIAL OFFER	£450.00	£517.50
■ **DISK DRIVES**			
AMSTRAD DDI-1		£160.00	£184.00

Educational, Government plus Overseas Orders Welcome. Delivery Printers (Securicor) £10.00

Printerland
Unit 27, Estate Buildings, Railway St, Huddersfield HD1 1JP
Showroom open Mon-Fri 9-6pm. Sat morn 9-1pm
Tel: Huddersfield (0484) 514105 or (0484) 687875

CAR CURE

CAR CURE will transform your micro into an **expert mechanic** to help you **find faults** on your car and then **determine the cure.** Technical Colleges buy **CAR CURE** to help students **learn** about the faults cars suffer, as well as their symptoms and cures. Now you too can benefit!
Complete with comprehensive **5000-word User Guide** and packaged in a long-life plastic wallet. Export orders get American-English cross reference. All for less than many garages charge for an hour's work!

Cassette for **Amstrad CPC464** and **48K Spectrum**, *(microdrive transfer instructions included free, on request)* at only £9.95.
Disc for **BBCB** *(state 40 or 80 track)* at £11.95

★ **REVIEWS**

"... works extremely well, a great deal of effort has gone into making it user friendly as well as functional ... a very worthwhile package"
-CRASH

"... an expert system ... incredibly easy to use"
-Popular Computing Weekly

"... well produced ... a worthwhile addition to your toolkit"
-ZX Computing

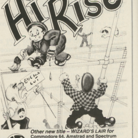
"... one of the most useful programs I have ever used. Makes 'xyz' look very childish and useless. A software company to keep an eye on"
-ZX Appeal (Canada)

★ **COMPREHENSIVE**
+ 900 faults
+ 300 symptoms

★ **POWERFUL**
+ Discover faulty parts for every symptom
+ Display faults for every part
+ Recommended cure for every fault
+ Trace symptoms of any fault
+ Identify faults causing two symptoms
+ Mark items for easy reference
+ Eliminate faults checked OK
+ Review progress on summary screen
+ Call help if in difficulty
+ Save diagnosis in data file

SIMTRON
PROGRAMS TO HELP YOU

DEPT CA, 4 CLARENCE DRIVE, EAST GRINSTEAD,
WEST SUSSEX RH19 4RX **Telephone (0342) 28188**



Other new title – WIZARD'S LAIR for
Commodore 64, Amstrad and Spectrum

bubble bus software
87 High Street, Tonbridge, Kent TN9 1RX
Telephone: 0732 355962 Telex: 95151

Rhythm is right, the typing wrong!

I AM a complete novice at computing, and slowly working my way through the Amstrad Teach Yourself Basic Tutorial Guide, Part 1.

On getting the July copy of Computing with the Amstrad, *I decided to try my first program – the Biorhythms Calendar.*

After a long time slowly typing it all in, I got a syntax error on running it in 780. I couldn't see any problem, and in desperation changed it to:

```
Zb = FNok(bd,bm,by)
```

I did a similar change in another line, and got the program to run, but the towers indicating the biorhythms are identical, whatever the month or birthday entered – that is, start and end of the month good, with a low around the 17-18-19th.

What's gone wrong? – **Lesley Tinker, St. Leonards-on-Sea.**

■ Functions can cause error messages to point to entirely the wrong line, especially if a function calls another function. This has misled a couple of readers.

The listing is correct, so it's simply a matter of looking for simple typing errors in your program.

In Daley's footsteps

ON Sunday, June 23, we got together to try to get through Daley Thompson's 10 gruelling events in Decathlon.

Our hopes rose when we got a good record in the 100 metres – 13.91 seconds. We qualified on the first four events, breaking a few personal records, but not on the high jump.

With two lives left, we got on to the pole vault, but failed to qualify.

On our last life, to our amazement, we qualified in the javelin.

With sweat pouring everywhere, we started the 1500 metres, taking it in turns to operate Daley Thompson.

Yes, we've done it. But where was the gold medal you should win like on the Spectrum version? It says "Go for Gold" on the pamphlet that you get inside.

But what we really write to ask is: Could you please tell us if we are the first trio to complete Daley Thompson's Decathlon? – **Scott Castleton, Nathan Haines, Martyn Bishop, Bristol.**

■ You're definitely the first we've heard of. Has anyone else managed it?

Double spacing

I HAVE a Canon PW-1080A printer which I primarily bought to use with a well-known word processor software utility.

Unfortunately I cannot get the printer to print in anything but double-spaced lines.

I was told that the software I purchased – Tasword464 – is compatible with all Epson-type printers, which I believe the Canon to be. – **A.H. Vaughan, Broadstairs, Kent.**

■ This is a common problem caused by the printer generating its own paper feed as well as the Amstrad sending one. There will be a dipswitch inside the printer to suppress its own paper feeds. Consult your printer manual for details.

Mod for mod

IN the June, 1985, edition of Computing with the Amstrad, *you published a letter from Peter Whiteley which proposed modifications to the Text Editor program in your February issue.*

Might I offer a further, minor, modification to Mr Whiteley's? I feel it might be of assistance to those who are not as conversant with keyboards as they could be – with myself the prime target, let me add.

Should the "P" option be exercised without a text file's being loaded, then the only escape is to press Escape and to rerun.

I would propose to amend Mr Whiteley's line number 2303 as follows:

```
2303 LOCATE 10,4:PRINT"Tota
l lines available for print
ing are ";(finish-here)/len
gth:IF (finish-here)/length
=0 THEN LOCATE 20,16:PRINT
" Insufficient text in file
 ":FOR delay=1 TO 3000:NEXT
:CLS:RETURN
```

This modification returns the user to the menu after the delay – which can be adapted to suit individual tastes.

