

YOUR

SEPTEMBER 1989 £1.30

COMMODORE

BATMAN

Waiting for a dark knight

GREENPEACE

Fight for the Foliage

GAMES

- Thunderbirds
- Rockstar ate my Hamster
- Escape from Drill
- Phobia

LICENCE TO KILL

007 – On his own and out for revenge



ISSN 0269-8277



THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...
NOW ACTION REPLAY Mk V PROFESSIONAL
 FOR CBM64/128 HAS ARRIVED

ONLY
£34.99
 POST FREE

AR.V
 THE ULTIMATE
 BACKUP CARTRIDGE
 IS HERE!

EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

ALL FEATURES AVAILABLE TO TAPE OR DISK USERS.

ACTION REPLAY Mk V differs from all other cartridges because it contains an 8K ROM with a FULL 128K operating system ROM. This means that you always use COMPLETELY ORIGINAL 8K ROMS.

BACKUP/RECALL systems use traditional technology which severely limits performance. Action Replay's mode of file set backup gives you 100% FILE SET backup success. Action Replay's mode may differ slightly from other competitors, but are just what you need for the business...

- ✓ **FORMER RELEASE** Action Replay Mk II has 8 unique "Folder Lockers" - "Remember" & "Way Off". Both work up to 20 times normal speed. ✓ **PROTECT AGAIN** tracks & maps all files on the system. ✓ **PROTECT AGAIN** uses any other multi sector method to store all data parts of a file set. ✓ **EXACT REPRODUCTION** full range of commands with 8K resolution. ✓ **REPRODUCIBLE COMMANDS** an extremely reliable built-in program.
- ✓ **DISK COMPACTOR** uses advanced program compression technology. ✓ **FAST BACKUP** 120 times normal, 200 times normal for a backup program. ✓ **DISK COPY** full disk operation that also backs up full disk operations. ✓ **EXACT DATA REPRODUCTION** you don't need a disk drive for the new backup feature. ✓ **TODAY'S COMMANDS** full range of all the new CBM commands. ✓ **EXACT REPRODUCTION** compatibility with software running on 128K ROMS (including disks).
- ✓ **COMPARABLE REPRODUCTION** for previous releases. ✓ **PROFESSIONAL, MACHINE CODE MONITOR** Full 8K monitor available on all discs with full range of commands.

TOTAL BACKUP POWER

The most powerful backup system ever devised. Uniquely allows system tracks at any point. And gives the single feature to backup your programs.

- **SAVE** - 100 files & more folders
- **DIRS** - 10 DIRS & more sub DIRS
- **FILE** - 100 files

NOTE: Use a reliable program compressor. Single point save at 10000 speed is 40K or more. Backup Professional (SUPERVISE) is available for "program log" of user knowledge required.

THE PROGRAM IS AVAILABLE... JUST OFFER THE NAME & YOUR NAME.

WHAT THE REVIEWERS SAY

"I'd always wanted excellent backup equipment... This is really the best value for money cartridge... (for Cartridge World)"

Cartridge World Team

UPGRADE INFORMATION

If you Professional to Mk II Professional, just need 20.00 & we will send you a tape Mk II operating system that the price for your original Mk II (included) to Mk II Professional. Also you old cartridge plus 25.00 & we will upgrade to (value 21.00).

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HOURS

HOW TO ORDER ...

<p>BY PHONE</p> <p>0202 764377</p> <p>9.00am - 5.00pm (incl. VAT)</p>	<p>BY POST</p> <p></p> <p>Next day plus 100p extra payable to "Cartridge World"</p>	<p>FAX</p> <p>0792 744399</p> <p>ON ORDERS COST MORE INCLUDES POST & PACKAGING CHARGE</p>
--	---	--

PRICE AND SPECIFICATION CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

0202 764377/0202 764399 - Please contact goods by telephone prior to visit.



DAtEL ELECTRONICS LTD., FERRIS INDUSTRIAL ESTATE
 GOWAN ROAD, FERRIS, STONE-ON-TRENT, ENGLAND.

