POPULAR
Computing WEEKLY

3 June 1982 Vol 1 No 7

30p

The Black Hole

Reviewed inside:
Jungle Maths

Party Tricks

40 column board
for Vic-20

ZX81 character
codes

Spectrum colour

ANGLO AMERICAN SOFTWARE CO.

SOFTWARE SPECIALISTS

Starfleet Orion	PET 8K TRS-80 Level II 16K TRS-80 32K w/disk APPLE Integer BASIC 16K APPLE Integer BASIC w/disk 48K	**£19.50**
Invasion Orion	TRS-80 Level II 16K — ATARI 32K TRS-80 32K — APPLE 48K w/disk ATARI 32K w/disk	**£19.50**
Temple of Apshai	TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk ATARI 32K ATARI 32K w/disk IBM 64K w/disk	**£23.50**
Hellfire Warrior	TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk	**£23.50**
Upper Reaches of Apshai	TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk	**£15.50**
The Keys of Acheron	TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk	**£15.50**
Curse of Ra	TRS-80 32K — APPLE 48K w/disk TRS-80 Level II 16K	**£15.50**
Introductory 3-Pack **3-Pack:** Morloc's Tower, Rescue at Rigel and Datesones of Ryn	24K PET w/disk TRS-80 32K APPLE 48K w/disk	**£32.00**
Datestones of Ryn	TRS-80 Level II 16K — ATARI 32K TRS-80 32K — APPLE 48K w/disk ATARI 32K 2/disk	**£15.50**
Morloc's Tower	PET 24K TRS-80 Level II16K TRS-80 32K — APPLE 48K w/disk APPLE 32K	**£15.50**
Dragon's Eye	32K PET APPLE 48K w/disk	**£19.50**
Sorcerer of Siva	TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk	**£21.50**
Rescue at Rigel	TRS-80 Level II 16K — ATARI 32K TRS-80 32K — APPLE 48K w/disk ATARI 32K w/disk	**£21.50**
Star Warrior	TRS-80 Level II 16K — ATARI 32K TRS-80 32K — APPLE 48K w/disk ATARI 32K w/disk	**£23.50**
Crush, Crumble and Chomp!	TRS-80 Level II 16K — ATARI 32K TRS-80 32K — APPLE 48K w/disk ATARI 32K w/disk	**£21.50**
Tuesday Morning Quarterback	TRS-80 48K — APPLE 48K w/disk	**£21.50**
Richochet	ATARI 16K — TRS-80 Level II 16K TRS-80 32K — APPLE 48K w/disk ATARI 32K w/disk	**£19.95**
Jabbertalky	TRS-80 32K — APPLE 48K w/disk IBM 64K w/disk	**£21.50**

SEND 75p FOR FULL CATALOGUE
(Refundable against purchase)
DEALER ENQUIRIES WELCOME

ANGLO AMERICAN SOFTWARE
138a Stratford Road, Sparkhill
Birmingham B11 1AG 021-771 2996/2736

ALL PRICES INCLUDE VAT & POSTAGE
24 Hour answering service
on 021-771 2995 for Access orders.

PLEASE SUPPLY .....

.....

I enclose a cheque/PO for £..... made payable to ANGLO AMERICAN or debit my

Access and number .....

Signature .....

NAME .....

ADDRESS .....

.....

.....

POPULAR Computing WEEKLY

The Team

Editor
Duncan Scot

Reporter
David Kelly [01-930 3271]

Sub-editor
Peter Harvey

Designer
Eric Robbie

Editorial Secretary
Fiona McCormick

Advertisement Manager
David Lake [01-839 2846]

Advertisement Executive
Alastair Macintosh [01-930 3840]

Publishing Director
Nick Hampshire

Popular Computing Weekly,
Hobhouse Court, 19 Whitcomb Street,
London WC2
Telephone: **01-839 6835**

Published by Sunshine Publications Ltd.

Typesetting, origination and printing by Chesham Press, Chesham, Bucks

Distributed by S M Distribution
London SW9. 01-274 8611. Telex: 261643

Subscriptions
You can have *Popular Computing Weekly* sent to your home: the subscription rate is **£19.95** per year, for addresses in the UK.

How to submit articles
Articles which are submitted for publication should not be more than 1000 words long.

All submissions should be typed and a double space should be left between each line.

Programs should, whenever possible, be computer printed.

At present we cannot guarantee to return every submitted article, so please keep a copy.

Accuracy
Popular Computing Weekly cannot accept any responsibility for any errors in programs we publish, lthough we will always try our best to make sure programs work.

This Week

Cover illustration by Ian Craig

Editorial

Most schools in the country must know by now about the various Government microcomputer support schemes.

The most ambitious of the Government's aims is to see a microcomputer installed in every secondary school by the end of this year.

To this end it has been offering to pay half the cost of each computer bought. But most of the computers approved under the scheme are expensive and, by now, largely obsolete.

The other half of the Government's campaign is Information Technology Year '82, a project supposed to stimulate public awareness. But ITY seems to have died of inertia less than half-way through.

Many teachers must have realised by now that if they want to equip their pupils for a computing future they will have to do it on their own initiative.

The only way to keep up is to go out and buy a computer now. Schools cannot afford to wait for the Government and local education authorities to catch up.

Next Week

It's a drab, drab world till you add a little colour with your BBC Micro. Paint the town red, blue, yellow . . . the choice is yours!

Classified

20 SIMPLE ELECTRONIC PROJECTS FOR THE ZX81 and other computers

Make the most of your Microcomputer with this great book of construction projects. **£6.45.**

Available from:
National ZX Users' Club
44-46 Earls Court Road
London W8 6EJ

Please make your cheque payable to Interface.

ZX81 WITH 16K RAM for sale. All original packing, £75. Tel: 01-642 9190 days, 01-771 6814 evenings (Croydon).

ZX81 WITH 16K plus two tapes, only £79. Tel: Burgh Heath 57658.

VIC-20, cassette deck, super expander, super lander ROM software worth £16, only £235 unboxed. Tel: 993 2778, 6—8 pm.

CBM 32K 4008 and cassette soundbox, books, much software, £499 ono (neg). Tel: 051-428 1500 (Liverpool).

VIC-20 and VIC cassette plus £16 books, cost £270. Best offer secures. Tel: Bill 01-794 4436.

SHARP PC1211 POCKET COMPUTER plus printer and cassette interface with manuals, £80. Tel: 041-884 3404.

ZX81 EDUCATIONAL SOFTWARE. Brush up on your "O" level maths and French. Three (16K) programs of generated maths questions plus answers plus help. Three (16K) programs of French grammar (plus tests) plus three (16K) programs of vocabulary. Both cassettes £4.50 each or send sae for catalogue to: Rose Cassettes, 148 Widney Lane, Solihull, West Midlands B91 3LH.

ZX81 PLUS 64K MEMOTECH RAM plus tapes. Guarantee, £95. Tel:: 01-455 6296.

VIC-20 COMPUTER, cassette deck, programmers guide, Introduction Basic (1), joystick, software, £200. Tel: 01-969 6374.

ZX81, vgc, 16K, user port, tapes, Chess Defender etc. Only £100. Tel: 0630 85565.

ATOM USERS interesting new range of hardware for the Acorn Atom, please send sae for full details. Poundgate E.D.S., Beguildy, Knighton, Powys. Tel: 05477 273.

ZX81 16K, excellent condition, 5 months old, leads, tapes, manual and books £100. tel: Carlisle 60842 after 5 pm.

ZX81 16K VGC 3 months old and 2 tapes £100. Phone Worthing 46344.

BBC MICRO. M/C Invaders + Breakout and Dodgems. All new on model A or B using sound and colour. Three programs on one cassette £4 inclusive. From B. Cridland, 8 Bowness Avenue, Fleetwood, Lancs FY7 8PA.

ZX81 WITH 16K RAM, as new, plus four Sinclair tapes. Phone Davies 0403 60304 daytime.

FOR SALE. 4K ROM ZX80. First offer over £30 secures, including leads. P. Gower, 4 Clifford Crescent, Newport Gwent.

ZX81 16K RAM with all leads etc, cost £120 will accept £85. Phone 061-431 3800 evenings.

ZX81 16K PROGRAMS, 2 tapes, 4 games + 1, £3.75 each. SAE details, M. Fox, 6 Lawson Close, Aldridge, Walsall WS9 0RX.

ZX81 16K. PAYE TAX PROGRAM based on "Which" Tax Guide. Check last years tax; estimate this years tax and code; check salary tax, N.I., and superannuation deductions. Updatable at next budget. Extensive notes. Dr Wilkinson, 57 Osbaldwick Village, York YO1 3NP. £5.95.

ZX81 BLEEP. Provides audible feedback. Improves your keyboard cheaply, fits inside. Simple plug-in, no soldering connections. Aids faster more accurate programming, £8.95 includes instructions, P&P+VAT. Fulcrum Products, Dept. W, Hillside, Steep Lane, Findon, W. Sussex.

ZX81 16K COMPUTER, five months old, large collection of books and cassettes, £90. Tel: 024-369 4207 evenings.

VIC-20 for £150 including presentation cassette and booklet and lots more. 67 Worcester Street, Oldham, Lancs OL9 7SE.

MZ80K 48K, extended Basic, £335. Some software available. Tel: 01-505 4470 evenings.

ZX81 16K plus many program cassettes, chess, finance, adventures, console, £130. Tel: 021-533 6734.

ACORN ATOM, 12K+12K. Colour monitor, cassette, £350 ono. Tel: 0203-465311 ask for H217 (evenings only).

SINCLAIR ZX81 plus 16K RAM, vgc, five months old, £80 ono. Tel: Hastings 753873 after 5 pm.

ACORN ATOM, 8K ROM plus 12K RAM. Hardly used, superb condition, PSU, leads, etc, £190 ono. Tel: Wrington (STD 0934) 862231.

VIC-20. Six months guarantee left, £140 ono. White, Hillside, Totterdown Lane, Weston-super-Mare.

ZX81 16K. Complete, six months old. Tel: Kingston St Mary 449.

ZX81 16K plus seven software cassettes including chess, £75. Tel: 01-524 4368.

BBC COMPATIBLE COMPUTER to cassette leads. Type (1) two 3.5mm plugs and one 2.5mm plug. Type (2) 5-pin DIN, £4.50 each. — Electronics Applied, 4 Dromore Road, Carrickfergus, Co Antrim, BT38 7PJ.

ZX80, 16K RAM, 8 K ROM, constant display, proper keyboard, cased, books, tapes including defender and ZXAS assembler, £100. Luton 411340.

VIDEO GENIE COMPUTER, 3 months old, and programs, £190. 0272-653300 (Bristol).

ZX81! Two 16K games (cassette), excellent value, guaranteed. Includes "Destroyer", great displays! £5.00 (including p&p). P. Carr, "S-S-S", Tunbeck Road, Wortwell, Norfolk.

ZX81. 1K factory built, two months old, for sale. £50. Gebaard, East Grange, Morpeth, Northumberland. 0670 513337.

ATARI 400, cassette, joysticks, manuals, complete, as new, £280, Star Raiders £20.00. Phone 461 1604.

16K ZX 81 with keyboard, 2 software cassettes and 2 books, £100 ono. Ring 0895 31361.

ZX81 16K plus push button keyboard, learning lab, Startrek, loading and saving tip, £110.00. Tel: 0933 676495.

FREE CATALOGUE of quality ZX81 programs, many available as inexpensive listings. Send large SAE to Aquarius Software, 53 Towncourt Crescent, Petts Wood, Kent BR5 1PH.

SINCLAIR ZX81, 16K RAM, Abacus controller, 2 books, works perfectly, very good condition, £80 ono. Ipswich 713305.

ZX81 16K plus d'Ktronic Graphic Rom, 4 tapes £100. Tel: 021-745 8471

CLASSIFIED ADVERTISING RATES:

Line by line: For private individuals, 20p per word, minimum 10 words.
For companies, traders, and all commercial bodies, 40p per word, minimum 20 words.

Semi-display: £10 per single column centimetre, minimum length 3 cm. (Please supply A/W as PMT. Or supply rough setting instructions.)

Conditions: All copy for Classified section must be pre-paid. Cheques and postal orders should arrive at least two weeks before the publication date.
If you wish to discuss your ad, please ring Alastair Macintosh 01-930 3840.

Here's my classified ad.

(Please write your copy in capital letters on the lines below.)

		Please continue on a separate sheet of paper	

I make this ..... words, at ..... per word so I owe you £.....

Name .....

Address .....

.....

.....

Telephone .....

Please cut out and send this form to: Classified Department, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2

News

Other end of the Spectrum?

Micro APL, which launched its Spectrum microcomputer last September, is concerned about possible confusion between its product and the new Sinclair ZX Spectrum.

Micro APL did not register the name because it was advised that the name was too common to be accepted as a registered trade mark.

Now the company is getting enquiries from customers who are confusing the two machines.

The two Spectrums appear to have little in common. Sinclair's ZX Spectrum is, at £125, the lowest priced colour and sound machine. Micro APL's Spectrum is a 16-bit multi-user multi-task APL machine aimed mainly at the business market, with a basic price of around £10,000.

However, Micro APL is considering ways of clarifying the differences between the two micros. One solution would be to publish advertisements highlighting the facilities of the different Spectrums.

Micro APL emphasises that there are no hard feelings, and is in friendly communication with Sinclair Research.

EEC looks for new teletext

Interactive full-channel teletext is now a real possibility following EEC funding for a research study group.

Logica Ltd, of London, together with Italian General Systems, has been given £50,000 to look at the possibilities of an interactive teletext system using cable tv. The group will also investigate the problems involved in the development of a full-channel system.

The advent of satellite and cable tv makes possible the use of complete tv channels for teletext, instead of the eight spare lines of tv signal that are currently used.

In this way a much greater volume of information could be transmitted and, with cable tv, a return signal would be possible, enabling interactive teletext.

Visitors check out the prize-winning ZX81 at the Design Council.

Design Council picks the ZX81 for award

Sinclair's ZX81 is the first micro computer to win a Design Council Award.

Judges for the 1982 awards praised Sinclair for bringing computers within reach of the general public. The panel concluded: "The price and easy-to-follow instructions mean that every member of the family can have the opportunity to learn about computers and how they are programmed."

The award comes as Sinclair Research reports sales of over 20,000 units per week to America.

At the same time an exhibition of micros and their uses — called 'Inside Information' has been mounted jointly by the Design Council and Information Technology '82. At the Design Centre in London's Haymarket, it features many micros, including the ZX81, BBC Model B and the new Osbourne 1. The display concentrates, not only on the hardware, but also on the wide-ranging applications of micros — in the home and at work — and their use in, for example, medicine and telecommunications.

The exhibition runs until June 26 and entry is free.

Now: the fully equipped remote control household

Stripeland Electronic Control Systems have introduced a range of control units enabling micros to program the operation of domestic appliances by remote control.

The system comprises the user's own host-micro, one TX008 interface and up to 32 remote receiver units.

Instead of direct wiring from the TX008 unit to the appliance, which could be a tv, radio, lighting or even motorised curtains, the Stripeland system uses the existing mains lines.

Richard Last, of Stripeland, told *Popular Computing Weekly* that he will shortly be selling a two-way version of the system with built-in memory at the remote point. The remote device would then be able to store information and send it back to the micro.

Further details from Stripeland, 111 Liverpool Road, Formby, Merseyside L37 6BR.

Scotland gets first micro show

Edinburgh ZX Computer Club is to hold a one-day show on July 24. More than 30 stands are planned for this, the first micro show of its kind in Scotland.

Organiser Gordon Hewit told *Popular Computing Weekly* that the time was right for such a show.

The ZX fair will be at Meadowbank Stadium, open from 10am to 6pm.

Further details from Gordon Hewit, 3 Baberton Mains View, Edinburgh EH14 3BR.

Commodore show is on

Britain's only consumer weekly for micro owners, *Popular Computing Weekly* will be at the Cunard Hotel from June 3 to 5 for the 3rd International Commodore Computer Show. With twice the space of last year's show — over 30,000 sq ft on both exhibition levels of the hotel — there will be 154 stands on display

Displays will feature all the new Commodore products, including the Vic-10, Vic-30 and Commodore 64, plus the Vic networking system from Datalect and IT '82.

The show is to be opened by Commodore International's Chief Executive, Jack Tramiel.