Thank you for a superlative magazine. – **T.J. Wilson, Maidstone, Kent.**

■ We can't see how Roland's Text Editor can be improved on any more . . . unless you know different!

Loading CPC664

A NUMBER of people have written to us asking our advice on tape loading problems with the new Amstrad CPC664.

The problem is caused by the incorrect wiring up of the cassette lead so that the computer does not stop the tape between program parts and misses the start of the next block.

The situation is not helped by the fact that the CPC664 user instruction manual contains an error in Chapter 7 page 39 where the tape socket and stereo socket pin outs are reversed.

The cassette recorder should be the sort with a remote socket and the 5 pin Din plug should be wired as follows:

EAR . . . pins 2 & 4
MIC . . . pins 2 & 5
REM . . . pins 1 & 3

If Amstrad had provided a lead with the computer the problem would not exist. – **Steve Thomas, Pride Utilities, Chalton, Luton.**

Can you beat it?

ALONG with all the other games I have, are Sorcery and Ghostbusters.

My average score with Sorcery is 92,750 and in Ghostbusters my account is as follows: R.O.C. 46310231 – £999,900 plus 21 ghosts and £5,000 for closing the portal. This amount is, it seems, the maximum sum that can be reached. – **Rosemary Clarke, newport, Gwent.**

■ Has anyone beaten these?

Level 9 tips

WE have received many letters about Level 9 adventures on the new Amstrad CPC664.

They are not yet available on disc, but we have discovered that the cassette versions work on the CPC664 if you type the commands below (to get "|", press Shift

→

+ "@").

For Colossal, Quest, Dungeon, Snowball and Lords of Time adventures, type:

```
|TAPE:MEMORY &2FFF
```

and

```
LOAD "ADVENTURE", &3000
```

Start the tape and press any key. When the tape has loaded, type:

```
CALL &3000
```

For Eden, Emerald Isle and Red Moon adventures, type:

```
|TAPE
```

and

```
RUN ""
```

Start the tape and press any key whenever the computer asks you to.

Disc versions should be available later this year, and owners of cassette games will be able to upgrade to disc versions fairly cheaply. In the meantime, I hope this letter will be of interest to readers. – **Pete Austin, Level 9 Computing, Weston-super-Mare.**

■ Many thanks, Pete. Once again you've proved how user-friendly Level 9 is.

Heads you win

I MUST write in reply to T.J. Knowles of Worcester (Postbag, July 1985). I too bought the game Sorcery – from Virgin Games – and had problems with the program not loading or crashing.

When I took the tape back to the shop I was given a replacement without question.

However, the shop assistant advised me to make sure that the cassette heads were not dirty, as this could have an adverse effect on the loading of the program.

As with any other tape recorder, the heads become contaminated with debris from the magnetic tapes.

I took his advice and bought a head-cleaner cassette, and after using it have had no further trouble.

I have a query of my own. I am doing a course in computer programming and the language we are learning is Comal.

Has anyone produced a Comal package for the CPC464 and if so where can I get it? – **Patricia A. Towler, Skene, Aberdeenshire.**

■ We haven't come across a Comal package for the Amstrad. Do any of our readers know of one?

664 outdated

IN July 1984 I purchased an Amstrad CPC464 computer with colour monitor. I was very pleased and impressed with the machine and began to build up a large amount of software for it.

I joined the User Club and was happy with the way Amstrad quickly made available documentation, software and add-ons.

Having decided that a disc drive would be beneficial to the system, I began saving up for one. When the CPC664 was announced earlier this year, I did a few sums and worked out that it was more economical to sell my 464 and, together with the money saved for a disc drive, buy a new 664.

This I proceded to do, but having sold my 464, rumours began about a 6128. I decided to wait until the situation clarified somewhat.

Time passed slowly without a machine and I was getting impatient. However things seemed to become clearer. The 6128 was going to be for the US market. It wasn't likely to be launched in the UK until spring 1986.

Its price in the States implied a UK price of, perhaps £500-£600, which would be more than I really wanted to pay.

Based on these facts a decision was made. Five weeks ago I purchased a CPC664. It cost me nearly £450. It's a nice machine and I like it.

What I don't like is the sudden news that the CPC 6128 is going to be in the shops by the end of August at only £399 for the colour version!

What on earth is happening? How come the 664, only four months old, is superseded by a more advanced but cheaper machine?

It is very notable that the 464 was complemented by a more advanced and more expensive machine.

To say that I am annoyed would be an understatement. What is going to happen to the 664? Unless it is discontinued its price must obviously have to be considerably reduced.

Does this herald a new marketing philosophy by Amstrad? Are they going to emulate the Sinclair syndrome – look at them now!

It is this kind of customer alienation that caused a lot of the home computer industry's current problems.

It would be a nice gesture to appease irate 664 owners if some means could be offered to upgrade their machines to a 6128 specification. This could possibly be achieved by offering a 6128 keyboard at a very reasonable price and/or trade-in, as I understand the monitors would be compatible. – **Mr G. Winks, Nottingham.**

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

A faster version

WITH the Amstrad CPC464, is it possible to send a screenful of text to the printer?

The ZX81 has a single key "COPY"; the Genie II uses "SHIFT P", but no joy on the Amstrad.

Some of the utility/toolkit tapes include a screen dump, but I am hoping that a short Basic routine which will do the job might be available. – **H.C. Manning, Lisburn.**

■ A Basic dump may take up to half an hour! Try our machine code version in the March issue of *Computing with the Amstrad.* This is much faster.