SALES ONLY 0792 744377
TECHNICAL ONLY 0792 744399

PERFORMANCE PROMISE

Action Replay has an unrivalled range of "recovery features". When you use Action Replay, if you don't find your system to be what you want, then return it within 10 days for a full refund.

GRAPHICS SUPPORT UTILITIES DISK

8.000 8.0000 - Give your favourite access to a whole new type disk.

BLAST UP - A unique utility to allow you to take any part of your picture & blow it up to full screen size.

GRAPH EDITOR - A complete picture editor helps you to create or edit graphics.

REVERSE ENGINEER - Take your favourite screen screen with a graphics package or replace it with Action Replay if you're tired of scrolling screen message displays with users.

ONLY £12.99

NOTICE 1988 COPYRIGHT ACT

DAtEL ELECTRONICS LTD. neither authorizes nor condones the use of its products to reproduce copyright material. It is illegal to make copies of such material without the expressed consent of the copyright owner or their licensee.



CONTENTS

007[™] LICENCE TO KILL



• Loose Wolf The game of the gamebooks.	8	• 128 Career Are you a troubled 128 user?	20
• James Bond Domains is back.	17	• PC Corner The regular series for the Commodore PC.	26
• Batman At 50 is Barry just 10?	22	• Next Month	45
• Greenpeace We've got atoms. We've got radiation...	28	• Big Wobbly Gumps We loads o' Thunderbirds goodies.	37
• News	4	• Software for Sale You need to type!	70
• Contributions So lad, you'd like to write for us?	12	• Extending Basic Part six sixth.	76
• Four Corners Plus 4 writers stop here.	14	• Back Page Hot jazz group and The Nitties.	82

• Thunderbirds 1... 2... 3... 2... 1...	62	• Chess Tiddly Three days of Side Playing.	69
• Gilbert Get smart!	64	• Tweezer Time travelling tweezers.	72
• Sage's Scramble See that engine.	65	• Fire King What, more role playing?	73
• Red Storm rising Nuclear conflict, comrades.	66	• Phobia I can't stand it any more. I've got Coder phobia.	74
• Shane Lord Can you be a hero?	67	• Journey to the Centre of the Earth Can it's with hot lava lava lava lava!	75
• Backdoor Ate My Hamster And it were right tasty.	68		

• Digs Get your hands round your tool.	29	• Print FX Correction for our previous errors.	83
• Boxes and Squares Tutorial into mathematical formulae.	42	• DOS 6.1 Enhanced DOS commands for disk users.	54
• Processing on the J28 Continuation of the expert system.	68		

FEATURES

REGULARS

PROGRAMS

GAMES

Editorial

If you flick through the pages of this month's *Four Commodore* you may find a few subtle (and a couple of blatantly obvious) changes. This is because the editorial team, upon my arrival to the field, has decided to offer a brighter future for FC, one that we can all enjoy together.

My first major decision was to revive this issue around *Burnas*, and I hope you all agree that this was a wise move. Apparently the movie grossed over \$40,000,000 in its first weekend, and seeing as it cost \$10m less to make it has more than doubled any fears from Warner Brothers that it was to flop, and this is before it reaches British shores. Further on in this issue you can read about the reasons the film was undertaken, and a preview of the game.

Other features of note are on *Lavigne* to *Kill* - the new Bond film, *Graffiti* and *Loch Waul*, to make sure you've got your adventure gloves on, they are HOT!

Although there are many entertainment based articles we haven't left out the usual listings, or the machine based columns. The programs can now be found in a section of with their own, so you can keep them separate from the rest.

I hope you enjoy this issue as much as we've enjoyed putting it together and that leaves me to say, welcome to the beginning...