The venue is the Cunard International Hotel, Hammersmith, London. Entry is £1 and the opening times are: June 3, 12am to 6pm; June 4, 10am to 6pm; and June 5, 10am to 5 pm.

You can find *Popular Computing Weekly* on Stand 140, on the lower floor.

IBM can't be too Personal

IBM still has no plans to introduce its Personal Computer to the UK.

The company now has an estimated backlog of 40,000 orders in the US and consequently no spare production to contemplate a UK launch.

Meanwhile, Mick Punter, managing director of Microcomputerland, has been importing the IBM micro.

Microcomputerland gets round IBM's export restrictions through its purchasing links with Computerland, an IBM US distributor.

However, Microcomputerland has reportedly been the subject of Fraud Squad enquiries and apparently a number of salesmen have been fired by Punter.

The IBM Personal Computer System is also being imported by KGB mircos.

Contact Microcomputerland, 1 Prince's Street, Richmond, Surrey, or KGB Micros, 14 Windsor Road, Slough SL1 2EJ.

Club Reports

Is your club involved in any special projects? Use this page to tell the world about it.

Three years on and it's still fun in Sunbury

David Kelly visits Sunbury-on-Thames Computer Club and talks to its founder

They're such a casual lot in Sunbury. Not for them the establishment rigours of membership fees, newsletters, and tutorial meetings. Though founder Simon Taylor originally planned that the club should take that sort of direction, the members unanimously decided against such formalities and instead created a regular weekly meet in the pub to offset the 'formality' of their monthly meeting in St Benedict's Church Hall in Ashford.

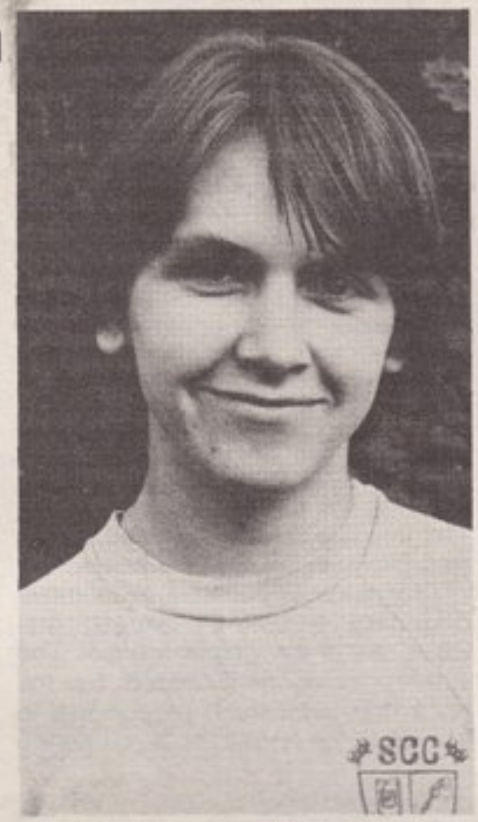
A lot has happened in the three years since the club was formed and Simon readily points out that it's been a long time in the world of micros.

Nowadays he's making a name for himself as creator of the game *Blitz*, which Commodore have contracted to market, and as a software programmer for Microgen and also writing programs for the new Sharp PC-1500. At 18 he's a budding expert quite naturally at home in one of the oldest computer clubs in Britain.

It all started just after he left school.

First he saved for a Mark 14, Sinclair's first micro, and began learning machine code. Then when a friend bought a Nascom I kit he decided to try to get in touch with other micro enthusiasts in and around Sunbury.

As so often happens, it was a letter in a magazine which really set the ball rolling. Simon got 10 letters and promptly organised meetings. He kept a list of names and addresses and every month someone would volunteer their house for the meeting. Everyone brought along their machine and exchanged ideas and programs.



Simon Taylor . . . a budding expert

They would meet on the first Friday of the month, bringing quite a variety of micros — Mark 14s, Nascoms, a Triton, Psycomp '80s, an Elector Junior and an Aim 65. Most of the machines operated only in machine code and if your micro understood a high-level language then that was really something!

Over the next eight months the club grew rapidly, with new members joining every month.

This arrangement ran into difficulties when more than 50 people, each with their machines turned up to the December 1980 meeting. The problem was no longer how to plug in all the micros but how to get all the members in through the front door! This was to be the last meeting of that type — just meeting in each other's homes was no longer practical.

The three people most involved in the running of the club at this time, Simon Taylor, Andy Lawrie and Stephen Battle, felt the club needed a formal set up — with membership fees, a newsletter and possibly lectures and tutorials.

They set out their ideas in a letter but the response from members suggested the most important feature of the club was its informality. To have a rigidly constituted group would be to destroy what the club stood for — a friendly meeting of people with a common interest. So it was decided to carry on, but to let the club, as far as possible, govern itself.

Over the next four months they held no meetings at all while Simon searched for a suitable monthly venue. At last he found St Benedict's Church Hall in Ashford and in April 1981 they met again for the first time in the hall.

Since then Sunbury Computer Club has met on the last Tuesday of every month and the air of informality is maintained. Simon keeps no list of names and addresses of those who attend and can only estimate that the membership is stable at somewhere around 60. In his own words "it is just a place where interested individuals can go and talk and exchange ideas."

He reckons that within the membership they now have at least two of every popular computer (except, strangely, the PET), and can provide help and advice on just about any machine.

The club also meets every week in the pub 50 yards from Simon's home — The Grey Horse.

As the club has developed, so has Simon's involvement in micro-computing. Together with Microgen he plans to produce a monthly cassette-based user club magazine, which should appear before the end of the summer.

All this, together with his full-time apprenticeship and his work for Sunbury Computer Club keeps Simon very busy — he admits he doesn't know where he finds the time. Simon's advice is never to forget the Sunbury Club's motto — Per ardua ad error!

Sunbury Computer Club meets at 8 pm in St Benedict's Church Hall, Napier Road, Ashford, on the last Tuesday of each month. The next hall meeting will be at 8 pm on June 29. On the other Tuesdays of each month the club meets for a drink and a chat at 8 pm in the Grey Horse, Staines Road East, Sunbury-on-Thames.

Further details from Simon Taylor, 8 Priory Close, Sunbury-on-Thames.

We want to hear from you!

Whether you are starting a new club, holding a special meeting, or just changing the venue, we want to hear from you.

Write to David Kelly, Club News, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF or call him on 01-930 3271.

For your diary

Norwich and District BBC Micro-Computer User Group meets twice-monthly, with workshops and talks, in Norwich City College. Contact Paul Beverley, Room B12a, Norwich City College. (Tel: 0603 60011 ext 233).

Mid-Cheshire Computer Club meets on the second Friday of each month in the main Winsford Library (in the Town Centre Precinct) at 7.30 pm. Contact Dave Clare, Providence House, 222 Townsfields Road, Winsford, Cheshire, CW7 4AX. (Tel: Winsford 51374.)

COVER STORY

Black

Learn to combat the greatest peril of space. By Dave Middleton

You are captain of a small spaceship carrying damaged androids to a repair ship and the quicker you get to the ship the larger your bonus. Your ship is fitted with only crude instruments which give your velocity components in the x and y directions of motion.

Long range scanning has already shown that there is a black hole in the area but because black holes do not emit light you obviously cannot see it. You will have to rely on gravitational effects on your velocity to fix its position on the screen.

Like any true space ship once you have accelerated to a velocity by giving thrust in one direction you have to thrust in the opposite direction to reduce velocity again.

If you move your ship out of the quadrant you are in, your on-board computer will advise you to use your warp drive and give you your current x,y position relative to the repair ship which is at co-ordinates 1,1; you still have control of your ship however and can manoeuvre using the normal controls. If you use the warp drive you will usually end up in the vicinity of the black hole but at least your ship will be back under control again.

How to get to the repair ship

The easiest way is to move outside quadrant, out of the gravitational effects of the black hole and fly by watching the x,y co-ordinates change. The average time I achieved for this was between 45 and 55 hours.

The more skillful way is to move diagonally across the screen and make use of the whiplash effect. The velocity component added by the black hole is inversely proportional to your distance from it, so if you go too close you will end up with a velocity which will either destroy your ship or fling you out at an uncontrollable speed. However if you get your approach correct your ship will be accelerated around the black hole into the vicinity of your repair ship. It is than a simple matter of decelerating and docking. The best time I achieved using this method was 20 hours. (You still have to spend some time outside the quadrant).

The game is in real time so you have to make your decisions quickly or another 'hour' will be added to your travel time.

To control your ship use the unshifted cursor keys. Pressing a key adds one velocity component in the direction the arrow is pointing, the key only works while information is being displayed. Positive x-velocity moves the ship to the right and positive y-velocity moves the ship down. Press 'w' if you are either too close to the black hole or you have lost control of your ship.

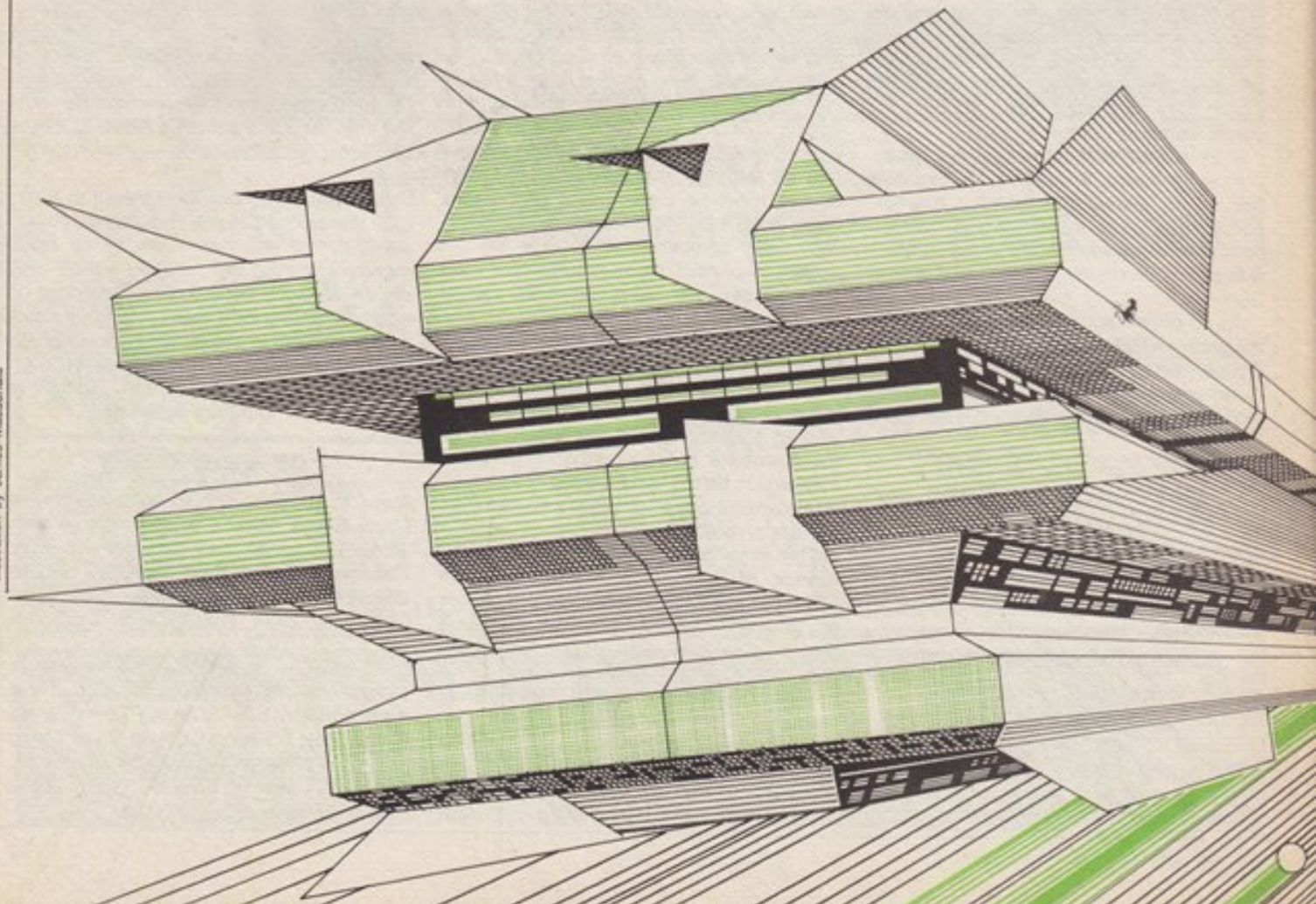
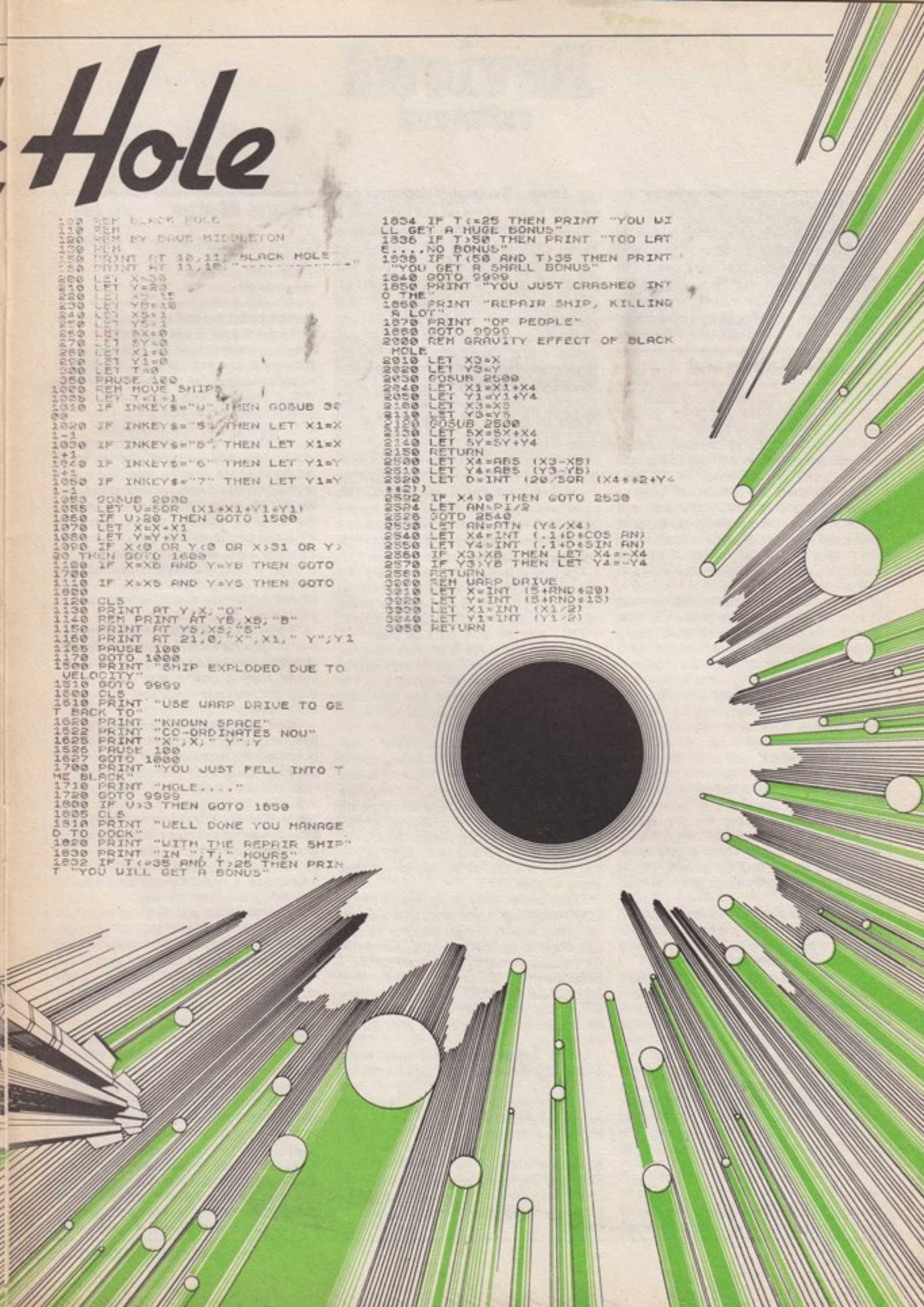