Key to the lock

NEIL Carter asks in your August issue whether Caps Lock can be turned on or off from Basic programs. This can be done as follows:

```
POKE 46642,0     ON
POKE 46642,255   OFF
```

Also, Shift Lock (Ctrl+Caps Lock) can be similarly handled. The only difference is that the address to be POKEd is 46641. – **Stephen Silver, Leatherhead.**

Book on its way

REGARDING the series of articles on machine code by Mike Bibby, could you tell me if they are likely to be reprinted as a booklet?

I would welcome the chance of buying the complete series in one book. – **A.M. Barker, Fareham, Hants.**

■ Mike is anticipating completion of a book based on his machine code articles by Christmas. He informs us that it will contain much, much more than his monthly series, but we find it hard to imagine how.

TextEd problem

CAN you help me with a problem I am having with Roland Waddilove's TextEd program?

I cannot send the proper printer codes to my Riteman printer when using option "C".

For example, to put the printer into NLQ mode, I need to send ESC m 1.

The only way I can achieve this is to put the computer into direct mode and enter:

```
CHR$(27);"m";CHR$(1)
```

– **C.D. Gore, Kettering, Northants.**

■ It sounds as if there may be a bug in your listing. If you select option "C" you should be able to enter 27, 109, 1 and –1 to finish. If you can't, then check lines 2440-2540.

Dedit listing

I FEEL that I must inform you of a potential problem with the Dedit program published in the August edition.

While the program was excellent, and thanks should go to Ches Jeske, there is one very small problem – or very big depending on the number of corrupt discs – with the listing as given.

I would like to add that if the program is entered exactly *as given, then no problems exist, but I feel that few people will enter it* exactly.

The problem exists in line 500 – that is:

```
500 IF ASC(k$)=114 THEN
CALL &90B1 ELSE CALL &90C3
```

Well, all is OK if you have entered the program as given using lower case and have let the CPC convert all keywords to upper case. But if like myself other users normally enter routines in upper case, a serious problem shows itself, in that when you expect the CPC to read a disc it is actually writing to it, *corrupting any sectors it encounters.*

This is due to line 320 – that is:

```
320 k$="read":GOSUB 480:IF
PEEK(&90E3) <> 0 THEN 120
```

As can be seen in line 500, the Ascii value of the first character in k$ is checked against that of lower case 'r' that is 114.

If the value is indeed 114, then all is well and the code at &90B1 is called performing a disc read.

If however, the value returned is not 114 – as will occur if the listing is entered in upper case when the value will be 82 (upper case R) – then the routine at &90C3 is called "performing a disc write". A little expensive if the disc in question is the master system disc . . .

I know that this is really the user's fault for not entering exactly as shown. But being a seasoned programmer of some five years on over a dozen various machines, I have become used to entering programs in upper case, leaving lower case for word processing and so on.

I am also aware that once the program is entered as printed it works fine, but I feel that it may be of use to other users if you make this fact known as soon as practicable.

I would like to add that being used to computers I placed a write protected disc in the drive when I first ran the program – a point less-aware users should remember. Until you know it works BEWARE.

I hope there are not too many corrupted discs in users' drives. I do know of four other users who did just as I did, but did not use a protected disc, and suffered for it.

It isn't really worth adding an amendment to the program, just change line 320 to lower case and all should be well.

And if you still have problems, then you are suffering from the other great problem with computers – or should I say users. Finger trouble. *Believe me, the program works, and works well.* – **R.J. Taylor, Selston, Notts.**

Using printer plotters

FURTHER to George Usch's enquiry (Computing with the Amstrad, July 1985) regarding printer plotters, we are a group who have had considerable experience using printer plotters on the Amstrad.

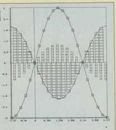
We have used both the MCP40 – 4.25in. roll paper – and the SCP800 – A4 size flatbed – of which there are various models.

Although printer plotters are mainly used for writing text – listing programs, and so on – they leave something to be desired compared with dot matrix printers.

Having said this, they certainly come into their own where there is the need for more graphical work – such as drawing lines, curves, circles – which is virtually impossible to achieve on a dot matrix printer, except for when doing a screen dump.

Both the aforementioned printer plotters are simple to connect to the Amstrad and are relatively simple to use. In fact the graphics instruction set is quite similar to the graphics commands in the Amstrad Basic.

Also in the graphics mode, both text and lines can be drawn so one can label drawings and so on, making it very versatile.



We have written a versatile program called Genplot which enables the user to create data sets and represent them graphically on the screen in the form of graphs, histograms, bar charts and high-low charts.

A hard copy can then be produced using one of the printer plotters mentioned above – see sample.

The MCP40 and SCP800 are available for around £125 and £175 respectively, making them a very attractive proposition for the home computer user, compared with other plotters available at £400-plus. – **David Lloyd, Al-Tukhaim Eng. Group, Nottingham.**

■ We hope this satisfies Mr Usch's needs, and are grateful for the response.

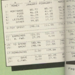
QUICK TO LEAR

THAT'S...

MINI OFFICE

FINALIST
BRITISH MICRO AWARDS
1985

SPREADSHEET



```
     A          B         C        D        E
1:  MONEY    JANUARY  FEBRUARY   MARCH    APRIL

2: MORTGAGE   85.72     85.72
3:     FOOD   46.24     41.47
4:     FUEL   46.25     47.28
5:  LEISURE   20.00     20.00
6:    OTHER   99.85     17.12

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
```

JUST LOOK WHAT THIS PACKAGE CAN DO!