Kit Henderson

Deputy Editor: **Michael 'Boy Wonder' Henderson**

Technical Editor: **Paul 'Riddler' Eves**

Group Editor: **Shaun 'Penguin' Cooke**

Advertisements Manager: **Paul 'Raycast' Kavanagh**

Sales Executive: **Marie 'Caracosse' Ward**

Ad Copy Control:

Karen 'Maggit' Baylyford

Artist: **Alan 'Mudley' Baylyford**

Reviewer: **Alan 'Kobay' Lohreid**

Designer: **Mark 'Dark Knight' Bennett**

Originator: **Shony**

Printer: **Chase Web**

The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights reserved by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction

requires the prior written consent of the Company. 1989. Distribution: EM Distributors, 4 Lighthouse Court Road, London SW16 2JG. Printed by Chase Web, Plymouth.

Opinions expressed in reviews are the opinions of the reviewers and are necessarily those of the magazine. While every effort is made to thoroughly check programs to thoroughly check programs to be published, we cannot be held responsible for any errors that do occur.

Your Commodore incorporating Four 48 is a monthly magazine appearing on the first Friday of each month. Argus Specialist Publications Limited, Editorial and Advertisement office, Four Commodore, Argus House, Boundary Way, Bazel Woodway, HP2 7SD Telephone: (0443) 64001. Subscription rates upon application to Your Commodore Subscription Department, Inform Ltd., 5 River Park Estate, Bockhampton, Basingstoke, Hampshire, RG24 1RL. Telephone: (0945) 812276. U.S.A. Subscription Agent: Wise One Worldwide Publications, 4114 West 238th Street, Torrance CA 90505 U.S.A.

Data Statements

The Bug Bites Back

Now that *The Bug*, the incredibly successful computer journal, has officially folded, editors Jason Lewis and Jeffrey Davey are to sell their entire software review library to raise funds for charity.

The collection includes five years of software reviews and other material and a majority of the profits are to go to a consortium of separate charities and voluntary organisations.

If you wish for more information, the current list, etc., write to 28 Lonsdale Avenue, Maffei Hill, London NW10 3BU. Please include an SAE.

EA's Horror Story.

Electronic Arts is to bring the horrific world of H P Lovecraft's *Cthulhu* mythos onto Amiga and PC screens in September. The *House of the Shadow* is a role playing game set in the roosting 20s, in which the player must use investigation research in order to unravel the secrets of arcane lore and discover many odious horrors lurking beneath seemingly normal life.

The game system that was developed by board game designers and fiction writers, Edostan Gianni, allows the player to create a character by selecting sex, nationality, one of six professions, and proficiency in over 20 skills. These will directly affect the outcome and his success in the game.

The *House of the Shadow* will appear first on the PC in early September and later on the Amiga. Both versions will cost £24.99.