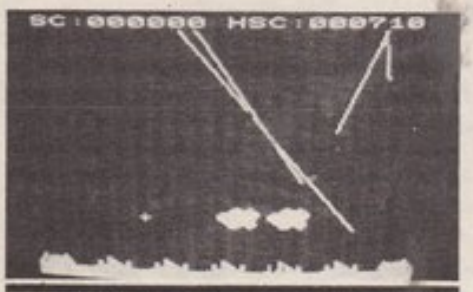
Illustration by James Macdonald

Hole

```
100 REM BLACK HOLE
110 REM
120 REM BY DAVE MIDDLETON
130 REM
150 PRINT AT 10,11;"BLACK HOLE"
160 PRINT AT 11,10;"----------"
200 LET X=20
210 LET Y=20
220 LET XB=15
230 LET YB=10
240 LET X5=1
250 LET Y5=1
260 LET BX=0
270 LET BY=0
280 LET X1=0
290 LET Y1=0
300 LET T=0
350 PAUSE 100
1000 REM MOVE SHIPS
1005 LET T=T+1
1010 IF INKEY$="U" THEN GOSUB 30
00
1020 IF INKEY$="5" THEN LET X1=X
1-1
1030 IF INKEY$="8" THEN LET X1=X
1+1
1040 IF INKEY$="6" THEN LET Y1=Y
1+1
1050 IF INKEY$="7" THEN LET Y1=Y
1-1
1053 GOSUB 2000
1055 LET V=SQR (X1*X1+Y1*Y1)
1060 IF V>20 THEN GOTO 1500
1070 LET X=X+X1
1080 LET Y=Y+Y1
1090 IF X<0 OR Y<0 OR X>31 OR Y>
20 THEN GOTO 1600
1100 IF X=XB AND Y=YB THEN GOTO
1700
1110 IF X=X5 AND Y=Y5 THEN GOTO
1800
1120 CLS
1130 PRINT AT Y,X;"O"
1140 REM PRINT AT YB,XB;"B"
1150 PRINT AT Y5,X5;"5"
1160 PRINT AT 21,0;"X",X1," Y";Y1
1165 PAUSE 100
1170 GOTO 1000
1500 PRINT "SHIP EXPLODED DUE TO
 VELOCITY"
1510 GOTO 9999
1600 CLS
1610 PRINT "USE WARP DRIVE TO GE
T BACK TO"
1620 PRINT "KNOWN SPACE"
1622 PRINT "CO-ORDINATES NOW"
1625 PRINT "X";X;" Y";Y
1626 PAUSE 100
1627 GOTO 1000
1700 PRINT "YOU JUST FELL INTO T
HE BLACK"
1710 PRINT "HOLE...."
1720 GOTO 9999
1800 IF V>3 THEN GOTO 1850
1805 CLS
1810 PRINT "WELL DONE YOU MANAGE
D TO DOCK"
1820 PRINT "WITH THE REPAIR SHIP"
1830 PRINT "IN ";T;" HOURS"
1832 IF T<=35 AND T>25 THEN PRIN
T "YOU WILL GET A BONUS"
1834 IF T<=25 THEN PRINT "YOU WI
LL GET A HUGE BONUS"
1836 IF T>50 THEN PRINT "TOO LAT
E....NO BONUS"
1838 IF T<50 AND T>35 THEN PRINT
 "YOU GET A SMALL BONUS"
1840 GOTO 9999
1850 PRINT "YOU JUST CRASHED INT
O THE"
1860 PRINT "REPAIR SHIP, KILLING
 A LOT"
1870 PRINT "OF PEOPLE"
1880 GOTO 9999
2000 REM GRAVITY EFFECT OF BLACK
 HOLE
2010 LET X3=X
2020 LET Y3=Y
2030 GOSUB 2500
2040 LET X1=X1+X4
2050 LET Y1=Y1+Y4
2100 LET X3=X5
2110 LET Y3=Y5
2120 GOSUB 2500
2130 LET SX=SX+X4
2140 LET SY=SY+Y4
2150 RETURN
2500 LET X4=ABS (X3-XB)
2510 LET Y4=ABS (Y3-YB)
2520 LET D=INT (20/SQR (X4**2+Y4
**2))
2522 IF X4>0 THEN GOTO 2530
2524 LET AN=PI/2
2526 GOTO 2540
2530 LET AN=ATN (Y4/X4)
2540 LET X4=INT (.1*D*COS AN)
2550 LET Y4=INT (.1*D*SIN AN)
2560 IF X3>XB THEN LET X4=-X4
2570 IF Y3>YB THEN LET Y4=-Y4
2580 RETURN
3000 REM WARP DRIVE
3010 LET X=INT (5+RND*20)
3020 LET Y=INT (5+RND*15)
3030 LET X1=INT (X1/2)
3040 LET Y1=INT (Y1/2)
3050 RETURN
```



Reviews
software



Missile Command

Available from Hi-Tech, or any Commodore Vic Dealer. Price £8.75.

All right, I admit it, I'm hooked! Another reproduction of a popular arcade game, this requires an additional 3K of RAM before the action can commence. It can be played using either the keyboard or a joystick, and a joystick is certainly to be preferred. Using the keyboard tends to get your fingers tied up in knots as you desperately try to press nineteen keys at the same time.

The same takes a while to load, as there is one setting up program before the main one comes in. Having selected keyboard or joystick control, sit back and wait for a few minutes while the second program is loaded. Once you've got there, you're in for a frantic time! You are defending five cities, which are under siege by missiles raining down from above. The method of defence is quite ingenious: you control a set of sights, which race about the screen at breakneck speed (they need to). When the sight is in an appropriate position, usually just in front of one of the missiles coming down, pressing the fire button launches a counter missile of your own, aimed at your sights. When it gets there, explosions occur, and the ensuing debris wipes out any enemy missiles which blunder into it.

However, any of the missiles which get through you defences are more than capable of wiping out a city if they score a direct hit, and when all your cities go . . .!

To score, you must demolish the enemy missiles, and use as few of your own missiles as possible, since you get points for any that remain after a particular attack wave is completed. The missiles come down in droves, and every wave gets successively more frantic, with seemingly hundreds pouring down at a time. There is a way of surviving this, which entails setting up a 'line' of your own missiles across the bottom of the screen, and hoping that the enemy missiles run out before your own do.

Summary
An extremely addictive game, and a fairly good reproduction of the existing arcade game. This is a difficult one for manufacturers to tackle, as the original game had a very novel way of moving your sights across the screen, and one which is not reproducible on any microcomputer. The efforts that Hi-Tech have put into being as faithful to the original as they could, considering the limitations imposed upon them, are commendable. A very good game. **PG**

Party Tricks

Video Software, Stone Lane, Kinver, West Midlands. ZX81 1K cassette,
Price £4.95.

So Video Software, long a supplier of sophisticated up-market 16K software for small businesses and training, is lowering its sights! They have just launched this cassette, containing ten BASIC 1K routines and promise more.

The kind of party they're aiming at is for children, I think. Introducing something different for children's parties is a major cause of ulcers in certain circles — to have a ZX81 play session can't be bad.

However, there are doubtless plenty of adults who will enjoy much of the material here, even if the overall novelty is not in the same league as that of, say, the Orwin packages or the "adult" games from Automata.

Video Software shows rather poor marketing in putting the least exciting programs first. We have *Shoot* — where you are taking a penalty which the goalie has to try to save; *Sketch* — differing from the million other etch-a-sketches only by having a SAVE facility; *Name The Day* — giving weekday for any date; and *Train* — you drive a train, in forward or reverse, along a track.

These are all fairly good, even if hardly world-shattering. Later programs are better. They include *Onger-Wonger* (a picture-drawing routine); *Weather* (a variant on the random poetry genre); *UFO* (shoot down the single space invader); *Who Shot JR?* (ZX81 Cluedo); *Field-Gun* (a nice target practise game); and *Follow* (you must copy the micro's wiggly path across the screen.

Video Software's well-known high-quality of presentation is used with this 1K package.

For your money you get a good cassette, with a set of saves on each side, and an impressive 26-page booklet.

Summary
Near top marks for this collection of 1K ZX81 programs — fun for all, and useful for those struggling to get into Sinclair BASIC. **KJL**

Jungle Maths

Scisoft, 5 Minster Gardens, Newthorpe, Eastwood, Notts.
ZX81 16K cassette.
Price £4.50.

It was with a great thrill of anticipation that I prepared to look at this package. There is not much material yet for ZX81 teaching, and a huge need. Scisoft's material is nicely packaged, not too costly and comes with a separate four-page leaflet.

But the thrill soon died down. The leaflet has been hastily and poorly written — there are ten grammatical errors in the eleven-sentence description of the material, for instance. And the recording quality on the cassette is very reminiscent of what we had to put up with nine months ago — the signal on one side was so weak that loading was impossible; that on the other was not quite as bad, but still bad.

Such lack of attention to detail is bad practice in any kind of software. It is not at all excusable where children's learning is concerned.

The ha'p'orths of tar are missing throughout the program that I loaded as well — slow reaction to key presses, punctuation marks missing, poor screen layout generally, inappropriate language, inadequate mugtrapping.

There are matters too that any competent maths teacher would frown on severely: inadequate restrictions on the questions posed, use of the "less than" symbol, incorrect use of the word "decimal".

All this is a great pity. Scisoft had a lovely idea — why, oh why, didn't they carry it right to the end before rushing to the market place?

The lovely idea is of course to link the educationist's need for drill programs to the child's need for games.

In *Jungle Maths* you must move across the screen avoiding a (rather strange) collection of hazards by answering the posed questions corrctly.

Type of question (from the four rules), range and type of numbers involved, and time limit may all be selected during the initial stages.

The hazards involve rather laborious moving graphics, but the children enjoyed them (despite their horrific nature) — they probably can't readily be improved without machine code. All the same, tar needs application to the BASIC coding generally.

Summary
An excellent idea — parents and teachers need this kind of program. It is a great pity that the authors haven't worked a bit longer on the coding: it is to be hoped that a polished version will soon appear. **KJ**

Reviews
hardware

Beebox

Available from Beelines Limited. Price £220 plus VAT.

This neat, compact unit, which sits underneath your Vic, is designed to give you a 40 column by 25 line display, and increase the amount of available memory from 3K to 32K. It connects up to the expansion socket on the Vic, but has a further socket of its own, so nothing is lost and quite a bit gained.

All this sounds very impressive, but is it as good as it's cracked up to be? The company's description creates the impression that a true 40 × 25 screen area is available on which you could merrily program impressive graphics for anything from arcade games to word processing packages. Not so.

On connecting up the unit and powering it all up, you have lost the traditional 'window' screen display of the Vic, and the whole screen is there for you to marvel at. This is somewhat reminiscent of the old Commodore Pet 4032 display. The old Vic appearance is not all you've lost however. Also gone are the Vic graphics, and indeed just about everything you've become familiar with. In their place is the Prestel character set, which in its defence is quite impressive, and a variety of other control characters. For some reason best known to Beelines, actioning these involves positioning them on the screen, thus losing some of the much valued 40 × 25 area.

These control characters include a facility to produce double height characters, flashing characters, and so on. What you can't do is alter the background colouring: you start off with white on a black background, and black it will remain, whatever you try and do about it.

One further unfortunate feature is that you cannot revert to the ordinary Vic screen, once that board is wired up. To get back into Vic mode you have to disconnect everything and start again. It would have been nice to be able to swap from one to the other at will.

On the plus side, the colour quality looks distinctly better than on a normal Vic, although there is a slight shimmer when scrolling through a listing.

Summary

It does give you an extra amount of memory, and all told is probably fairly cheap for an additional 29K of RAM and a 40 × 25 display area, even if it is only a display area. At £220 it will probably be of most use to the businessman who wants to use a larger area (for say stock control, or whatever), but for the average hobbyist I would say that it's a waste of time. **PG**

20 Simple Projects

By Stephen Adams, published by Interface, 44-46 Earls Court Road, London W8 Price £6.45.

This is the latest offering from the Interface publishing house that specialises in books for home computer users, in particular Sinclair users.

Author Stephen Adams is well-known in the microcomputer world as the man who knows his way around Sinclair hardware, and who has had much of his work published in the microcomputer press. The idea of this book is a good one, microcomputer hardware projects being a subject not often covered in magazines.

Adams makes no claims about the quality of his projects, each being the cheapest and simplest way of performing a task, not necessarily the best. For this reason the book deserves praise as a source of ideas or questions, rather than answers. Adams has provided you with the route, it's up to you to explore it.

Some of the projects are specific to the ZX81, but not so many that other computer users will feel left out. The circuit diagrams use a non-standard series of symbols, but are clear enough. As in other Interface books, there are a lot of pictures that have no relationship with the text — an interesting quirk. One other point I find most annoying is the liberal use of upper case letters throughout the book.

It should be remembered that books of this nature are not judged by the quality of their production, but by the information they contain, and this book contains a fair amount of that. The construction projects contain a minimum of software, concentrating on the electronics.

Among them are a tape recorder control, which every computer user will find handy, a light pen, which is the ultimate in low-technology, and an analogue to digital converter. **SB**

ZX81 EPROM board

EPROM Services, 3 Wedgewood Drive, Leeds LS8 1EF. Tel: 0532 667183. Prices: board, £17.50 including p&p; EPROM, £3 each and programming them £2/K.

This printed circuit board comes ready made to connect up to your ZX81 and provide it with your own "commands" stored as subroutines in a ROM.

The type of ROM used is called an EPROM which means it can be erased by ultra violet light and reprogrammed.

The board can take four 2716 (+5 volt type) EPROMs and connect up to the 16K pack as well.

The EPROM can be programmed by you or the company which supplies the board. EPROM Services supply the complete service, erasing the EPROM and reprogramming it from your machine code listing. The advantage of using this method of program storage is that no RAM is used up and the program is still safe in the ROM when the power is switched off.

The space allocated to the EPROMs on the board is 8K to 16K in the memory map, but due to the fact that only one IC is used to decode the address, it also appears in the 40K to 48K section as well. This means that you are limited to 16K of RAM as the Sinclair ROM takes up the space from 32K to 40K.

The instructions for inserting the EPROMs are easy to understand and include instructions on how to alter the board so that it can take 6116 RAMs instead.

The board arrived with an EPROM containing seven machine code routines, in the first 2K socket. They were RENUMBER (in steps of 10, starting from 10, but with no GOSUBs or GOTOs altered.), FREE MEMORY, PROGRAM LENGTH, MEMORY LENGTH, FILL (fills the screen with the character selected).

The last two convert decimal numbers POKEd into the system variables to hexadecimal numbers printed on the screen. There were a couple of errors in the last two, due to address changes being made but not clearly explained. The address changes were 16514/16515 to 16507/16508.

This EPROM board could be very useful on saving RAM, if you can write your own routines in machine code (EPROM services intend to make more programs available soon).

It does however restrict the amount of RAM you can use and as a number of boards that plug into the same address space, you will be limited as to what sockets you can use. **SA**

Open Forum

Open Forum is for you to publish your programs and ideas.
It is important that your programs are bug free before you send them in. We cannot test all of them.
Contributions should be sent to: Popular Computing Weekly, *Hobhouse Court, 19 Whitcomb Street, London WC2H 7HF.*

Flashing pound

on Vic-20

This program places the Vic pound sign as a defined character randomly on the screen and then scrolls it in pixel format vertically in its cell space.

Lines 150 to 190 copy the character down, redefine the space character and switch the character sets. Lines 200 to 240 move the bytes through the character cell to perform the scroll.

If the same routine was applied to more characters in machine code it should be possible to move displays on the screen in fine scroll, thus opening up interesting areas in the games and visual presentation fields.

Flashing pound
by Chris Palmer

```
10 REM FLOATING POUND
20 REM
30 REM CHRIS PALMER
40 :
50 :
60 SC=7680:CL=38400
70 CH=28*8:CB=7168
80 PRINT"□"
90 POKE 36879,8
100 FORI=1TO10
110 PS=INT(RND(1)*506+1)
120 POKE SC+PS,28
130 POKE CL+PS,7
140 NEXT
150 FORI=0TO7
160 POKECB+I+CH,PEEK(32768+I+CH)
170 POKE CB+32*8+I,0
180 NEXT
190 POKE36869,255
200 TV=PEEK(CB+CH)
210 FORI=0TO6
220 POKECB+CH+I,PEEK(CB+CH+I+1)
230 NEXT
240 POKE CB+7+CH,TV
250 GOTO 200
```

YOUR PROGRAM COULD WIN A PRIZE!