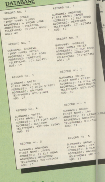
WORD PROCESSOR – Ideal for writing letters or reports! *Features:* Constant time display ● Constant word count (even shows words per minute) ● Normal or double-height text on screen or printout.

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.

DATABASE



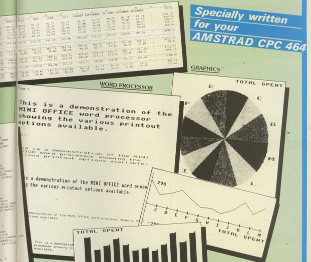
```
RECORD No. 1                         RECORD No. 1

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: 853-8092
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 LEAWA
ADDRESS2: GULLHAM                    ADDRESS2: NORWICH
TELEPHONE: 452-986 76543             TELEPHONE: 831-343
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: 681-
AGE: 13                              AGE: 11
```

...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	41.23	38.29	32.71	27.98	35.99	40.89	39.89	46.45		460.26
32.61	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		305.00
100.87	49.29	16.45	29.96	19.49	36.89	107.90	38.02	79.49		651.56
276.91	232.50	191.15	201.38	180.69	219.21	293.90	228.80	289.51		2858.14
321.21	353.31	353.31	353.31	353.31	353.31	353.31	353.31	353.31		4111.32
28.49	18.20	34.80	49.24	50.29	55.73	47.46	26.72	37.81		25.40
349.70	371.51	388.11	402.55	403.60	409.04	400.77	380.03	391.12		4136.72
276.91	232.30	191.15	201.38	180.69	219.21	293.90	228.80	289.51		2858.14
72.79	139.21	196.96	201.17	222.91	189.83	106.87	151.23	101.61		1278.58
54.59	104.40	147.72	150.88	167.18	142.37	80.15	113.42	76.21		958.94
18.20	34.80	49.24	50.29	55.73	47.46	26.72	37.81	125.40		319.65

WORD PROCESSOR

Page 1

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

e 1

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

s 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.

This is a demonstrat
processor showing th
available.

ROAD
RD
423451

TER
LF ROAD
FORD
-423451

VIETH
MILL ROAD
ARINGTON
53-80923

4

OWN
IAN
17 LEAWARD CLOSE
NORWICH
031-3430

o. 5

BROWN
ME: JIM
1 8 ELM R
: NANTWIC
E: 681-45

GRAPHICS

l at the unbelievable st £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.

CA10

Printers: CP-80 80cps Fric & Trac	£189.95
CPA-80 100cps Fric & Trac	£199.95
Amstrad Printer Cable 6'6"	£7.95
3" HITACHI Second disc drive complete	£99.95
Second drive cable only	£7.95
3" double sided discs for 2 **£8.95** – 10 **£42.95**	
Cassettes: C15's boxed with label & card 3 **£1.15** – 10 **£3.20** – 100 **£30.00**	
Dk'tronics Speech Synthesiser	£37.95
RS232 Serial Interface For CP464	£43.95
For CP664	£53.95
SIDEWAYS ROM BOX for four ROM's CPC464	£29.95
+ Connector for CPC664	£9.95
ROM's Super Power MAILING LIST ROM	£39.95
Super Power DISK UTILITY ROM	£39.95
Super Power PROGRAMMERS TOOL ROM	£39.95
Super Power WORD PROCESSOR ROM	TBA
Super Power DATABASE ROM	TBA
(Buy any three ROM's and deduct £5.00)	
DISC PAYROLL Contracted in or out (Weekly, 2 or 4 Weekly & Monthly)	£49.95

Prices include VAT. Postage & Packing add 50p

Micro-Aid (Amstrad).
25 Fore Street, Praze, Camborne, Cornwall.
or Tel: 0209 831274 with ACCESS

AMSTRAD CPC464
BAD NEWS

(For anyone who has bought a fast-load converter/header reader/de-protector for £5 or over.) A useful bit of software, but usually over-priced! We include one as just ONE of the four programs on our TOOLBOX cassette. Also included is an animated Sprite Designer, a Screen Designer program with dual-mode split-screen option, and a Monitor utility for entering M/C programs and examining the operating system. The price remains at ***£4.95.*** *"Excellent value"* – HCW review. *"A bargain"* – PCN review.

GRASP. Graph & Function plotter. With built-in trend calculations, arbitrary user-defined maths. Function calculation, auto or manual scaling, line graphs or bar/pie charts, interactive control of screen lay-out and colour. *"The range of charts is impressive – a good easy-to-use package"* – HCW review. *Price:* ***£8.50.***

Cheques/P.O.'s to:

**CAMEL MICROS, Wellpark, Willeys Avenue,
EXETER, Devon EX2 8BE**

SUNARO software

** STAR **		FRANK BRUNOS BOXING	7.50	** STAR **	
CHOICE		WAY OF THE EXPLODING FIST	8.50	**CHOICE**	
JUMP JET	8.50	GREMLINS	8.50	HI-SOFT 'C'	29.95
RED ARROWS	7.50	ALIEN	7.50	THE QUILL	14.95
KNIGHT LORE	8.50	RED MOON	5.95	TASPRINT	8.50
ALIEN 8	8.50	MINDER	8.50	TASWORD	16.95
BINKY	5.25	MORDENS QUEST	5.95	TASCOPY	8.50
ROCKY HORROR SHOW	7.50	MASTER OF THE LAMPS	9.50	FONT 464	6.75
BEACH HEAD	8.50	SCOUT STEPS OUT	7.50	AZIMUTH TAPE ALIGN	7.50
HARD HAT MACK	7.50	CONFUZION	5.95	USE IT	7.50
DUN DURACH	8.50	THE COVENANT	5.95	OPERATION CARETAKER	8.50
HIGGINS POOL	7.50	BRAXX BLUFF	7.50	SCREEN DESIGNER	10.95
HIGGINS SNOOKER	7.50	EVERYONES A WALLY	8.50	CONCISE BASIC	10.95
PSYCHEDLIA	5.25	ERIK THE VIKING	8.50	CONCISE FIRMWARE	17.95
FRANK N STEIN	7.50	THE WILD BUNCH	ANY	MINI OFFICE	5.25
MUTANT MONTY	7.00	MR FREEZE	TWO	DFM DATABASE	21.95
TECHNICIAN TED	6.00	SHORT FUZE	ONLY	MACHINE CODE TUTOR	12.75
WONGS LAUNDRY	7.00	SUB SUNK	4.75	TASWORD DISK	21.95

New titles always available on release
Prices include postage and VAT

**SUNARO SOFTWARE (CA)
PO Box 78, Macclesfield, Cheshire SK10 3PF**

USE IT – **Available for CPC 464
from Bubble Bus Software.
Three programs in one:**

GRAF IT – For 3D colour visual representation – bar chart, line graph and pie chart.

TIME IT – For a giant time piece – stop, lap, alarm. Analogue or digital.

CALC IT – For a giant calculator with binary and hexidecimal calculations.

Send cheque, access or. barclaycard to:

bubble bus software

**87 High Street, Tonbridge,
Kent TN9 1RX
Telephone: 0732 355962
Telex: 95151**

WIN UP TO £10,000 (or more)
Playing

THE ISLAND OF RIDDLES

Adventure Competition

Only £9.50 (inc p&p)

On Tape for Amstrad CPC464

Available by mail order from

**REDBEARD COMPUTERS
(Dept B), 45 Regent Place, Birmingham 1**
Trade and Overseas enquiries welcome

SIMPLE ACCOUNTS
AMSTRAD CPC464/664

Easy To Use, Comprehensive Book-Keeping System
** Used In Business Throughout The UK **

- Full Analysis of Income & Expenditure
- Powerful Entry Search Routines
- Monthly and Year to Date Reports
- Creditors/Debtors Report
- Quarterly VAT Report
- Printed or Screen Output

Only £29.95 Including VAT and Carriage (£34.95 on Disc)
Available from Approved Dealers or Direct from:

**CORNIX SOFTWARE LTD
16 Kneesworth Street, Royston, Herts. SG8 5AA
Tel: Royston (0763) 852989**
Full Specifications and Sample Output Available on Request

AMSDOC
A NEW TOOL THAT UNLOCKS THE LOGIC!

GET THE MOST benefit when you key in Basic programs! Tabulated explanations and listings without effort! AMSDOC-EXPLAIN lists programs with YOUR COMMENTS on variable and routine names alongside the original coding so you really know what is going on, and updates a support file for re-entry so you BUILD your understanding.

KEEP THE BEST easy automatic documentation of the programs you may want to change! AMSDOC-FLOW-LIST shows the statements IN THE ORDER THEY ACTUALLY RUN, notes each use of every variable and address, connects related routines by lines and references. AMSDOC-EXPLAIN listings safeguard your effort for the future.

SAVE YOUR COSTS of memory space, running time and backing store! Save effort in program writing! AMSDOC-TRIM saves space in coding, generates new shorter names for variables, full cross-reference/comparison list. AMSDOC-EXPLAIN changes specified variables. AMSDOC-COMPARE shows up differences between program versions.

FOUR PROGRAMS TO HELP YOU and your Amstrad 464 or 664. Printer optional. Listings to screen, printer or backup. Tape based system **£12.95.** Disc based system **£14.95** (including post and packing).

**Session Developments Limited
12 Falmouth Road, Congleton CW12 3BH**

AL-TUKHAIM

Micro Computer Centre

Minerva house, Spaniel Row, Nottingham NG1 6EP
Tel: (0602) 410502. Telex: 37369

Middle East Agents:
(T.M.C. Ibn Khaldon Street, PO Box 43265, Hawalli, Kuwait. Telex: 30282 TMC KT)
(Rahhal Electronic Establishment, Amman Street, Sueleh, Amman)

We are specialists in various aspects of the Micro-Computer Industry for Business, Education and Home Users. We can provide Hardware, Software, Peripherals, Interfaces, Accessories, Computer spares, Computer Furniture, Personal Robotics and Books, in fact you need look no further, we can fulfil your every Computer need.

We are specialists in the Exporting of all the above into various Middle East countries.

The products we support are:

COMPUTERS: Ericsson, Octopus, Wren, Amstrad, MSX Goldstar, Oric, BBC, Commodore 64.

PRINTERS: Brother Range, Epson Range

MONITORS: Microvitec Range, Fidelity Range

SOFTWARE: Business – Educational – Home – Systems, for various Micros

MISC: Tape duplicating/copying machine, software display racks

EXCITING NEWS FOR ALL AMSTRAD USERS
"GENPLOT" IS HERE!

GENPLOT is a general plotting package developed by experts for the AMSTRAD, it contains a sophisticated "DATA EDITOR" and a "VERSATILE PLOTTING FACILITY".

GENPLOT can also support a variety of Dot Matrix printers, also Pen-Plotters.