OFFICIAL COMMODORE/MANICA DEALER

COMMODORE	
4000	£199.95
4000 286	£249.95
4000 386	£299.95
4000 486	£349.95
4000 586	£399.95
4000 686	£449.95
4000 786	£499.95
4000 886	£549.95
4000 986	£599.95
4000 1086	£649.95
4000 1186	£699.95
4000 1286	£749.95
4000 1386	£799.95
4000 1486	£849.95
4000 1586	£899.95
4000 1686	£949.95
4000 1786	£999.95
4000 1886	£1049.95
4000 1986	£1099.95
4000 2086	£1149.95
4000 2186	£1199.95
4000 2286	£1249.95
4000 2386	£1299.95
4000 2486	£1349.95
4000 2586	£1399.95
4000 2686	£1449.95
4000 2786	£1499.95
4000 2886	£1549.95
4000 2986	£1599.95
4000 3086	£1649.95
4000 3186	£1699.95
4000 3286	£1749.95
4000 3386	£1799.95
4000 3486	£1849.95
4000 3586	£1899.95
4000 3686	£1949.95
4000 3786	£1999.95
4000 3886	£2049.95
4000 3986	£2099.95
4000 4086	£2149.95
4000 4186	£2199.95
4000 4286	£2249.95
4000 4386	£2299.95
4000 4486	£2349.95
4000 4586	£2399.95
4000 4686	£2449.95
4000 4786	£2499.95
4000 4886	£2549.95
4000 4986	£2599.95
4000 5086	£2649.95
4000 5186	£2699.95
4000 5286	£2749.95
4000 5386	£2799.95
4000 5486	£2849.95
4000 5586	£2899.95
4000 5686	£2949.95
4000 5786	£2999.95
4000 5886	£3049.95
4000 5986	£3099.95
4000 6086	£3149.95
4000 6186	£3199.95
4000 6286	£3249.95
4000 6386	£3299.95
4000 6486	£3349.95
4000 6586	£3399.95
4000 6686	£3449.95
4000 6786	£3499.95
4000 6886	£3549.95
4000 6986	£3599.95
4000 7086	£3649.95
4000 7186	£3699.95
4000 7286	£3749.95
4000 7386	£3799.95
4000 7486	£3849.95
4000 7586	£3899.95
4000 7686	£3949.95
4000 7786	£3999.95
4000 7886	£4049.95
4000 7986	£4099.95
4000 8086	£4149.95
4000 8186	£4199.95
4000 8286	£4249.95
4000 8386	£4299.95
4000 8486	£4349.95
4000 8586	£4399.95
4000 8686	£4449.95
4000 8786	£4499.95
4000 8886	£4549.95
4000 8986	£4599.95
4000 9086	£4649.95
4000 9186	£4699.95
4000 9286	£4749.95
4000 9386	£4799.95
4000 9486	£4849.95
4000 9586	£4899.95
4000 9686	£4949.95
4000 9786	£4999.95
4000 9886	£5049.95
4000 9986	£5099.95
4000 10086	£5149.95
4000 10186	£5199.95
4000 10286	£5249.95
4000 10386	£5299.95
4000 10486	£5349.95
4000 10586	£5399.95
4000 10686	£5449.95
4000 10786	£5499.95
4000 10886	£5549.95
4000 10986	£5599.95
4000 11086	£5649.95
4000 11186	£5699.95
4000 11286	£5749.95
4000 11386	£5799.95
4000 11486	£5849.95
4000 11586	£5899.95
4000 11686	£5949.95
4000 11786	£5999.95
4000 11886	£6049.95
4000 11986	£6099.95
4000 12086	£6149.95
4000 12186	£6199.95
4000 12286	£6249.95
4000 12386	£6299.95
4000 12486	£6349.95
4000 12586	£6399.95
4000 12686	£6449.95
4000 12786	£6499.95
4000 12886	£6549.95
4000 12986	£6599.95
4000 13086	£6649.95
4000 13186	£6699.95
4000 13286	£6749.95
4000 13386	£6799.95
4000 13486	£6849.95
4000 13586	£6899.95
4000 13686	£6949.95
4000 13786	£6999.95
4000 13886	£7049.95
4000 13986	£7099.95
4000 14086	£7149.95
4000 14186	£7199.95
4000 14286	£7249.95
4000 14386	£7299.95
4000 14486	£7349.95
4000 14586	£7399.95
4000 14686	£7449.95
4000 14786	£7499.95
4000 14886	£7549.95
4000 14986	£7599.95
4000 15086	£7649.95
4000 15186	£7699.95
4000 15286	£7749.95
4000 15386	£7799.95
4000 15486	£7849.95
4000 15586	£7899.95
4000 15686	£7949.95
4000 15786	£7999.95
4000 15886	£8049.95
4000 15986	£8099.95
4000 16086	£8149.95
4000 16186	£8199.95
4000 16286	£8249.95
4000 16386	£8299.95
4000 16486	£8349.95
4000 16586	£8399.95
4000 16686	£8449.95
4000 16786	£8499.95
4000 16886	£8549.95
4000 16986	£8599.95
4000 17086	£8649.95
4000 17186	£8699.95
4000 17286	£8749.95
4000 17386	£8799.95
4000 17486	£8849.95
4000 17586	£8899.95
4000 17686	£8949.95
4000 17786	£8999.95
4000 17886	£9049.95
4000 17986	£9099.95
4000 18086	£9149.95
4000 18186	£9199.95
4000 18286	£9249.95
4000 18386	£9299.95
4000 18486	£9349.95
4000 18586	£9399.95
4000 18686	£9449.95
4000 18786	£9499.95
4000 18886	£9549.95
4000 18986	£9599.95
4000 19086	£9649.95
4000 19186	£9699.95
4000 19286	£9749.95