Each week the editor goes through all the programs that you send to Open Forum in order to find the Program of the Week.
The author of that program will qualify for DOUBLE the usual fee we pay for published programs (the usual fee is £10).

Programs which are most likely to be considered for the Star Prize will be computer printed and accompanied by a cassette.

The programs will be well documented, the documentation being typed with a double spacing between each line.

The documentation should start with a general description of the program and then give some detail of how the program has been constructed and of its special features.

Listings taken from a ZX Printer should be cut into convenient lengths and stuck down on to white paper.

Please enclose a self-addressed envelope.

Cone
by Jeremy Rowntree

```
10 REM 3-D CONES
20 REM BY J.ROWNTREE
30 REM AND S.LINDSAY
40 REM FOR BBC MICRO
50 REM
60 MODE 4
70 REM DEFINE COLOURS
80 VDU 19,1,2,0,0,0,19,0,7,0,0,0
90 A=640:B=130
100 MOVE A,B:PLOT 22,A,1024
110 J=RND(180)+220:K=RND(100)+20
120 I=K/J:V=RND(30)+25:S=50+RND(50)-V
130 MOVE A,B+K
140 FOR X=0 TO 2*PI STEP PI/180
150 DRAW A+J*SIN(X),B+K*COS(X)
160 NEXT X
170 FOR Q=0 TO 2*PI STEP PI/9
180 MOVE A+J*SIN(Q),B+K*COS(Q):DRAW A+(J-S)
*SIN(Q),B+V+I*(J-S)*COS(Q)
190 NEXT Q
200 B=B+V:J=J-S
210 REM TRY 'V+B' AT LINE 220
220 V=V-B/100
230 K=I*J
240 GOTO 130
```

Cone

on BBC Micro

This program will run on a BBC Micro Model A or B in any available graphics mode. When RUN it will draw a random cone in 3D in the form of a moulded grid.

The screen colours are defined at Line 80; the '2' sets the foreground, ie plotting, colour to green while the '7' sets the background, ie screen, colour to white.

These numbers can be altered to give different colours: 1 — red; 2 — green; 3 — yellow; 4 — blue; etc.

Program notes

Lines 90-120 define a random ellipse, centre A, B; major axis J; minor axis K. This ellipse is then plotted by Lines 130-160.

V and S define how the next ellipse relates to this one — V = vertical distance between them, S = amount by which ellipse shrinks — while I keep the ratio of the axes constant.

Lines 170-190 draw lines connecting the ellipses.

Line 220 adjusts the vertical step as the cone is plotted causing the shape to curve — with a '−' sign it curves inwards while with a '+' sign it curves outwards.

The STEP in Line 140 can be altered to give a more rapid plot (try PI/90) at the expense of resolution while altering the STEP in Line 170 will vary the spacing of the vertical lines.

Depth charge

on ZX81

In this program you command a frigate with a substantial supply of depth-charges. You must destroy the sub-marines, which travel at various depths, before five parts of the dam-wall are destroyed. The deeper the

to next page

from previous page

submarines are the more you score for hitting them.

You can move your frigate with the "5" and "8" arrow keys. To drop a depth-charge you have to press "6". The submarines will continue to move even after you have dropped a depth-charge but with a more inconsistent movement.

To start the game from the instructions you have to press NEW-LINE.

Program notes

Lines 1 to 10 set up the main variables, 15 and 16 set up the screen by calling a sub-routine at 1000.

Lines 20 to 250 make up the main body of the program.

Lines 270 to 295 make an explosion when the depth-charge reaches the lowest point it can move to and has not hit anything.

Lines 300 to 395 generate and move the submarine.

Lines 400 to 470 create an explosion and increment the score when a submarine is hit.

Lines 600 and 610 finish the program when five parts of the dam-wall have been destroyed.

Lines 1000 to 1110 generate the screen display.

Lines 2000 to 2110 print out the instructions.

Depth charge
by Clive Canten

```
   1 RAND
   2 LET CC=0
   3 GOSUB 2000
   4 LET SC=0
   5 LET X=10
   6 LET Y=3
   7 LET X3=0
   9 LET CC=PEEK 16396+PEEK 1639
7*256
  10 LET A$="   "
  15 GOSUB 1000
  16 PRINT AT 0,0;"SCORE=0"
  20 LET X$=INKEY$
  25 IF X$="5" THEN LET X=X-(X>0
)
  30 IF X$="8" THEN LET X=X+(X<2
5)
  40 PRINT AT 2,X;A$
  50 IF X$="6" AND Y=3 THEN LET
X1=X+2
  60 IF X$="6" AND Y=3 THEN GOTO
230
  70 IF RND>.9 THEN GOTO 300
  80 IF X2=0 THEN GOTO 320
 140 IF Y=3 THEN GOTO 20
 200 IF PEEK (CC+((X1+1)+(Y+33))
)<>136 THEN GOTO 400
 210 PRINT AT Y,X1;"  "
 220 PRINT AT Y,X1;" "
 230 LET Y=Y+1
 240 IF Y=15 THEN GOTO 270
 250 GOTO 20
 270 FOR K=1 TO 5
 280 PRINT AT Y,X1-1;"   "
 285 PRINT AT Y,X1-1;"  "
 290 NEXT K
 292 LET Y=3
 295 GOTO 20
 300 IF X2=0 THEN GOTO 140
 310 LET Y1=INT (RND*10)+7
 320 PRINT AT Y1,X2;"     "
 330 LET X2=X2+1
 340 IF X2=27 THEN GOTO 360
 350 GOTO 140
 352 IF PEEK (CC+(32+(Y1*33)))=1
28 THEN LET CC=CC+1
 354 FOR K=1 TO 5
 360 PRINT AT Y1,X2+1;"    "
 365 PRINT AT Y1,X2;"     "
 366 NEXT K
 370 IF CC=5 THEN GOTO 600
 390 LET X2=0
 395 GOTO 140
 400 FOR K=1 TO 5
 410 PRINT AT Y1,X2;"   "
 420 PRINT AT Y1,X2;"  "
 430 NEXT K
 440 LET SC=SC+Y1
 445 PRINT AT 0,6;SC
 450 LET Y=3
 460 LET X3=0
 470 GOTO 20
 600 PRINT AT 10,10;"GAME OVER"
 610 STOP
1000 PRINT
1010 PRINT
1020 PRINT
1030 FOR K=1 TO 19
1040 PRINT "
"
1050 NEXT K
1060 PRINT AT 21,0;"
"
1070 PRINT AT 20,0;"
"
1080 FOR K=1 TO 21
1090 PRINT AT K,31;"
"
1100 NEXT K
1110 RETURN
2000 PRINT TAB 10;"DEPTH-CHARGE"
2010 PRINT TAB 10;"------------"
2020 PRINT
2030 PRINT "  YOU HAVE AN UNLIMI
TED NUMBER OF DEPTH-CHARGES TO
DESTROY THE SUBMARINES WITH. YOU
 MUST BLOW- UP THE SUBMARINES BE
FORE THEY  DESTROY 5 PARTS OF T
HE DAM-WALL."
2040 PRINT
2050 PRINT
2060 PRINT " USE ""5"" TO MOVE L
EFT"
2070 PRINT " USE ""8"" TO MOVE R
IGHT"
2080 PRINT " USE ""6"" TO DROP A
 DEPTH-CHARGE"
2090 PAUSE 10000
2100 CLS
2110 RETURN
```

3D noughts and crosses

on Vic-20

The following program is not only for the Pet/Vic, it can be for anybody who owns a computer that can handle data statements — if not then the array will have to be put in as LET A(1) = 2 etc.

After every move, the program first identifies all cells in a particular direction; it then determines the situation in that line by multiplying the cell values M(P1,R1,C1) together; finally, it adds a value S to the priority values N(P1,R1,C1) for the cells in line. S depends upon the line situation determined previously, as you will see.

Figure 1 Illustrative game

For convenience, plays are only considered on the top plane. The player moves first with O's inputting Plane, Row, Column. The program replies on all even numbered moves with X's. Note that on all its moves the program has a choice of several moves, as cells N(P1,R1,C1) of the same value. After every move N(P1,R1,C1) is updated but only for cells in line with the move cell. Examination of each move in conjunction with figure 3 will make the process clear.

Note that N(P1,R1, C1) is updated even for already-occupied cells: checks could be introduced to avoid that, but the saving in time would probably not be very great.

Figure 1

MOVE	NEW Position				N(P1,R1,C1)			
	*	*	*	*	10	0	0	10
start	*	*	*	*	0	0	0	0
of	*	*	*	*	0	0	0	0
game	*	*	*	*	10	0	0	10
(1)								
1,1,1	0	*	*	*	20	10	10	20
	*	*	*	*	10	10	0	0
	*	*	*	*	10	0	10	0
	*	*	*	*	20	0	0	20
(2)								
1,1,4	0	*	*	X	10	0	0	38
	*	*	*	*	10	10	14	14
	*	*	*	*	10	14	10	14
	*	*	*	*	34	0	0	34
(3)								
1,2,1	0	*	*	X	108	0	0	38
	0	*	*	*	118	20	24	24
	*	*	*	*	108	14	10	14
	*	*	*	*	132	0	0	34
(4)								
1,4,1	0	*	*	X	10	0	0	138
	0	*	*	*	20	20	124	24
	*	*	*	*	10	114	10	14
	X	*	*	*	148	14	14	48
(5)								
1,2,2,	0	*	*	X	108	10	0	138
	0	0	*	*	118	226	222	122
	*	*	*	*	10	124	108	14
	X	*	*	*	148	24	14	146
(6)								
1,2,3	0	*	*	X	108	10	14	1138
	0	0	X	*	20	128	1138	24
	*	*	*	*	10	1124	122	14
	X	*	*	*	1148	24	28	146

Figure 2 Top plane of cube

0,0,0	0,0,1	0,0,2	0,0,3
0,1,0	0,1,1	0,1,2	0,1,3
0,2,0	0,2,1	0,2,2	0,2,3
0,3,0	0,3,1	0,3,2	0,3,3

Figure 3 Priority values for line situations

Line situation	Line value A(N)	Priority value B(N)
0 * * *	3	10
X * * *	2	14
0 0 * *	9	98
X X * *	4	100
0 0 0 *	27	900
X X X *	8	1000
0 X * *	6	−14
0 0 X *	18	−98
X X 0 *	12	−100

Note that the last three values of b(n) are used only at certain times. If a line was already blocked, neither B(7) or B(8) is used — line 5040. If 0X** is preceded by 0***, S is set to −10 line 5030.

Figure 4 Computation of − D −

SITUATION IN LINE:	0	*	0	*
VAL OF M(P1,R1,C1):	3	1	3	1
Value of D:	3×	1×	3×	1=9

Program notes

The program has been written deliberately to be machine-independent, and offers several opportunities for changes. Subroutine 2000, which draws the board after every move,

could be revised using PEEK and POKE to give a static display.

If line 3050 is deleted, the program will now play the same replies in a new game if the player repeats moves from a previous game. Various strategies can be examined to determine the program's weaknesses.

For an alternative game, reverse the inequality signs in lines 3030 and 3040. The program will now play anti-noughts and crosses, trying to avoid creating lines of four. Remember also to change lines 5000 and 5020. If lines 140-170 are deleted, and this line is substitued:

```
140 GOSUB 3000: PRINT:PRINT"YOUR MOVE IS";:PRINT P+1;R+1;C+1
```

The computer will take both sides and play itself. Experimentation is possible with the program playing both parts with different versions of B(n). If the losing values of B(N) are modified while the winning values are retained, the program becomes self-teaching and will eventually improve.

The values of B(0)-B(8) are not optimal and can be varied to change the program's play — the only changes required are in line 20.

With a few extra lines the value can be altered according to who is playing eg low values for XXX* and 0000* when you play and very high when your friends play.

Please remember you input plane, row and column (P,R,C) as 1,1,1 not 111. Good luck!

Message scfoller

on ZX81

The program asks you to type in a message. It then scrolls your message four letters at a time up the screen, enlarging each character 64 times; forming an 8 × 8 matrix. A machine code routine is used to help speed the printing up.

Program notes

The program is divided into two sections; the first to enter the machine code (Lines 1-30), when this section of the program has been entered (take care to get line 2 exact) apparent rubbish will appear in line 1 — this is the machine code, lines

to next page

3D noughts and crosses
by Martin Burke

```
1 OPEN1,4 : CMD 1
5 POKE537,PEEK(537)+3:REM RUN /STOP OFF.
10 REM-SET UP BOARD & INITIALISE VARIABLES.-
20 DATA3,10,2,14,9,98,4,100,27,900,8,1000,6,-14,18,-98,12,-100
30 DIM M(3,3,3),N(3,3,3),A(8),B(8)
40 D=1:A$="ROWS"
50 FOR P=0 TO 3:FOR R=0 TO 3:FOR C=0 TO 3
60 IFP=R ANDP=C THEN 110
70 IF P=C AND P=3-R THEN 110
80 IF P=3-C AND R=C THEN 110
90 IF P=R AND P=3-C THEN 110
100 GOTO 120
110 N(P,R,C)=10
120 M(P,R,C)=1:NEXT C:NEXT R:NEXT P
130 FOR N= TO 8:READ A(N),B(N):NEXT N:GOSUB 2000
140 PRINT"INPUT YOUR MOVE";
150 INPUT P,R,C:P=P-1:R=R-1:C=C-1
160 IF P>3 OR R>3 OR C>3 OR P<0 OR R<0 OR C<0 THEN 140
170 IF M(P,R,C)>1 THEN PRINT"CELL OCUPIED":PRINT:GOTO 140
180 M(P,R,C)=3: GOSUB 1000:GOSUB 2000:GOSUB 3000
190 PRINT"MY MOVE IS ":P+1;R+1;C+1 MC=MC+2
200 PRINT:M(P,C,R)=2:GOSUB1000:GOSUB2000
210 IF MC=64 THEN PRINT" GAME DRAWN ":END
220 GOTO 140
990 REM** FIND ON WHICH LINES THE MOVE CELL LIES.**
1000 FOR Q=1 TO 3:GOSUB 4000:NEXT Q
1020 IF P<>R AND P<>C AND R<>C THEN 1060
1030 IF P=RTHEN Q=4:GOSUB 4000
1040 IF P=C THEN Q=5:GOSUB 4000
1050 IF R=C THEN Q=6:GOSUB 4000
1060 IF P<>3-R AND P<>3-C AND R<>3-C THEN1130
1070 IF P=3-R THEN Q=7:GOSUB 4000
1080 IF P=3-C THEN Q=8:GOSUB 4000
1090 IF R=3-C THEN Q=9:GOSUB 4000
1100 IF P=R AND P=3-C THEN Q=10:GOSUB 4000
1110 IF P=C AND P=3-R THEN Q=11:GOSUB 4000
1120 IF P=3-C AND R=C THEN Q=12:GOSUB 4000
1130 IF P=R AND R=C THEN Q=13:GOSUB 4000
1140 RETURN
1190 REM###DRAW BOARD###
2000 PRINTTAB(10);"COLOUMNS":PRINT
2020 PRINTTAB(4);"1234 1234 1234 1234"
2030 FOR R=0 TO 3:PRINTMID$(A$,R+1,1);R+1;
2040 FOR P= 0 TO 3:FOR C=0 TO 3
2050 ON M(P,R,C) GOTO 2070,2080
2060 PRINT"o";:GOTO2090
2070 PRINT"/";:GOTO2090
2080 PRINT"X";
2090 NEXT C:PRINT" ";:NEXT P:PRINT:NEXT R:PRINT:PRINT:RETURN
2990 REM@@@@FIND CELL WITH HIGHEST PRIOROTY VALUE@@@@
3000 HV=0:FOR P=0 TO 3:FOR R=0 TO 3:FOR C=0 TO 3
3020 IF M(P,R,C)>1 THEN 3060
3030 IF N(P,R,C)<HV THEN 3060
3040 IF N(P,R,C)>HV THEN HV=N(P,R,C):P1=P:R1=R:C1=C:GOTO 3060
3050 IF RND(8)>0.5 THEN HV=N(P,R,C):P1=P:R1=R:C1=C
3060 NEXT C:NEXT R:NEXT P:P=P1:R=R1:C=C1:RETURN
3990 REM################ ADD PRIOROTY VALUES TO TOTAL################
3995 REM////////////////FOR CELLS IN LINE WITH MOVE CELL.////////////////
4000 FOR T=0 TO 3:P1=P:R1=R:C1=C
4020 ON Q GOTO 4050,4060,4070
4030 P1=T
4040 ONQ-3GOTO4060,4070,4100,4110,4120,4130,4140,4150,4160,4170
4050 P1=T:GOTO 4180
4060 R1=T:GOTO 4180
4070 C1=T:GOTO 4180
4100 P1=P:R1=T:C1=T:GOTO 4180
4110 R1=3-T:GOTO 4180
4120 C1=3-T:GOTO 4180
4130 P1=P:R1=T:C1=3-T:GOTO 4180
4140 R1=T:C1=3-T:GOTO 4180
4150 R1=3-T:C1=3-T:GOTO 4180
4160 R1=3-T:C1=3-T:GOTO4180
4170 R1=T:C1=T
4180 IF F=1 THENN(P1,R1,C1)=N(P1,R1,C1)+S:GOTO 4200
4190 D=D*M(P1,R1,C1)
4200 NEXT T:IF F=0 THEN F=1:GOSUB 5000:GOTO 4000
4210 F=0:RETURN
4900 REM***FIND NEW PRIORITY VALUE TO BE STORED AND ADDED.***
5000 IF D=16 THEN GOSUB 2000:PRINT:PRINT"    I WIN !!!":END
5020 IF D=81 THEN PRINT:PRINT"    YOU WIN  ":PRINT:END
5030 IF D=6 AND M(P,R,C)=2 THEN S=-10:GOTO 5070
5040 IF D/M(P,R,C)=6 THEN S=0:GOTO 5070
5050 FOR N= 0 TO 8:IF D=B(N) THEN S=B(N):GOTO 5070
5060 NEXT N
5070 D=1:RETURN
```

Open Forum

from previous page

2,5,10,20,25,30 can then be deleted. Lines 40-195 can then be entered — this is the main program. In the program, the two characters used for printing are a space and inverse space. You can change the characters by 'POKE'ing 16529 and 16536 with two new characters.