GENPLOT can help in the Business world in the preparation of Sales Projection Charts and Graphs, it can also be very useful to the Home User, as visual displays are far easier to read and comprehend than a screen full of crunchy numbers.

Please send me........copies of "**GENPLOT**" at £14.95 (tape) £29.95 (disk) incl. P&P

If you are charging your order to Access of Barclaycard please tick where necessary:

□ Access No. ______________ □ B/Card No. ______________

NAME ______________________________

ADDRESS ______________________________

____

POST CODE ______________ SIGN ______________

MONEY MANAGER

for the

AMSTRAD CPC464/664

Money Manager is an easy-to-use system for recording all financial transactions as a matter of record (unlike most 'budgeting' systems), and for analysing them in a number of very powerful ways in order to facilitate sound financial management. It is ideal for controlling the finances of a small business, or for users wishing to control their personal finances in a businesslike way.

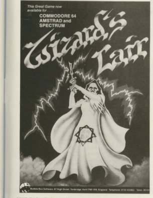
UNRIVALLED RANGE OF FEATURES:

- 50 user-defined income/expenditure classes, in logical groups.
- 9 user-defined accounts (bank, cash, credit card etc.)
- 12 months of data in memory.
- Automatic updating of starting balances when advancing months.
- 100 entries per month.
- Validated entry of date, account, reference, class, descriptive text, amount, user-defined flag.
- Easy editing of entries (also validated).
- Entries optionally sorted e.g. by date.
- Full selectivity of report categories by account, class and flag.
- Detailed statements for selected months.
- Summary tables (class vs month) fo whole year, with totals.
- Summary tables (class vs account) for whole year, with totals.
- Bar graphs (amount vs month) of 4 categories simultaneously.
- Pie charts (amount per year) of 10 categories simultaneously.
- Global search for any string.
- User-selected colours/shades.
- Drives any 80-column printer.
- Realistic set of demonstration/practice data.
- Comprehensive 12-page manual.

Cassette: ***£14.95*** *Disc:* ***£17.95*** *(incl. VAT, p&p)*

Connect Systems

3 Flanchford Road, London W12 9ND


This Great Game now available for . . .
COMMODORE 64
AMSTRAD and
SPECTRUM
Wizard's Lair
Bubble Bus Software, 87 High Street, Tonbridge, Kent TN9 1RX, England Telephone: 0732 355962 Telex: 95151

LOAD &RUN

Send your Cheques/POs (Sterling only) or stamp and address for list to:
LOAD & RUN, DEPT, CWA P.O. BOX 123
DARTFORD DA1 5AA

RING (0322) 73128 FOR ENQUIRIES

OUR POLICY FOR 1985 IS TO PROVIDE SOFTWARE THAT EVERYONE CAN AFFORD.

CASSETTE BASED GAMES

Item	RRP	Our Price
3 D Time Trek	£7.95	£6.95
Binky	£5.95	£4.45
Catastrophes	£8.95	£6.85
Chopper Squad	£5.95	£4.95
Execution	£6.95	£5.95
Fantastic Voyage	£8.95	£5.95
Gremlins	£9.95	£8.45
Master of the Lamps	£9.99	£8.95
The Scout Steps Out	£8.95	£6.85
Traffic	£8.95	£6.95
Hard Hat Mac	£9.95	£8.45
Way of Exploding Fist	£9.95	£8.45

ADVENTURE & STRATEGY

Item	RRP	Our Price
Alex Higgins Pool	£8.95	£6.95
Alex Higgins Snooker	£8.95	£6.95
Castle Black Star	£6.95	£5.95
Dun Darach	£9.95	£8.45
Jump Jet	£9.95	£8.95
Mordons Quest	£6.95	£5.95
Red Arrows	£8.95	£7.95
Red Coats	£6.95	£5.95
Red Moon	£6.95	£5.95
Ring of Darkness	£9.95	£8.95
Tripods	£11.50	£9.95
Zorakk	£6.95	£5.95

DISC ADVENTURE & STRATEGY

Item	RRP	Our Price
Alex Higgins Pool	£12.95	£10.15
Alex Higgins Snooker	£12.95	£10.15

HARDWARE

(In U.K. please remember to add the P&P in brackets)

Item	RRP	Our Price
CL1 (cas in'tfce lea	£3.95	£3.25
DL2 (664 d'drive lead)	£6.95	£5.85
LIGHT PEN (D'Ki)	£24.95	£21.95
DMP 1 PRINTER RIBBON	£6.50	£5.45
SSA1 SPEECH SYNTH. (£1)	£29.95	£25.95

TUTORIAL

Item	RRP	Our Price
Guide to Logo	£9.95	£8.95

EDUCATIONAL TAPES

Item	RRP	Our Price
Rhymeland	£5.95	£5.25
Here & There with Mr Men	£8.95	£7.95

DISC BASED UTILITIES

Item	RRP	Our Price
Font 64	£12.95	£11.55
Mastercalc	£29.95	£24.50
Pitman Typing Course	£13.95	£11.55

CASSETTE BASED UTILITIES

Item	RRP	Our Price
Abersoft 4th	£24.95	£21.95
Artwork (Kuma)	£9.95	£8.95
Mastercalc	£19.95	£17.45

UK: All prices include VAT and P&P in U.K. Add P&P in brackets for hardware.
For European & Overseas: All prices exclude VAT & include P&P in Europe.
Elsewhere add 50p extra for P&P hardware outside Europe add £2.50.

THIS IS ONLY A SELECTION OF NEW RELEASES AND RECENT ADDITIONS TO OUR STOCK. PLEASE RING FOR LATEST RELEASES
E.&O.E.