4000 19386	£9799.95
4000 19486	£9849.95
4000 19586	£9899.95
4000 19686	£9949.95
4000 19786	£9999.95
4000 19886	£10049.95
4000 19986	£10099.95
4000 20086	£10149.95
4000 20186	£10199.95
4000 20286	£10249.95
4000 20386	£10299.95
4000 20486	£10349.95
4000 20586	£10399.95
4000 20686	£10449.95
4000 20786	£10499.95
4000 20886	£10549.95
4000 20986	£10599.95
4000 21086	£10649.95
4000 21186	£10699.95
4000 21286	£10749.95
4000 21386	£10799.95
4000 21486	£10849.95
4000 21586	£10899.95
4000 21686	£10949.95
4000 21786	£10999.95
4000 21886	£11049.95
4000 21986	£11099.95
4000 22086	£11149.95
4000 22186	£11199.95
4000 22286	£11249.95
4000 22386	£11299.95
4000 22486	£11349.95
4000 22586	£11399.95
4000 22686	£11449.95
4000 22786	£11499.95
4000 22886	£11549.95
4000 22986	£11599.95
4000 23086	£11649.95
4000 23186	£11699.95
4000 23286	£11749.95
4000 23386	£11799.95
4000 23486	£11849.95
4000 23586	£11899.95
4000 23686	£11949.95
4000 23786	£11999.95
4000 23886	£12049.95
4000 23986	£12099.95
4000 24086	£12149.95
4000 24186	£12199.95
4000 24286	£12249.95
4000 24386	£12299.95
4000 24486	£12349.95
4000 24586	£12399.95
4000 24686	£12449.95
4000 24786	£12499.95
4000 24886	£12549.95
4000 24986	£12599.95
4000 25086	£12649.95
4000 25186	£12699.95
4000 25286	£12749.95
4000 25386	£12799.95
4000 25486	£12849.95
4000 25586	£12899.95
4000 25686	£12949.95
4000 25786	£12999.95
4000 25886	£13049.95
4000 25986	£13099.95
4000 26086	£13149.95
4000 26186	£13199.95
4000 26286	£13249.95
4000 26386	£13299.95
4000 26486	£13349.95
4000 26586	£13399.95
4000 26686	£13449.95
4000 26786	£13499.95
4000 26886	£13549.95
4000 26986	£13599.95
4000 27086	£13649.95
4000 27186	£13699.95
4000 27286	£13749.95
4000 27386	£13799.95
4000 27486	£13849.95
4000 27586	£13899.95
4000 27686	£13949.95
4000 27786	£13999.95
4000 27886	£14049.95
4000 27986	£14099.95
4000 28086	£14149.95
4000 28186	£14199.95
4000 28286	£14249.95
4000 28386	£14299.95
4000 28486	£14349.95
4000 28586	£14399.95
4000 28686	£14449.95
4000 28786	£14499.95
4000 28886	£14549.95
4000 28986	£14599.95
4000 29086	£14649.95
4000 29186	£14699.95
4000 29286	£14749.95
4000 29386	£14799.95
4000 29486	£14849.95
4000 29586	£14899.95
4000 29686	£14949.95
4000 29786	£14999.95
4000 29886	£15049.95
4000 29986	£15099.95
4000 30086	£15149.95
4000 30186	£15199.95
4000 30286	£15249.95
4000 30386	£15299.95
4000 30486	£15349.95
4000 30586	£15399.95
4000 30686	£15449.95
4000 30786	£15499.95
4000 30886	£15549.95
4000 30986	£15599.95
4000 31086	£15649.95
4000 31186	£15699.95
4000 31286	£15749.95
4000 31386	£15799.95
4000 31486	£15849.95
4000 31586	£15899.95
4000 31686	£15949.95
4000 31786	£15999.95
4000 31886	£16049.95
4000 31986	£16099.95
4000 32086	£16149.95
4000 32186	£16199.95
4000 32286	£16249.95
4000 32386	£16299.95
4000 32486	£16349.95
4000 32586	£16399.95
4000 32686	£16449.95
4000 32786	£16499.95
4000 32886	£16549.95
4000 32986	£16599.95
4000 33086	£16649.95
4000 33186	£16699.95
4000 33286	£16749.95
4000 33386	£16799.95
4000 33486	£16849.95
4000 33586	£16899.95
4000 33686	£16949.95
4000 33786	£16999.95
4000 33886	£17049.95
4000 33986	£17099.95
4000 34086	£17149.95
4000 34186	£17199.95
4000 34286	£17249.95
4000 34386	£17299.95
4000 34486	£17349.95
4000 34586	£17399.95
4000 34686	£17449.95
4000 34786	£17499.95
4000 34886	£17549.95
4000 34986	£17599.95
4000 35086	£17649.95
4000 35186	£17699.95
4000 35286	£17749.95
4000 35386	£17799.95
4000 35486	£17849.95
4000 35586	£17899.95
4000 35686	£17949.95
4000 35786	£17999.95
4000 35886	£18049.95
4000 35986	£18099.95
4000 36086	£18149.95
4000 36186	£18199.95
4000 36286	£18249.95
4000 36386	£18299.95
4000 36486	£18349.95
4000 36586	£18399.95
4000 36686	£18449.95
4000 36786	£18499.95
4000 36886	£18549.95
4000 36986	£18599.95
4000 37086	£18649.95
4000 37186	£18699.