For example:

```
POKE 16529,128
POKE 16536,0
```

will give inverse video character.

You can use the program to give a continuously repeating message by adding

```
200 GOTO 110
```

Message scroller
by Philip Haywood

```
    1 REM 12345678901234567890123
4567890123
    2 LET B$="2100400E0406005E3E0
0CB03D2994003E800710F423DDC20D40C
9"
    5 LET N=16020
   10 FOR R=1 TO LEN B$ STEP 2
   20 POKE N,(16*(CODE (B$(R))-28
))+CODE B$(R+1)-28
   25 LET N=N+1
   30 NEXT R
   40 PRINT "TYPE IN YOUR MESSAGE
"
  100 INPUT A$
  110 IF LEN A$/4=INT (LEN A$/4)
THEN GOTO 130
  115 LET X=A$ (LEN A$-(4*INT (L
EN A$/4))-4)
  120 FOR U=1 TO X
  122 LET A$=A$+" "
  124 NEXT U
  130 FOR G=1 TO LEN A$ STEP 4
  140 FOR E=0 TO 7
  150 LET N=1
  160 FOR R=0 TO G+3
  170 POKE N+16513,PEEK (7680+8*C
ODE  A$(R)+E)
  178 LET N=N+1
  180 NEXT R
  190 RAND USR 16520
  195 SCROLL
  199 NEXT E
  198 NEXT G
```

Graph
by Stephen Read

```
   10 REM 'GRAPH"
   20 PRINT AT 1,13;"GRAPH"
   30 PRINT AT 4,5;"  AXIS = 0 TO
 48"
   40 PRINT AT 5,5;"  AXIS = 0 TO
 20"
   45 PRINT AT 8,6;"1 STATE AXES'
   48 PRINT AT 9,8;"(12 CHARACTER
S)"
   45 PRINT AT 12,6;"2 PLOT FUNCT
ION"
   50 PRINT AT 13,6;"(G=1=PLOT,G=
0=UNPLOT)"
   55 PRINT AT 14,6;"(R=X TO FINI
SH PLOT)"
   58 PRINT AT 17,6;"3 LABEL FUNC
TION"
   65 PRINT AT 18,6;"(G=1=PRINT,G
=0=RUBOUT)"
   70 PRINT AT 19,8;"(R=X TO FINI
SH)"
   75 PRINT AT 21,19;"PRESS ENTER
"
   80 INPUT H$
   90 CLS
  120 FOR X=3 TO 63
  125 PLOT X,3
  130 NEXT X
  135 FOR Y=3 TO 41
  140 PLOT 3,Y
  145 NEXT Y
  150 PRINT AT 1,20;"X=?"
  155 LET A$="X"
  170 INPUT A$
  173 PRINT AT 21,7;"X="+A$
  175 PRINT AT 1,20;"    "
  185 PRINT AT 0,20;"Y=?"
  187 LET B$=""
  190 INPUT B$
  195 PRINT AT 0,0;"Y="+B$
  200 PRINT AT 0,20;"    "
  250 REM SUBROUTINE STARTS HERE
  260 GOSUB 300
  300 PRINT AT 0,23;"FROM(A,B)"
  310 INPUT A
  311 IF A=X THEN GOTO 1270
  312 PRINT AT 0,27;"  "
  315 PRINT AT 0,27;A
  320 INPUT B
  325 PRINT AT 0,30;"  "
  328 PRINT AT 0,30;B
  330 PRINT AT 0,20;"    "
  340 PRINT AT 1,20;"TO(C,D)"
  350 INPUT C
  352 PRINT AT 1,27;"  "
  355 PRINT AT 1,27;C
  360 INPUT D
  362 PRINT AT 1,30;"  "
  365 PRINT AT 1,30;D
  370 PRINT AT 1,25;"  "
  650 LET A=A+3
  670 LET B=B+3
  900 LET C=C+3
  930 LET D=D+3
 1000 LET U=C-A
 1010 LET V=D-B
 1020 LET D1X=SGN U
 1030 LET D1Y=SGN V
 1040 LET D2X=SGN U
 1050 LET D2Y=0
 1060 LET M=ABS U
 1070 LET N=ABS V
 1080 IF M>N THEN GOTO 1130
 1090 LET D2X=0
 1100 LET D2Y=SGN V
 1110 LET M=ABS V
 1120 LET N=ABS U
 1140 LET S=INT (M/2)
 1150 REM PLOT/UNPLOT STARTS
 1152 PRINT AT 21,24;"G=1 OR 0"
 1154 INPUT G
 1156 IF G=0 THEN GOTO 1164
 1158 FOR I=0 TO M
 1160 PLOT A,B
 1162 GOTO 1168
 1164 FOR I=0 TO M
 1166 UNPLOT A,B
 1168 PRINT AT 21,24;"  "
 1170 LET S=S+N
 1180 IF S<M THEN GOTO 1230
 1190 LET S=S-M
 1200 LET A=A+D1X
 1210 LET B=B+D1Y
 1220 GOTO 1250
 1230 LET A=A+D2X
 1240 LET B=B+D2Y
 1250 NEXT I
 1260 GOTO 300
 1270 PRINT AT 0,23;"  "
 1280 PRINT AT 1,20;"  "
 1290 PRINT AT 21,21;"LABEL:A ,B"
 1300 INPUT A
 1305 IF A=X THEN GOTO 1450
 1310 PRINT AT 21,27;A
 1320 INPUT B
 1330 PRINT AT 21,30;B
 1340 PRINT AT INT (40-B)/2,INT (
A+3)/2;"'"
 1350 PRINT AT 21,21;"  G=1 OR 0
 "
 1370 INPUT G
 1380 PRINT AT 21,24;"  "
 1390 IF G=1 THEN GOTO 1430
 1420 PRINT AT INT (40-B)/2,INT (
A+3)/2;"  "
 1425 GOTO 1950
 1430 INPUT A$
 1440 PRINT AT INT (40-B)/2,INT (
A+3)/2;A$
 1450 GOTO 1300
 1460 PRINT AT 21,21;"
 "
 1470 STOP
```

Graph

on ZX81

This program draws and labels — as per example — including unplot (rub-out) options. Instructions are at the beginning so as not to interfere with plotting. Just copy the result from screen to printer for presentation.

For those whose ZX81 works for them!

Train race
by Eric Deeson

```
 10 VDU 23, 225, 0, 255, 65, 73, 77, 75, 127, 127      (
 20 VDU 23, 226, 0, 0, 0, 0, 0, 24, 255, 255           (
 30 VDU 23, 227, 0, 28, 28, 28, 28, 28, 254, 254       ( ergine blocks
 40 VDU 23, 228, 127, 127, 127, 127, 255, 13, 7, 2     (
 50 VDU 23, 229, 255, 255, 255, 255, 255, 155, 14, 4   (
 60 VDU 23, 230, 254, 254, 254, 254, 255  54, 28, 8    (
 70 VDU 23, 231, 16, 168, 84, 170, 84, 120, 20, 40     ( smoke
 80 MODE 6
 90 VDU 5
100 VDU 19, 1, 0, 0, 0, 0, 19, 0, 3, 0, 0, 0,          ( colour defination
110 CLS
120 A=0: A1=0: PRINT TAB (36,10); "1"; TAB(36,16);"2"
130 REPEAT                                             ( start loop
140 A=A+RND(3):  B=9                                   ( train 1
150 PROC T(A,B)                                        (
160 A1=A1+RND(3): B=15                                 (
170 PROC T(A1,B)                                       ( train 2
180 UNTIL A>26 OR A1>26                                ( end loop
190 PRINT TAB (2,20); "Number ";
200 IF A >26 THEN PRINT "1"; ELSE PRINT "2";
210 PRINT " won."
220 FOR Z=1 to 3000:NEXT                               ( pause
230 GOTO 110                                           ( restart
240 DEF PROC T (A,B)
250 PRINT TAB (A + 4, B); CHR$ (231): SOUND 1, -15,    (
    B * 4, 4                                           ( smoke, whistle
260 PRINT TAB (A,B+1); "- - -" + CHR$ (225) + CHR$ (226) (
    + CHR$ (227); TAB (A,B + 2); "- - -"+ CHR$ (228) +  ( engine
    CHR$ (229) + CHR$ (230)                            (
270 FOR Z = 1 TO 1000: NEXT                            ( pause
280 END PROC
```

Train race

on BBC Micro

No way is this a novel game — two engines race across the screen, moving forward a random step each move until one or other reaches the end. The principle is the basis of a number of similar exercises, non-interactive or interactive.

It is straightforward to add twiddly bits like reverse as well as forward motion, barriers and switching tracks (one reason I chose MODE 6, which gives you the "railway lines" for free).

Observe the BBC features of:

- user-definable graphics (lines 10-70);
- calling procedures with different parameters (compare 150, 170, 240);
- sound for the "whistles" (line 250).

Open Forum

Space warrior

on Vic-20

The idea is simple but requires some skill to manoeuvre the cross until it is over the centre of an 'alien', when the player must fire and destroy it. The cross can be maneouvred in both vertical and horizontal directions with a 'repeat' action so to avoid having to keep on pressing the same key to move it a few spaces. The keys I have chosen are:

Z — left
C — right
V — down
G — up
M — fire

They may seem a little 'clustered up' but I assure you after only a few games they become no problem.

There are ten aliens which must be destroyed within the time limit. The player enters his time at the beginning of the program (4-10 min; this can be shortened or lengthened by changing one or two of the lines from 66-72).

The aliens are randomly positioned at the top of the screen and then come down the screen at totally random movements. The reaction from pressing a key to the movement of the cross is very good.

When, or if, the alien reaches a red border line then it disappears and another is generated. Also if the cross touches the border then it marks the end of the game.

The sound generators have been put to good use especially when an explosion occurs.

Space warrior
by Gerhard Nath

```
10 PRINT"□"
20 PRINT"  SPACE  WARRIOR!!"
30 PRINT"  ──────────────────"
40 PRINT"■■■"
50 PRINT"Z-LEFT C-RIGHT V-DOWN"
60 PRINT"G-UP & M-FIRE."
65 PRINT"ENTER YOUR TIME(4-10)":INPUTF
66 IFF=4THENS$="000400"
67 IFF=5THENS$="000500"
68 IFF=6THENS$="000600"
69 IFF=7THENS$="000700"
70 IFF=8THENS$="000800"
71 IFF=9THENS$="000900"
72 IFF=10THENS$="001000"
75 PRINT:PRINT"PRESS A KEY TO START"
80 GETA$:IFA$=""THEN80
98 TI$="000000"
100 POKE36879,14:PRINT"□"
101 L=0
110 FORI=1TO30
120 Q=INT(470*RND(1))+7680
130 POKEQ,46:NEXTI
140 FORI=0TO20:POKE7680+I,160:POKE38400+I,2:NEXTI
150 FORI=0TO463STEP22
160 POKE7702+I,160:POKE38422+I,2:NEXTI
170 FORI=0TO20:POKE8154+I,160:POKE38884+I,2:NEXTI
180 FORI=0TO485STEP22
190 POKE8163-I,160:POKE38883-I,2:NEXTI
300 REM ALIENS
305 GOSUB2000
410 J=0:Q=INT(18*RND(1))+7703
411 POKE36876,0:FORT=1TO1000:NEXTT
420 POKEQ+J,60 POKEQ+J+1,216 POKEQ+J+2,62
430 POKE36878,3:POKE36876,220:GOSUB2000
435 FORT=1TO50:NEXTT:GOSUB2000
440 POKEQ+J,32:POKEQ+J+1,32:POKEQ+J+2,32
450 V=INT(3*RND(1))+1
452 IFV=1THENJ=J+21
454 IFV=2THENJ=J+22
456 IFV=3THENJ=J+23
460 IFPEEK(Q+J+1)=160THEN410
462 IFPEEK(Q+J)=160THEN410
464 IFPEEK(Q+J+2)=160THEN410
467 IFTI$>=S$THEN9000
468 POKE36876,0
470 GOTO420
1000 IFPEEK(Q+J+1)=91THEN1150
1020 POKE36878,13:POKE36874,220
1030 FORT=1TO50:NEXTT
1040 POKE36874,0
1060 RETURN
1150 REM HIT ALIEN
1151 POKE36876,0:FORT=1TO60:NEXTT
1160 POKE36878,15:FORM=135TO239STEP2
1170 POKE36876,M:NEXTM
1175 FORT=1TO700:NEXTT
1180 POKE36876,0:POKE36877,220
1181 FORM=15TO0STEP-1
```

Car race

on ZX81

The object of this game is to manoeuvre your racing car (shown as a multiplication sign) round the racing circuit in a clockwise direction without crashing into the barriers.

Every time a lap is completed the computer adds to your score and randomly places a number of obstacles on the circuit which you have to avoid.

As well as keeping your score the computer also keeps the highest score.

to next page

Open Forum

```
1182 POKE36878,M
1183 FORT=1TO300:NEXTT
1184 NEXTM
1185 POKE36877,0:L=L+1:IFL>=10THEN8000
1190 GOTO300
2000 REM YOUR SHIP
2020 POKE7910+D,91
2040 S=PEEK(197)
2050 IFS=33THEND=D-1:POKE7911+D,32
2060 IFS=34THEND=D+1:POKE7909+D,32
2065 IFS=36THENGOSUB1000
2070 IFS=19THEND=D-22:POKE7932+D,32
2080 IFS=27THEND=D+22:POKE7888+D,32
2085 IFPEEK(7910+D)=160THEN9000
2090 RETURN
8000 PRINT"■":POKE36879,93
8010 PRINT"■■■■■■■■■■":PRINT"  WELL DONE YOU'VE"
8020 PRINT"DESTROYED THE ENEMY!!"
8030 END
9000 REM YOUR DEAD
9005 POKE36876,0
9010 POKE36877,220
9020 POKE36879,42
9030 FORI=15TO0STEP-1
9040 POKE36878,I
9045 FORT=1TO300:NEXTT
9046 NEXTI
9050 POKE36877,0
9060 PRINT"■"
9070 PRINT"■■■■■■■■■■■"
9080 PRINT"■        YOU'VE BEEN      "
9090 PRINT"■        ATOMISED!!       "
9100 FORT=1TO3000:NEXTT
9200 END
```

from previous page

Program notes

The variables are as follows:

S — Your score
HS — High Score
Y — Y co-ordinate of car
X — X co-ordinate of car
L — Number or points to be added to score
D$ — String which determines direction of car

Lines 10 — 450 print out the circuit
Lines 1000 — 1100 randomly place obstacles

Array D contains the set-up of the circuit.