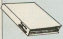
MAIL ORDER OFFERS — Computing with the Amstrad — MAIL ORDER OFFERS

Exclusive offer to subscribers!

It's been given rave reviews in Computing with the Amstrad and other magazines. Everyone who's played it says its one of the most gripping arcade games they've ever experienced. It's already taken America by storm. Now it's doing the samé in Britain. It's called . . .

BEACH-HEAD

– and it's a game no Amstrad owner will want to be without! It's in the shops now and costs **£9.95** on cassette. But we'll send it to you for just **£3** if you take out a subscription on the form opposite. *(Not available without subscription.)*

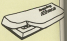


Binder: £3.95 (UK)

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.

Dust cover: £3.95

Protect your keyboard with our luxury dust cover made of soft, pliable, clear and water-resistant vinyl, bound with strong cotton and decorated with the magazine's logo.



GIVE YOUR FINGERS A REST!

JANUARY: *Smiley:* *Guide Smiley avoiding Grumpies.* **Code:** *Be a master-mind.* **Binary:** *Baffled by binary bits? Let our utility help you out.* **Dancer:** *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.* **Origins:** *Changing the graphics origin.*

MARCH: *Starfleet:* *Can you qualify as a starfleet pilot?* **Swamp:** *Join the battle against giant frogs.* **Parachute:** *Parachute to earth.* **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:** *A CAL language.* **Spelling:** *Learn to spell.* **Aleatoire:** *The odd ways mariners make decisions.* **Palindrome Tester:** *Slick string handling techniques.*

MAY: *Tronn Cycles:* *Two player arcade classic.* **Castle of Fear:** *A baffling fantasy adventure.* **Mouse:** *A simple but addictive game* **Cedric:** *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'.*

JUNE: *Easydraw:* *A utility to create and manipulate your own pictures.* **Da Bells:** *Hilarious arcade action.* **AMmon:** *A superb machine code monitor.* **Menu:** *Create colourful menus.* **Nim:** *Baffle your*

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 October 31
Please enter number required in box. £ p

Annual subscription

UK & Eire £12 (Sterling only)	6001	□
Overseas (Surface) £20	6003	□
Overseas (Airmail) £40	6004	□
Add £3 to your subscription and receive your cassette of Beach Head (worth £9.95). **Only available if accompanied by subscription order.**	6058	□

Commence with ________________ issue

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 ______

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	□
October (Le Mans)	6029	□
		□

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	□
October (Le Mans)	6069	□
		□

TOTAL ______

Back issues

£1.25 UK
£1.50 Overseas (Surface)

January	6006	□
February	6007	□
March	6008	□
April	6009	□
May	6010	□
June	6011	□
July	6012	□
August	6013	□
		□
		□

______

Binder

£3.95 UK
£5.00 Overseas

6038 □

TOTAL ______

Dust Cover

£3.95 UK
£4.45 Overseas

6037 □

TOTAL ______

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 ______

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/10

*All the program listings from each month's issue are available for only **£3.75** on cassette and **£6.75** on disc.*

friends. **Quads:** *Display expanding quadrilaterals.*

JULY: ***Missile Command:*** *Save earth from the invading hordes.* **Jumbo's Wordgame:** *Our Amstrad version of Odd One Out.* **Biorhythms:** *Find out if you're going to have "one of those days".* **Raw:** *Superb machine code assembler.* **Compacter:** *Reduce the memory used while saving a screen.*

AUGUST: ***Ah Bod:*** *Collect the eggs and avoid the evil Stompers.* **Dedit:** *Edit disc sectors and recover lost programs.* **Polygons:** *Create and display superb rotating shapes.* **Events:** *Demonstrations of interrupt routines.* **Password:** *Generate your own password.* **Graphics:** *A 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.*

OCTOBER: ***Le Mans:*** *Exciting 3D car racing spectacular. Be first across the line in this simulation of the 24 hour race.* **Pontoon:** *Win a fortune or lose your shirt in this superb version of the favourite gambling game.* **Bellringers:** *Take your lightpen bellringing with this entertaining music utility.* **PI:** *A program to calculate PI and leave the first mainframe's attempts standing.* **M/code Graphics:** *A sprite designer and print routine.* **Graphics:** *All the listings from this month's graphics article.*

Efficient, fast programs for small business

THE AWA Amstrad, already established as the only contender to Commodore in the home computer market, is now building itself a reputation as an excellent small-business machine.

New business software, which arrived this week at VieWest, should consolidate its position.

Camsoft is the umbrella title of some new business packages from a Welsh company called Cambrian Software Works. Programs include a sales ledger, a nominal ledger system, purchase ledger, payroll system, invoicing system, stock control and data base.

Complete

According to a report, one of the UK's most popular computer magazines listed all the user-friendly features required of business software and the Cambrian software incorporated every single item mentioned.

The Cambrian software seems so complete it leaves other small-business software at the starting post. It is so well thought-out and easy to use it is a credit to Cambrian and must grace the shelves of any small businessman with an Amstrad who takes his business seriously.

For speed and efficiency, the Camsoft packages are written in machine code. And although this is the best space-saving meth the programs st quire more

Camsoft gets highest rating

So this system can run a business package much more powerful than the capabilities of the machine.

In the file management sector of any of the packages, a record can be called by name. For example, if you do not know Fred's ac count number, you c call it up by typing F and the compu comes back with a of all Freds on the for you to choose fr The list is displayed alphabetical order so sorting is required.

All CamSoft system can be switched multi-user accessin where more than o computer or termin required to upda file.