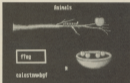
Trendy Lefties in Fun School Controversy

Fun School 2, Duxbury's excellent teaching software, has been rejected for use in schools by 'Trendy' authorities, who believe that it emphasizes the 3Rs too much.

The 3Rs (Reading, Writing, and 'Rithmetic) are currently being weeded out of the national curriculum by such left-wing educationists for the 'old fashioned' methods of teaching used.

Peter Davidson, one of the co-authors of the software, states "What they don't like about it - apart from it being based on the 3Rs - is that it involves an element of competition, with a clock coming up if the question is answered correctly".

Shelley Gibson, a teacher who has been using **Fun School 2** in her class at Poplar Street Primary School, Audenshaw, has written to the Minister of Education, Kenneth Baker, to urge him to explore matters further.



PC Formula One.

Electronic Arts is to release a PC version of its best selling Amiga racing game, **Ferrari Formula One**. Planned for an early August date, the game aims to bring together the excitement of driving in a **Formula One** championship against Alain Prost, Nigel Mansell and Ayrtton Senna and the strategy of managing a team.

The game will feature all 16 tracks from the 1984 season including Monaco, Brazil's Hung and Monza

plus two additional tracks so that the player can compete in the current season. Each track is modified accurately down to its length, the background and the weather conditions the drivers will face.

The races themselves aim to put you in the middle of the action as you attempt to drive your Ferrari to victory. **Ferrari Formula One** will cost £29.99. Available on both 3.5 and 5.25 disks it will support both CGA and EGA graphics.

Socket to Them

Computers, monitors and TVs are all vulnerable to mains interference. This can damage hardware, corrupt data, and cause games to crash, not ideal when the high score is about to be broken. The most common type of interference is a mains 'spike', provided by home equipment such as washing machines, power tools, and microwave ovens. The same applies with your neighbour's electrical equipment, or even the Electricity company having problems maintaining a stable voltage.

One solution to such a problem comes from Apollo Electronic Products Ltd, who have introduced a new Apollo Spike Suppressor which, for the price of £12.50



for a plug unit, can protect your computer system. These British made devices are simple to use, the most you have to do is replace the mains plug.

The range also includes a 3-way adapter for £16 and a 4-gang strip for £35 (prices include P&P and VAT). All the units are capable of absorbing a spike of 4500 Amps/215 Joules.

For more information contact Apollo Electronic Products Ltd, on (04458) 8136.