When the program is run the screen will go blank for approximately five seconds while the circuit is being printed.

Listings taken from a ZX printer should be cut into convenient lengths and stuck down on to white paper. Please enclose a self-addressed envelope.

Car race

by Alasdair Sanderson

```
1 LET HS=0
2 *AAT
5 LET L=0
6 LET Y=3
7 LET X=8
8 LET D$="S"
9 PRINT AT 2,5;"■";AT 4,5;"■"
10 DIM D(31,21)
20 FOR F=1 TO 30
30 PRINT AT 1,F;"■"
40 LET D(F,1)=1
50 NEXT F
60 FOR F=1 TO 30
70 PRINT AT 20,F;"■"
80 LET D(F,20)=1
90 NEXT F
100 FOR F=1 TO 20
110 PRINT AT F,1;"■"
120 LET D(1,F)=1
130 NEXT F
140 FOR F=1 TO 20
150 PRINT AT F,30;"■"
160 LET D(30,F)=1
170 NEXT F
180 FOR F=5 TO 26
190 PRINT AT 5,F;"■"
200 LET D(F,5)=1
210 NEXT F
220 FOR F=9 TO 30
230 PRINT AT 9,F;"■"
240 LET D(F,9)=1
245 NEXT F
250 FOR F=5 TO 26
260 PRINT AT 13,F;"■"
270 LET D(F,13)=1
280 NEXT F
290 FOR F=5 TO 16
300 PRINT AT F,5;"■"
310 LET D(5,F)=1
315 NEXT F
320 FOR F=14 TO 18
330 PRINT AT F,17;"■"
340 LET D(17,F)=1
350 NEXT F
360 FOR F=18 TO 30
370 PRINT AT F,9;"■"
380 LET D(9,F)=1
390 PRINT AT F,26;"■"
400 LET D(26,F)=1
410 NEXT F
```

```
420 PRINT AT 17,13;"■"
430 PRINT AT 17,21;"■"
440 LET D(13,17)=1
450 LET D(21,17)=1
455 LET I=0
456 LET J=0
457 LET S=(S+L)*1
460 PRINT AT 0,2;"POINTS ";L;AT
0,14;"HI-SCORE ";HS
465 LET D$=""
500 PRINT AT Y,X;C$
505 A=0U
510 IF INKEY$="5" THEN LET D$="
5"
511 IF INKEY$="6" THEN LET D$="
6"
512 IF INKEY$="7" THEN LET D$="
7"
513 IF INKE/$="8" THEN LET D$="
8"
515 PRINT AT Y,X;"
530 IF D$="5" THEN LET X=X-1
535 IF D$="6" THEN LET X=X+1
540 IF D$="8" THEN LET Y=Y+1
550 IF D$="7" THEN LET Y=Y-1
560 IF D(X,Y)=1 THEN GOTO 2000
580 IF X=5 AND Y>1 AND Y<5 THEN
GOTO 591
585 PRINT AT Y,X;C$
590 GOTO 510
591 LET S=S+L
592 IF S>HS THEN LET HS=S
593 PRINT AT 2,5;"■";AT 3,5;AT
4,5;"■"
595 GOSUB 1000
600 PRINT AT 0,9,5;AT 0,23;HS
602 GOTO 585
700 PRINT AT 21,0,"PRESS NEWLIN
E FOR ANOTHER GO"
701 INPUT I$
705 CLS
706 GOTO 2
1000 FOR F=1 TO INT (RND*10)+1
1010 LET XX=INT (RND*31)+1
1020 LET YY=INT (RND*21)+1
1030 IF XX=31 OR YY=21 THEN GOTO
1010
1040 IF XX>4 AND XX<7 AND YY>1 A
ND YY<5 THEN GOTO 1010
1050 IF XX=4 AND YY=3 THEN GOTO
1010
1060 IF D(XX,YY)=1 THEN GOTO 101
0
1065 PRINT AT Y,X;"*"
1070 PRINT AT YY,XX;"■"
1080 LET D(XX,YY)=1
1090 NEXT F
1095 LET L=L+F
1100 RETURN
2000 PRINT AT Y,X-1;"■";X=";AT Y-1
,X-1;"■■■";AT Y+1,X-1;"■■■"
2010 GOTO 700
2500 REM SPEEDWAY BY
          A.M.SANDERSON 1982
```

Vic-Orator

on Vic-20

This program which will run on an unexpanded Vic will utilise the facility of a user-defined character set to display double height characters on the screen, the visual equivalent of a shout, ideal for group or display activities.

To write this program to fit into the limited space I have had to cheat a little and only produce 128 characters in the set so the user must be careful to not print any reverse characters (128 onwards) otherwise strange things may happen!

Vic owners with additional memory can get round this by setting-up a 4K character generator and they will also need to change the top of memory locations that are poke-ed in lines 3, 4, 33, 34.

The character set used in the program is the upper case letters and graphics set. If you require upper and lower case letters the peek address in lines 10 and 11 needs to be changed to peek(34816+c).

The REMs are reasonably self-explanatory.

Open Forum

Corrections

A number of mistakes have crept into the programs published in the first issues of *Popular Computing Weekly*. We will print corrections as quickly as possible.

Here is a list of the corrections we have to date:

Vol 1 No 1 page 9
Space Amaze

Line 25 should be renumbered line 15.
Line 560 should read IF D$="<" AND X>1 THEN LET X=X+1
Line 2000 should read LET Q(X,Y)=0

Vol 1 No 2 Page 17
Squash

```
 80 PROCset-up
130 PRINT CHR$(RND(5)+128);"for";(TIME DIV
    10)/10;"seconds"
170 IF A$="Z" OR A$="M" THEN PROCmove-bat
180 PROCmove-ball
260 SOUND T+16,-15,RND(100)+100,255
340 DEF PROCset-up
380 PRINT TAB(10,T+10);STRING$(20), CHR$
    (255); TAB(30,T+10);CHR$(255)
460 TIME=0
```

Vol 1 No 2 Page 18
Subchase

```
  5 LET TH=5
100 LET B$="TAN (YT/XT)"
355 GOSUB 1000
400 UNPLOT INT (H/(P/64)),30
```

Vol 1 No 3 Page 9
Hell Driver

```
 10 GOSUB 1000
 80 DIM A1(14), B1(14), C1(14), G$(5), B$(10),
    M$(14)
930 ML=0: PRINT "36 spaces"
```

Vol 1 No 3 Page 15
Scrabble Scorer

```
1110 PRINT AT 6, J, 10; Q$(4,J)
1225 REM C=COUNT OF PLAYERS
1430 LET Z=SUM*(X-28)
1500 LET S(R,N)=Z
1510 LET T(N)=T(N)+Z
1520  IF N<>2 OR P>2 THEN PRINT AT
     X(N,1),X(N,2);T(N);" "
1530  IF N=2 AND P=2 THEN PRINT AT
     X(3,1),X(3,2);T(N);" "
1540 NEXT N
1550 LET R=R+1
1560 GOTO 1210
4000 CLS
4010 PRINT Q$(1); Q$(2);
4020 IF P>2 THEN PRINT Q$(3);
4030 IF P>3 THEN PRINT Q$(4)
4040 PRINT AT 1,0;"--------------------------------"
4050 FOR J=1 TO R
4060 PRINT TAB 1;S(J,1); TAB(10); S(J,2);
4070 IF P>2 THEN PRINT TAB 19;S(J,3);
4080 IF P>3 THEN PRINT TAB 28;S(J,4)
4090 PRINT
4100 NEXT J
4110 STOP
8000 CLEAR
8010 SAVE "SCRABBLE"
8020 RUN
```

Vol 1 No 3 Page 21
Programming

```
50 FOR I=CODE"(Graphic 1)" TO CODE "(Graphic
   A)"
60 IF B$(F1,I)=" "THEN GOTO CODE "(Graphic W)"
```

Vol 1 No 4 Page 17
Space Pilot

```
370 LET P1=INT((S3/1000)/2)
730 LET X=10-(INKEY$="3") (the rest of the line
    remains the same)
```

Vol 1 No 4 P 23
Programming

Some of you noticed that the programs were missing from Barry Cornhill's article on chaining ZX81 programs in the 13 May 1982 issue. To put matters right, here they are:

```
  5 REM P1 CREATE DATA
 10 DIM B(10)
 20 FOR I=1 TO 10
 30 LET B(I)=I
 40 NEXT I
 50 LET J=PEEK 16400+256*PEEK 1
5401
 60 LET K=PEEK 16396+256*PEEK 1
5369
 70 FOR I=0 TO 55
 80 POKE K+I,PEEK (J+I)
 90 NEXT I
```

```
  9 REM P2
 10 DIM B(10)
 20 LET J=PEEK 16400+256*PEEK 1
5401
 30 LET K=PEEK 16396+256*PEEK 1
5369
 40 FOR I=0 TO 55
 50 POKE J+I,PEEK (K+I)
 60 NEXT I
 70 FOR I=1 TO 10
 80 LET B(I)=10*B(I)
 90 PRINT B(I)
100 NEXT I
```

```
  9 REM P3
 10 DIM B(10)
 20 LET J=PEEK 16400+256*PEEK 1
5401
 30 LET K=PEEK 16396+256*PEEK 1
5369
 40 FOR I=0 TO 55
 50 POKE J+I,PEEK (K+I)
 60 NEXT I
 70 FOR I=1 TO 10
 75 LET B(I)=PI*B(I)
 80 LET B(I)=10*B(I)
 90 PRINT B(I)
100 NEXT I
```

```
  9 REM P5
 10 DIM B(10)
 20 LET J=PEEK 16400+256*PEEK 1
5401
 30 LET K=PEEK 16396+256*PEEK 1
5369
 40 FOR I=0 TO 55
 50 POKE J+I,PEEK (K+I)
 60 NEXT I
 70 FOR I=1 TO 10
 80 LET B(I)=10*B(I)
 90 PRINT B(I)
100 NEXT I
110 FOR I=0 TO 55
120 POKE K+I,PEEK (J+I)
130 NEXT I
```

```
  9 REM P6
 10 DIM B(10)
 20 LET J=PEEK 16400+256*PEEK 1
5401
 30 LET K=PEEK 16396+256*PEEK 1
5369
 40 FOR I=0 TO 55
 50 POKE J+I,PEEK (K+I)
 60 NEXT I
 70 FOR I=1 TO 10
 80 LET B(I)=PI*B(I)
 90 PRINT B(I)
100 NEXT I
110 FOR I=0 TO 55
120 POKE K+I,PEEK (J+I)
130 NEXT I
```

```
  9 REM DP
 10 DIM B(10)
 20 LET J=PEEK 16400+256*PEEK 1
5401
 30 LET K=PEEK 16396+256*PEEK 1
5369
 40 FOR I=0 TO 55
 50 POKE J+I,PEEK (K+I)
 60 NEXT I
 70 FOR I=1 TO 10
 90 PRINT B(I)
100 NEXT I
110 PRINT "TO SAVE ENTER N/L"
120 INPUT Z$
125 GO SUB 130
126 STOP
130 SAVE "DATAFILE"
140 LET J=PEEK 16400+256*PEEK 1
5401
150 LET K=PEEK 16396+256*PEEK 1
5369
160 FOR I=0 TO 55
170 POKE K+I,PEEK (J+I)
180 NEXT I
190 RETURN
```

```
 1 REM  VIC-ORATOR   (C)1982 KEN CLARK   RUNS ON 3.5K VIC-20
 2 REM***LOWER MEMORY TO PROTECT MEMORY @ 5120 ONWARDS***
 3 POKE51,255:POKE52,19
 4 POKE55,255:POKE56,19
 5 CLR
 6 REM***BLANK SCREEN WHILE SETTING UP REGISTERS ETC.***
 7 POKE36867,128
 8 REM***SET UP CHARACTER GENERATOR @ 5120 ONWARDS***
 9 I=0:C=0
10 POKE5120+I,PEEK(32768+C)
11 POKE5120+I+1,PEEK(32768+C)
12 C=C+1:I=I+2
13 IF C<1024 THEN10
14 REM***SET REGISTERS FOR USER DEFINED CHAR GENERATOR***
15 POKE36869,253
16 POKE36866,PEEK(36866)OR128
17 REM***AJUST SCREEN SIZE AND SELECT 16 X 8 CHARACTERS***
18 POKE36867,149
19 REM***PUT YOUR MESSAGE/TEXT FROM HERE....***
20 PRINT"■LARGE CHARACTERS ARE"
21 PRINT
22 PRINT"POSSIBLE ON THE VIC-20"
23 PRINT
24 PRINT"WITH A USER DEFINABLE"
25 PRINT
26 PRINT"CHARACTER SET."
27 PRINT:PRINT:PRINT"■PRESS ANY KEY"
28 REM***.....TO HERE - THEN RESET VIC WHEN FINISHED***
29 GETA$:IFA$=""THEN29
30 POKE36869,240
31 POKE36866,150
32 POKE36867,174
33 POKE51,255:POKE52,29
34 POKE55,255:POKE56,29
```

Vic-Orator
by Ken Clarke

PROGRAM OF THE WEEK

How long is a piece of string?

***David Lawrence** explains the use of character codes on the ZX81*

Lest ZX-81 owners should ever be tempted to forget the importance of the humble byte, here are three practical and elegant ways of handling strings which depend entirely upon the fact that a single string character can take any one of 256 forms.

First, the formatting of strings which have been stored in dimensioned arrays:

If you were to enter lines 20 to 50 of the demonstration program you would quickly discover that though you had intended to store 'FOX' in line 1 of the array A$, what is actually stored there is 'FOX' followed by 17 spaces. This is because the ZX-81 fills a dimensioned string array with spaces until the positions are used for something else. This could be overcome by changing A$(1) in line 50 to A$(1,1 TO 3) but the array may be intended to hold a large number of strings of different lengths to be fitted into a text at various points. What is needed is a simple method enabling the program to know how much of a dimensioned string is useful information and how much is padding.

An effective answer is illustrated by lines 60-80. Line 60, which could be used with any string up to 254 characters, simply tags a single character on to the front of the string — the CODE value of that character being equal to the length of the string plus the extra character. Line 80 now shows how the useful section of the string can be unerringly identified. A$ (2,2 TO CODE A$(2,1)) is the original B$ without its padding — in the case of 'FOX' the character with a CODE value of 4 is tagged on to the front and the complex term boils down to no more than A$(2,2 to 4).

This technique of adding 'string length indicators' can considerably add to speed and flexibility compared to commonly used methods such as examining the string, character by character, to assess its length.

ZX81 . . . something to byte on

Our second usage for single character codes is in relation to the storage of data in long strings rather than in multi-dimensioned arrays. Suppose, for instance, that we have a large number of names to store and later access. This can be done by setting up an array with sufficient lines to take all the names. The problem is that if the longest name is likely to be 20 characters long then every line will have to be 20 spaces long, even though most of the other names will only need around 10 characters, a massive waste of memory space.

Using indicators

Alternatively, the names can be stored in one long string, for instance 'Smith, John, Adams, Bill, Brown, Alison. Hence, no space is wasted but there is, equally, no way for the program to know where one name ends and the next begins. We could put a special marker, such as an asterisk, in between the names, but this would entail examining every character in the string whenever individual names had to be identified.

The section of the demonstration program starting at line 100 illustrates how a long string can be made up of individual entries, each with an SLI tacked on to the front. Lines 200 onwards then show how much indicators can be used to retrieve items from the string. The loop at line 240 simply uses the SLIs to jump from the beginning of one item to the next until the correct item is reached. Line 280 is not more than a slightly more complicated version of line 80, except that instead of starting to print at position 2, we start at C+1, where C is the position of the SLI of the desired entry.

This section can be used with a little adaptation to produce an effective filing system, nor is it limited to single items of information such as names, since within each entry further SLIs can identify sub-divisions such as: name, address, telephone.

Finally, we shall examine how single character codes can aid in the production of well formatted interactive programs. The program section titled 'Typical Input Routine' illustrates some of the functions that have to be performed when requesting information from the program user.

If the program contains many different requests for information, many of these functions can valuably be transferred to a single subroutine such as that from line 400 to line 560. Before this subroutine can be called, however, the string output requesting information will have to be specified (even if the same request has been made elsewhere), together with the position it is to be printed on the screen which, together with the line calling the subroutine, makes four lines for each call.