Help

Although a user Amstrad might not power of ar work, it c and is ter b

If required, the password can be associated with any program tion from th menu of a can be

The Camsoft complete range of Business Software for the Amstrad CPC464 and CPC664

Includes:

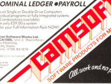
- ***DATABASE***
- ***INVOICING***
- ***STOCK CONTROL***
- ***SALES LEDGER***
- ***PURCHASE LEDGER***
- ***NOMINAL LEDGER*** ●***PAYROLL***

- Runs on Single or Double Drive Computers
- Individual programs or fully Integrated systems (26 Combinations available)
- From only £39.00 a system

Send for your Full Information Pack NOW!

Cambrian Software Works Ltd.
Dept CA1085, Unit 2, Maenofferen,
Blaenau Ffestiniog,
Gwynedd, Wales.
Telephone: 0766 831878
Telex
444102 BB CO G for Camsoft

A Member of the Berlyne Group of Companies



AMSTRAD IN EDUCATION

- LATEST AMSTRAD COMPUTER MODELS
- SUBSTANTIAL DISCOUNT AND EDUCATION SOFTWARE
- 1 YEAR FREE HARDWARE SERVICE CONTRACT
- £200 EDUCATION SOFTWARE PACK (SUPPLIED FREE)
- AMSTRAD BBCBASIC(Z80) LANGUAGE PACK
- AMSTRAD HARD DISC SYSTEM
- AMSTRAD LOCAL AREA NETWORK SYSTEM (FOR AMSTRAD, BBC, APPLE, APRICOT, IBM SYSTEMS)

Further details for Educational Establishments

Contact the Amstrad Educational Distributors:
NORTHERN COMPUTERS LTD
Churchfield Road, Frodsham, Cheshire WA6 6RD
Tel: (0928) 35700
Telex: 265871 (MONREF G) Quote: (UK)84:CXX123

ADVERTISERS INDEX

NOTICE TO ADVERTISERS

In view of legislation which may shortly be introduced, *Computing with the Amstrad* will not publish advertising for copiers or other equipment which may encourage copyright theft.

A UNIQUE PACKAGE

A SUITE OF PROGRAMS INCORPORATING:

AMSEM. A ROUTINE (modular) conditional assembler allowing the user to assemble very long programs in modules. References to external labels allowed. Output code is relocatable. Memory size does not limit size of source/object code.
AMSLINK. Links the modules created by the assembler to produce the executable code.
AMSLIB. (DISK ONLY) – Keeps often used modules in a library.
AMSMON. A MONITOR/DISASSEMBLER/DEBUGGING TOOL including symbolic disassembly and a single step debugger. Fully relocatable.

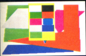
PRICE £14.95 Tape £21.95 Disk
(AMSMON available separately £7.95 or £11.95)

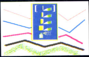
AM-MONOPOLY. Based on the popular board game. Plays standard rules. Superb graphics.
PRICE £4.95 Tape £9.45 Disk

AMSREC. A disk utility to automatically recover deleted files.
PRICE £9.95

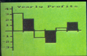
AMSRND. Provides random access files for the Basic user.
PRICE £9.95
BOTH FOR ONLY £14.95

Send s.a.e. for a full list to:
Anne Brooke
NIGHTINGALE SOFTWARE
29 Nightingale Road,
Southsea, Hants PO5 3JJ

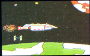

CHANGE COLOUR

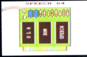

CHANGE BRUSH

ADD TEXT



GREEN SCREEN COMPATIBLE


HIGH-RES COLOUR GRAPHICS



TECHNICAL DRAWING

Compatible with: GT64 ● CTM640 ● MP1 ● DDI 1

Sophisticated graphics package on tape includes:
Colour palette ● 'nudge' control for one pixel accuracy ● brush choice ● text handling ● user defined characters ● magnify ● shrink ● circles ● rectangles ● lines ● curves ● colour fill ● tape & disc ● picture storage & retrieval ● pen calibration utility ● printer dump

GRAPHICS LIGHTPEN

For AMSTRAD CPC464

dk'tronics

Colour box artwork produced using lightpen, Amstrad CPC464 & colour monitor

Saffron Walden, Essex CB11 3AQ
Tel: (0799) 26350 10 lines

the first choice

Kuma AMSTRAD CPC464 software

A rapidly expanding range of Entertainments and Application Software for the Amstrad CPC 464 micro-computer featuring:

Draw pictures using sophisticated circle, ellipse, line and box drawing facilities with full use of all available screen modes and colours. This superb graphics program includes Fast Colour Fill, complete Screen Scroll (all directions) and the ability to save and load completed pictures to tape and disc.

Other titles include:

Star Avenger, Galaxia, Fruity Frank, Zen Assembler, Music Maestro, Rock Raid, North Sea Bullion, Stock Control, Datafile II.

Books:

- ***THE AMSTRAD CPC 464 EXPLORED*** *by John Braga*
- ***ZEN AND THE AMSTRAD CPC 464*** *by Ian R. Sinclair*

Visitors wishing to ca
at our Pangbourne Manufacturin
and Distribution Centre are advised t
phone 07357-4335 first for an early appointmen

Kuma Computers Ltd., Unit 12, Horseshoe Park,
Horseshoe Road, Pangbourne, Berks RG8 7JW.

Please send full catalogue on Amstrad CPC464 products.

Name ....
Address ....
.... Phone..............

I own an Amstrad CPC 464 computer □

Trade Enquiries Phone 07357-4335