The effective use of single character codes is illustrated by the section from lines 570 to 780, which works on the assumption that A$ is a two dimensional array containing the questions to be printed. Each question has an SLI attached, followed by two bytes which indicate the screen position at which the string is to be printed. Further single characters could be included which would allow all the printing, whether or not an input is required, to be performed by the subroutine.

The codes themselves are simply attached by the use of a subroutine such as that found at line 790 (which would only be required during program development) and every code character replaces a line defining a variable in the program.

Perhaps the humble byte is not so humble after all.

mming

Demonstration program
by David Lawrence

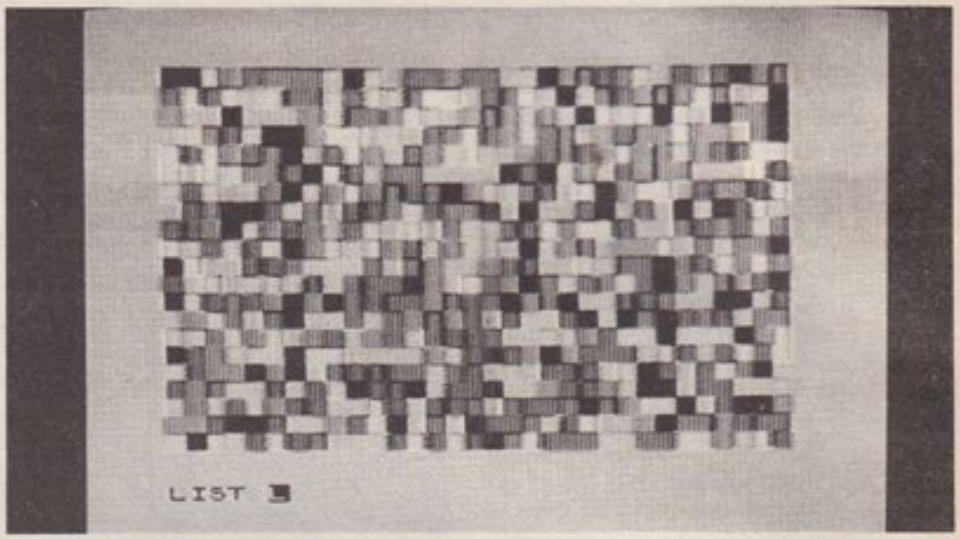
```
  10 REM ************************
        REMOVE PADDING
        ************************
  20 DIM A$(2,20)
  30 LET B$="FOX"
  40 LET A$(1)=B$
  50 PRINT "THE QUICK BROWN ";A$
(1);" JUMPS OVER THE LAZY DOG."
  60 LET B$=CHR$ (LEN B$+1)+B$
  70 LET A$(2)=B$
  80 PRINT "THE QUICK BROWN ";A$
(2,2 TO CODE A$(2,1));" JUMPS OV
ER THE LAZY DOG."
  90 STOP
 100 REM ************************
        DATA IN STRINGS
        ************************
 110 LET C$=""
 120 PRINT "ENTRY:";
 130 INPUT Q$
 140 PRINT Q$
 150 IF Q$="STOP" THEN GOTO 200
 160 LET Q$=CHR$ (LEN C$+1)+Q$
 170 LET C$=C$+Q$
 180 CLS
 190 GOTO 120
 200 PRINT "NUMBER OF ENTRY? (0
TO STOP)"
 210 INPUT N
 220 IF N=0 THEN STOP
 230 LET C=1
 240 FOR I=1 TO N-1
 250 IF C>LEN C$ THEN STOP
 260 LET C=C+CODE C$(C)
 270 NEXT I
 280 PRINT N;":";C$(C+1 TO C+COD
E C$(C)-1)
 290 GOTO 200
 300 REM ************************
        TYPICAL INPUT ROUTINE
        ************************
 310 PRINT AT 17,0;"INPUT NUMBER
REQUIRED:";
 320 INPUT Q$
 330 PRINT Q$
 340 PRINT AT 19,0;"IS THAT CORR
ECT? (Y/N)"
 350 INPUT P$
 360 PRINT AT 17,0;"
                   "
 370 PRINT AT 19,0;"
                   "
 380 IF P$(1)<>"Y" THEN GOTO 310
 390 LET N=VAL Q$
 400 REM ************************
        SUBROUTINE FOR PRINTING
        ************************
 410 LET P1=17
 420 LET P2=0
 430 LET P$="NAME OF FUNCTION RE
QUIRED:"
 440 LET Q$=""
 450 REM ************************
 460 REM        THIS SUBROUTINE
        REQUIRES THE FOLLOWING
        TO BE DECLARED BEFORE
        IT IS CALLED:
        1)STRING TO BE PRINTED=
        P$.
        2)PRINT POSITIONS (P1,
        P2).
        3)A COMPLETE LINE OF
        SPACES=Q$
        ************************
 470 PRINT AT P1,P2;P$;
 480 INPUT Q$
 490 PRINT Q$
 500 PRINT AT 19,0;">>";Q$;"<<"
 510 PRINT AT 21,0;"IS THAT CORR
ECT? (Y/N)"
 520 INPUT R$
 530 PRINT AT P1,P2;Q$
 540 PRINT AT 19,0;Q$;Q$;Q$
 550 IF R$(1)<>"Y" THEN GOTO 470
 560 STOP
 570 REM ************************
        PRINTING WITH CODES
        ************************
 580 LET Q$="
           "
 590 PRINT AT 0,0;"NUMBER OF STR
ING TO BE PRINTED? (0 TO STOP)"
 600 INPUT P
 610 IF P=0 THEN STOP
 620 CLS
 630 GOSUB 660
 640 GOTO 590
 650 REM ************************
        THIS SUBROUTINE
        NEEDS ONLY THE NUMBER
        OF THE DESIRED STRING
        TO BE DECLARED.
        ************************
 660 LET P1=CODE A$(P,2)
 670 LET P2=CODE A$(P,3)
 680 PRINT AT P1,P2;Q$
 690 PRINT AT P1,P2;A$(P,4 TO CO
DE A$(P,1));":";
 700 INPUT Q$
 710 PRINT Q$
 720 PRINT AT 19,0;">>";Q$;"<<"
 730 PRINT AT 21,0;"IS THAT CORR
ECT? (Y/N)"
 740 INPUT R$
 750 PRINT AT 19,0;Q$;Q$;Q$
 760 IF R$(1)<>"Y" THEN GOTO 66
 770 PRINT AT P1,P2;Q$
 780 RETURN
 790 REM ************************
        INPUT OF STRINGS
        ************************
 800 PRINT "HOW MANY STRINGS"
 810 INPUT S
 820 DIM A$(S,20)
 830 FOR I=1 TO S
 840 PRINT "STRING NO.";I;":";
 850 INPUT Q$
 860 PRINT Q$
 870 PRINT "LINE FOR PRINTING:";
 880 INPUT P1
 890 PRINT P1
 900 PRINT "COLUMN FOR PRINTING:
";
 910 INPUT P2
 920 PRINT P2
 930 LET Q$=CHR$ (LEN Q$+3)+CHR$
P1+CHR$ P2+Q$
 940 CLS
 950 LET A$(I)=Q$
 960 NEXT I
 970 STOP
 980 REM ************************
```

Spectrum

In this new slot various contributors explore different aspects of the ZX Spectrum.

This is why they called it Spectrum

***Nick Hampshire** discusses the colour commands on the ZX Spectrum*



All the fantastic colours: (due to technical reasons, here only in black and white).

The Spectrum screen is organised as 24 lines of 32 characters, and the character and background colour of each one of these 768 character spaces can be individually programmed to one of the eight possible colours which can be displayed by the Spectrum.

The two colours associated with each character space are the foreground or character colour, this is referred to as the ink colour, and the background colour or paper. In the normal power up mode the INK colour is black and the PAPER colour white.

There are eight different colours, including black and white, which can be displayed, they are as follows:

0 — black
1 — blue
2 — red
3 — purple or magenta
4 — green
5 — pale blue or cyan
6 — yellow
7 — white

These colours are produced on a colour tv by mixing just three primary colours — blue, red and green. Thus magenta, which is colour 3, is produced by mixing colours 1 and 2 — blue and red. Likewise colour 5, cyan, is a mix of colours 1 and 4, and colour 6, yellow is a mix of colours 4 and 2.

From this you can see that the colour number is in fact the sum of the primary colours required to produce that colour. Thus white, which is produced by having all three primary colours mixed, has colour number 7 or colours 1 + 2 + 4.

The number associated with each colour on the above list is important since it is used in the colour commands to designate that colour.

The INK command is used to set the character or foreground colour of characters subsequently displayed using PRINT commands starting at the current cursor position. The command:

```
INK 4 : PRINT "ink colour green"
```

will print the statement "ink colour green" on the screen starting at the current cursor position in green characters on the existing background colour (normally white) of the screen. To show the range of colours try the following program:

```
10 FOR Q = 0 TO 7
20 INK Q
30 PRINT "ink colour number"; Q
40 NEXT Q
```

The PAPER command is identical to the INK command except that it sets the background colour for the printed characters. Thus the command:

```
PAPER 4 : PRINT "paper colour is green"
```

will display the statement "paper colour is green" starting at the current cursor position and using the existing ink colour (normally black). The following short program shows the 64 different combinations of INK and PAPER colours which can be obtained.

```
10 PRINT "01234567 ink colours"
20 FOR Q = 0 TO 7
30 FOR Z = 0 TO 7
40 INK Z: PAPER Q
50 PRINT "*";
60 NEXT Z
70 PRINT "paper colour"; Q
80 NEXT Q
```

Besides the foreground and background colours there is also the colour of the border around the screen display area. This border can have its colour set by use of the BORDER command followed by one of the eight colour code numbers. Thus:

```
BORDER 5
```

will set the border to a cyan colour.

The original INK or PAPER colours can be retained for a character by setting the colour value to 8. This means that the characters printed following the command are "transparent", with the previously defined colours on the screen being used to display the new characters. Thus if the command

```
PAPER 8
```

is executed then the PAPER colour will be left as currently displayed on the text following the cursor. However, the INK colour will be that defined in the previous statement. Similarly the command:

```
INK 8
```

will leave the INK colour unchanged but the PAPER colour changed to that defined in the previous colour definition statement. Both INK 8 and PAPER 8 can be used together to leave all colours unchanged.

There is a very poor contrast between some of the colours. For example it is virtually impossible to read a character which has an INK colour of cyan and a PAPER colour of green.

To overcome this and ensure enhanced character contrast there is an extra character code value. To do this you have to use the colour code number 9 after either the INK or PAPER commands.

These set the colour used with either the defined INK or PAPER colour to a colour with the maximum contrast. Thus if the colour is dark (eg. black, blue, red or magenta), then the complimentary colour will be made white. If light, then the complimentary colour will be black.

Sound & vision

Beep-Beep, Beep Beep, yeah!

Now that the initial excitement of the ZX Spectrum launch is out of the way, and the computers are starting to be used, its functions are beginning to be explored. My first impression was quite good, even though the machines I saw were pre-production models. There are a number of weaknesses but the machine is a vast improvement on its older brother, especially to readers of this column who will be interested to hear that the Spectrum has sound.

Spectrum sound is governed by the BEEP command, which sounds silly but then so do PEEK and POKE. BEEP is used with two parameters, that is the word Beep is followed by two variables, which may be numbers or variable names, separated by a comma. The first one of these parameters governs the duration of the sound, the second its pitch. Duration is specified in seconds. I didn't have the opportunity to test the duration for accuracy, but at a guess it should be OK for most music. After all, notes don't usually extend beyond a couple of seconds.

The pitch variable is interesting. If it is given the value 0 then the pitch is that of middle C. Add one to get the next semitone, ie C sharp or D flat. Adding one always gives the next semitone up, subtracting gives the one lower.

The pitches are so organised to make an octave rise equal an extra twelve added to the pitch value. This continues to rise all the way up to a pitch value of around 73, way beyond my hearing, where an illegal parameter error message is given. I'm sure people can think up some good uses for the very high frequencies, such as disturbing bats and opera singers.

Another nice touch is that these pitches don't have to be integers — in other words quarter tones — and smaller pitch variations can be programmed. This gives rise to two more possibilities.

The first is the playing of Arabic, or Chinese music where the scales are organised differently.

The second is tuning of the Spectrum to other musical instruments. This can be done by ear, adding tiny fractions as an increment until two pitches coincide.

It also raises the possibility of portamento between two notes.

It is quite likely that BEEP is not accurate over more than a couple of octaves, so this limitation should be kept in mind. Also I expect BEEP will be affected greatly by dirty power supplies.

Sam Blythe

The BEEP key on the Spectrum is next to CAPS SHIFT.

The colourful plot thickens

The graphical complexity of the BBC Micro is such as to make it one of the most useful machines around, yet the very wide range of options, permutations and cunning tricks can get rather confusing.

This week, we'll look at one of the most important graphics capabilities and how it can be used.

The feature is called XOR plotting — XOR standing for 'exclusive or'. You probably recognise this as a term from logic, to be grouped with others like AND, OR and NOT. Whilst the latter are fairly easily understood, XOR is more difficult to grasp.

What is easy to appreciate, however, is the fact that it applies to plotting a colour on the screen, and means that the colour you draw with is *modified* by the colour already there, 'underneath it'.

XOR plotting, simply, means that the computer does a quick check on the information already present in the bit of memory looking after each pixcel — individual dot — on the screen.

Normal plotting would just replace whatever information was there with new stuff — hence replacing the old colour with the new. Red might be changed to black, or white made yellow, for example.

But XOR plotting implies that if, say, red is laid over yellow, the result is a new colour altogether. Or — what is even more useful — if a colour is plotted on the screen in XOR mode, then plotted *again*, it disappears. What is more, it vanishes leaving whatever was underneath still intact!

Only a few other machines, such as the RML 38Z, can do this. They let you move shapes (or text) around over an already existing coloured background, leaving the original image just as it was before.

Here's how to use it. Having set up a graphics mode (try MODE 5), you can determine the colour of any plotting commands by the use of GCOL. GCOL needs two numbers following it, separated from each other by a comma. In normal use, the first digit is 0, and the second is 0 to 3, which gives colours black, red, yellow and white (or their monochrome equivalents on a black-and-white tv).

So GCOL 0,3 means 'use colour 3, normally'. But change the 0 to a 3, and you're in XOR mode. GCOL3,3 means 'use colour 3, in XOR mode'.

Next week, I'll be presenting two programs — rather brain-damaging ones — using this and other graphics effects. This week, try the program on the left.

Brian Reffin Smith

```
10 MODE 5
20 CLG : clear graphics area
30 FOR I=1 TO 1000
40 GCOL 3,1 : REM plot in XOR red
50 GOSUB 100
60 GCOL 3,2 : REM XOR yellow
70 GOSUB 100
80 NEXT I
90 END
100 REM plot twice
110 X1=(1239) : Y1=RND(1023)
120 X2=RND(1239) : Y2=RND(1023)
130 FOR J=1 TO 2
140 MOVE X1,Y1 : DRAW X2,Y2
150 NEXT J
160 RETURN
```

Our classifieds are faster.

Do you want to sell your computer and buy a bigger and better one?

Have you ever thought of trying to make some money out of selling tapes of your own programs?

Whatever it is you want to buy or sell why not use our classified pages?

It has to be better than waiting for up to nine weeks to get into one of the old monthly magazines.

Not only that, but our rates are very reasonable.

For private individuals it only costs 20p per word, with a minimum of 10 words.

We can make it so cheap because we charge companies using the classified columns 40p per word.

The classified pages can be used for semi-display advertising.

The cost for this is £10 per single column centimetre, with a minimum charge of £30.

All copy for the classified pages must be pre-paid. (You'll find a handy form on page 22).

Cheques and postal orders should be made out to *Popular Computing Weekly*. Your advertisement should arrive at least two weeks before the publication date.

If you have any queries regarding Classified or semi-display advertising please call Alastair Macintosh on 01-930 3840

Popular Computing Weekly.

The fast one.

Peek & poke

Peek your problems to our address. Ian Beardsmore will poke back an answer.

A QUESTION OF INTERPRETATION

David Geach of Graeme Road, Ross-on-Wye writes:

Q I am fairly new to computing, using a second-hand Atom at the moment. Slowly I am learning the jargon but two things are still confusing me. These are Compiler and Interpreter. I do not seem to understand several descriptions that I have read. If anything it makes things seem more complex. Could you please explain them?

A A compiler de-codes a program in a high level language, such as Basic or Cobol, into machine code or an assembly language. Initially this is comparatively slow, nevertheless once it has been done the program will RUN faster than an interpreter program. This is because the de-coded program will be stored in the memory.

An Interpreter does essentially the same, except that each statement is done individually and not stored. So each statement has to be re-read and re-coded each time it is used. The advantage of the Interpreter is that it uses less memory, as no interim program has to be stored. The disadvantage is that it takes more time. Obviously the continual re-reading needed for the Interpreter, takes more time to RUN, than the stored de-coded program of the compiler.

IT'S ALL PART OF THE EDUCATION

K. Daniels of Poole, Dorset writes:

Q At the recent computer fair in Earls Court, I heard the name MUSE on two different occasions. No one I have asked seems to have heard of them beyond someone who said that they had heard of them and EZUG but did not know what they were. I haven't a clue. Have you?

A Yes. MUSE stands for Micro Users in Secondary Education while EZUG stands for Educational ZX Users' Group. As you can guess they are both concerned with computers in the school. EZUG was formed out of MUSE, and I gather that both groups are quite active within education, having their own news letters and software libraries.

THIS SHOULD RAM IT ALL HOME

Nick Starking of Caister-on-Sea, Norfolk writes:

Q I am writing to you in the hope that you can answer a question (well two really) for me. I am interested in the Commodore Vic-20, but I feel that the 3.5K RAM is too small. I hear that extra RAM is available, but my query is this: Do the 3K, 8K, and 16K RAM cartridges for the Vic-20 fit inside the computer, or is an expansion unit (like the Afron Expansion Unit) necessary?

Also, I have read about the introduction either later this year, or early next year of the Vic-20/16, a 40-column 16K RAM computer which is a big brother to the Vic-20, and the Commodore 64/40, a 40-column, 64K RAM computer which will sell for about £395. Is there an approximate price available for the Vic-20/16?

A The extra RAM cartridges for the Vic-20 are external, however, a memory expansion port is already supplied, so an expansion unit is not needed unless you want to add other peripherals as well. As for the new Vics, if you look at your third issue of *Popular Computing Weekly* you will find your question answered on page 5. The Vic-20/16, is in fact the Vic-30. Cost will be about £250, and it is due to be launched in January next year.

PUT MORE POKE IN YOUR RACER

J. R. Johnson of Tottenham, London writes:

Q I have had a BBC model B micro since early this month and now I'm writing a *Grand Prix* game. I have tried tabbing the cars on to the screen, but this slows the game down. I would prefer to PEEK and POKE to and from a screen location. Could you tell me and many other BBC owners how to use PEEK and POKE to and from a screen location?

A The first thing that has to be done is that the SCROLL function has to be stopped, or at least controlled by setting up a text window. This still scrolls the screen, but the VDU RAM locations do not change. Enter this:

```
VDU 28,0,24,0,39,0
```

This sets up a screen window for the entire screen. To POKE use the following:

```
?(HIMAN x+y*40) ASC" character you want "
```

Here x and y are the co-ordinates that you want. To PEEK use the following:

```
CH=(HIMAN+x+y*40)
```

This makes CH equal to whatever is at x,y. When you want to bring the character on to the screen just enter the line:

```
CH$=CHR$ "CH"
```

STRICTLY FOR THE KNOB TWIDDLER

B. W. Bailey of Hampstead, London NW3 writes:

Q As a display for my ZX81 I am using a Toshiba model 10TB battery/mains portable with a 9in screen. It has an integral loop antenna marked, and a co-axial socket marked, into which I plug my ZX81. My problem is that no amount of tuning or setting of the contrast or brilliance controls gives me a clear background but a pattern of alternating light and dark lines persists over the usable area. Can you help me?

A Several things could be the cause of the trouble but no one factor presents itself as the most likely cause of the problem. There are two important things that you do not say in your letter. Have you tried your ZX81 with another television, or another ZX81 on your portable? Also I would guess that when you say background that you are at least getting a curser. If the tuning is all right, then two possible causes are the power lead, and the coax aerial lead.

The power supply jack can be very fickle on both the ZX80 and 81, try twiddling this in and out. The smallest increment in the right direction can make a vast difference. In the same way check your video lead. When I first got mine the two wires inside one of the plugs were so badly wired that the slightest pressure would cause them to touch, with all the attendant screen decay.

I would have thought that the internal antenna would be cut out as soon as an external lead was connected, but my hardware knowledge, particularly of televisions, is not all it might be. Try using another television, or computer, this will help reduce the number of possible causes of your troubles. Then try checking all the leads, making sure the power lead does not cross the signal lead, if you are using a RAM Pack, try it without the Pack as they usually add to problems like this.

If you still do not get any luck, then all I can suggest is that you go to your local electrical shop and ask their advice, and possibly if you might try out your computer on one or two other models.

If you still get the same sort of problem, then it would seem that the frequency modulator in the ZX81 is at fault, which will mean a return to Sinclair Research. If it works with other televisions then your Toshiba is the cause, and I could not tell you how to rectify that.

● *Stop agonising over that problem. Write to Ian Beardsmore, Peek & Poke,* Popular Computing Weekly, *Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.*

Competitions

Puzzle No 7

One of the side-stalls at our summer fete was attracting some attention. Called *Lucky Seven*, it was the simplest of games, requiring only nine wooden discs — plain on one side, numbered on the other from 1 to 9.

The nine discs were placed face down on a table and were mixed up. For the payment of a 10p stake you could pick up four of the discs at random, which were then turned over to reveal the digits painted on the reverse. The person in charge would then arrange these four digits to form one four-digit number. If this number was divisible by seven then you lost 10p. If, however, it was impossible for a multiple of seven to be formed then you would win £1.

How would you assess the odds against winning this game? (Of course, such 'tricks' as inverting the six and nine are not allowed.)

Solution for June 4

In order for a man to divide the pile of coconuts into equal fifths and have one left over for the monkey the formula is:

$$A = 4/5\ (B - 1)$$

where B equals the number the pile originally contained and A those remaining after the division. A and B, of course, must be integers. By rearranging this equation we get:

$$B = (5*A/4) + 1$$

In order for the second man to be able to divide these remaining nuts equally (and have one left over for the monkey), B − 1 must also be exactly divisible by five. If it is, we can repeat the procedure, and so on.

As the final number of nuts must be a multiple of five we start with this number and increase by five each time. (To find the answer to part (b) of the question then we must start with a minimum of six to have one left over.)

```
10 LET N = 5
20 LET M = 0
30 LET A = N
40 LET B = 5*A/4 + 1
50 IF (B - 1)/5 = INT ((B - 1)/5) THEN GOTO 100
60 LET N = N + 5
70 GOTO 20
100 LET M = M + 1
110 IF M = 5 THEN PRINT B
120 IF M = 5 THEN STOP
130 LET A = B
140 GOTO 40
```

Run this and you get (a) 3121 coconuts, (b) 15,621 coconuts.

Winner of Puzzle No 3

The winner is: David Robinson, Montgomery Hill, Frankby, Wirral, who receives £10.

Solution to Crossword No 3

Across: 3 CPU, 8 Adder, 9 Shampoo, 10 Chop, 11 Gridiron, 13 Lie low, 14 Duplex, 17 Tropical, 19 Anal, 21 Real Ale, 22 Metro, 23 Lip.

Down: 1 Calculators, 2 Odd ode, 3 Cry, 4 Users, 5 Eardrum, 6 Spar, 7 For next loop, 12 Nominal, 15 Length, 16 Panel, 18 Opal, 20 Amp.

Winner of Crossword No 3

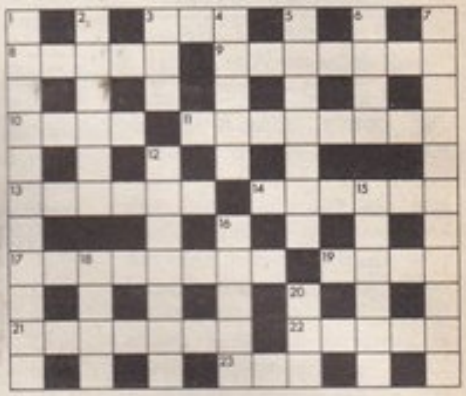
The winner is: D D R Sibbald, St. George's Road, Ilford, Essex, who receives £10.

Rules

The winner for the crossword and the winner of the puzzle will be the first name out of the hat (in each case).

Closing date for both the crossword and the puzzle is the Monday, three weeks after the cover date.

Crossword No 7



ACROSS

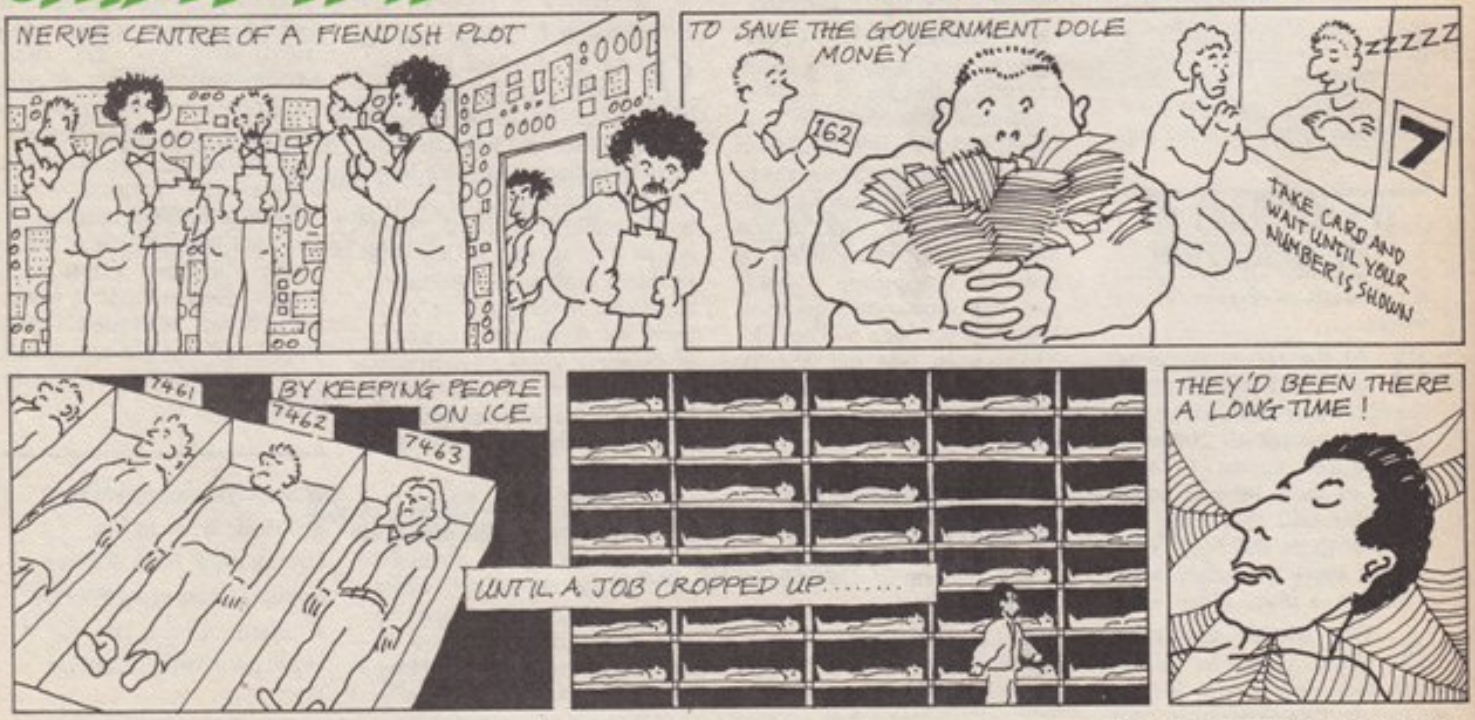
8. Program modules that agree with directions (5).
9. Program the company to come back and press around (7).
10. Program to change source of specified item (4).
11. Program to translate ROM chip, with some help (8).
13. Program bit swapping and calling (6).
14. Three, to start with, in endlessly nice acid (6).
17. Replace a switching centre (8).
19. Program switch may go through 17A (4).
21. Program running order — put to death! (7).
22. Disable short record in a week (5).
23. Slippery when caught on the lee shore (3).

DOWN

1. Program to translate, table and reprint (11).
2. Computing sum of a penny ring (6).
3. A quiet tree (3).
4. Program storage raises more page inserts (1.1.1.1.1).
5. Food container, shaken, inputs nothing (4,3).
6. Storage unit for church students (4).
7. Head underground to avoid the facts (7-4).
12. Nu-speak versions get closer without being observed (5,2).
15. Student of an input device (6).
16. Say yes to nuclear reactor, electrical engineer (5).
18. Credit opposite points of the workers (4).
20. Tool amends law (3).

CITIZEN PAIN

BY DAVID IRELAND and JAMES MACDONALD



commodore

COMPUTING

Own or use a Pet or a Vic? Fed up with being ignored by all the traditional monthly magazines? Fed up with listings which are too simple or simply do not work?

You need Commodore computing, the new monthly magazine. It is published by Nick Hampshire, author of The Pet Revealed, Pet Graphics, A Library Subroutines and the Vic Revealed. Each issue is packed with advanced advice on how to make the most of your computer, whether you use a Pet or a Vic, and whatever your application. Software, hardware, machine code, games, business use — it is all covered in every issue.

If you want to learn more about your computer, take out a subscription to Commodore Computing. It is the only way to get it, and get it straight.

Send £12.50 for a subscription for 1 year (10 issues) to:

COMMODORE COMPUTING, MAGSUB, OAKFIELD HOUSE, PERRYMOUNT ROAD, HAYWARDS HEATH, SUSSEX RH16 3DH

MORE FUN WITH YOUR ZX81!

16K RAM PACK

PUSH-ON PLUG-COMPATIBLE MEMORY

The addition of our fully-compatible, assembled, tested and guaranteed 16K Ram Pack means **more memory for better games and programs**

ORDER YOURS TODAY FROM:

CAPS LTD. Dept. E, 28 The Spain, Petersfield, Hants GU32 3LA

Please send me my Byg Byte 16K Ram Pack. I enclose cheque/ postal order for £34.95 (includes VAT, Postage & Packing)

Name .....

Address .....

.....

.....

Dept. E

Allow 28 days for delivery

Fully **£34.95** inclusive

WE CHALLENGE YOU, EARTHLING.

Load this COSMOS program in your ZX81 16K and see if you can defend your despicable Earth convoys against our supreme attack ships and warp mines.

A totally new space game concept written in machine code to produce fiendishly clever effects and deadly results.

You poor Earthlings have no chance.

£5.99 ON CASSETTE.

PLUS OTHER 16K PROGRAMS ALSO ON CASSETTE

TRIAD *Three new puzzles to perplex and frustrate even the Rubik Cube expert. All with SAVE facility.* ***£3.99***

QUARTET *An exciting compendium of crosswords, puzzles and memory tests with up to 10 grades of difficulty.* ***£3.99***

TABLE TUTOR *Using concepts within a schoolchild's experience, this menu-driven tutor includes instruction, table lists, speed tests, homework and answer service.* ***£3.99***

PET LISTINGS
Black Box; Mastermind; Pontoon.
£1.60 each or 3 for £4.00

Only high quality cassettes used
Send cheque or p.o. to:

VORTEX SOFTWARE (DEPT POP)
26 CRAWFORD ROAD,
HATFIELD, HERTFORDSHIRE, AL10 0PG.

Vortex SOFTWARE

MAIL ORDER ONLY PLEASE. SEND SAE FOR CATALOGUE.

NEW

ZX 81 16K SOFTWARE

DIGGLES KITCHEN

SIMPLE SUPPERS TO CELEBRATION DINNERS

VOLUME 1

50 PAGES OF WORLDWIDE RECIPES

£4.99 (inc. P&P and VAT)

VOLUME 2

50 PAGES EUROPEAN RECIPES

£4.99 (inc. P&P and VAT)

Special price for two volumes

£9 (inc. P&P and VAT)

More volumes to follow

Please specify which volume(s) – Mail order only

Send remittance to:-

MICRO COMPUTER SOFTWARE
Unit D6, Pear Industrial Estate, Stockport Road, Lower Bredbury, Stockport SK6 2BP. Tel: 061-494 2441