Game, Set and Mat

Your Commodore are proud to present mouse mats and dust covers for the A500 and C128. The covers are especially designed to protect your computer from dust, grime, coffee, tea, etc, ensuring your screens do not get zapped by appliances and internal acid rain.

Made exclusively for your favourite computer range, they are stylish and robust, and come complete with

their own nifty logs.

The mouse mats too are emblazoned with the 'Your Commodore' logo, and they help to allow your mouse grip on any surface.

At the price of £4.45 for the covers and £8.45 for the Mouse Mat, they represent terrific value for money.

They are available from: **Readers Services, Argus House, Boundary Way, Hemel Hempstead HP1 1BT.**



It's A-Maze-ing

Brown's latest game comes our way in late August. Titled *Mazemonster*, it features a character called Flippo who must travel through the maze of Tubular Balls mazing tiles as he goes - hence his name Flippo.

Gliding alone hinder his quest, but there are items to supply him with extra power. Once granted, the maze will fade away to leave Flippo with yet another to

tackle.

The Amiga version will have 30 levels to complete, while the C64 will only feature 12-16. The price of the C64 version will be (on cassette) \$9.99.



Anyone for Tennis

Microsoft, under their Imageworks label, has secured the rights for the Sega coin-op, *Passing Shot*. Programmed by Tapes - the team behind *Thunderbirds* and *Masteroids*, among others - it captures all the options of the arcade

machine and will be available on both the C64 and the Amiga.

Passing Shot features doubles and singles championships, taking place on clay and grass courts, in an effort to win the elusive Grand Slam. It offers an impressive variety of gameplay options with the player controlling both the posi-

tioning and type of shot taken, lob, spin, etc.). The court scrolls to allow you to follow the action and there are two perspectives to encounter during the game - overhead for gameplay, HD for service.

Ball boys, the net judge and the line-man are all there and we can expect to see it in late Autumn.



Oil Sec You in the Morning

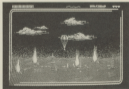
Rainbow Arts, the company responsible for the scandalous *Duress/Katalka*, has undertaken a different kind of project.

Released on the Poline Software label, *Oil Imperium* deals with the cut and throat world of the oil business. You can become J.R.

as you try to put your competitors out of business. What is even better is that their enemies can be your friends, or the computer.

Bullying, selling, trading, spying and sabotaging are all involved in getting to the top, so only the devious minds may apply.

Oil Imperium will be available in September on the C64 and Amiga!



Only the Lonely

The Lone Wolf bandwagon continues to roll along the road of success. Rick Henderson hitches a ride and prepares for a one-way trip to fantasy land.



The Lone Wolf Legends are a new series of books by Joe Devry, who hopes to revitalize the success of previous efforts. Co-written with John Grant — whose credits to date include *Demons*, *Evilbeasts* (co-written with David Langford) and, strange as it may seem,

the *Encyclopedia of Walt Disney's Animated Characters* — they are not short of technical brilliance, although one wonders whether the fantasy-accented plot is a little too clichéd.

Edge of the Kai is the first in the indeterminate number of books. It deals with the origin of Lone Wolf

plus the downfall of the Kai, the warrior elite of Summerland (as it quotes on the back). Although it contains many good fantastical ideas, there is always a feeling of deja-vu connected.

The Dark Door Opens is not much better in theory, but perhaps a more enjoyable read. Now that the Kai have been eliminated, Lone Wolf is born and the plot follows his quest to reach King Ulmar and warn him of impending doom.

Analogic Software has now ceased the effort of Lone Wolf, and thus released a new computer game based on his exploits.

Entitled simply *The Mirror of Death*, the game places Lone Wolf in an arcade-adventure. Although one could be forgiven for thinking that it has more than a smidgen of hack-and-slash action more usual in a beat-'em-up.

The plot follows Lone Wolf as he tries to recover a stolen gem, once belonging to his mentors, the Kai. This is no easy task as the tower in which it is imprisoned is full of fire-spitting gargoyles, demons and countless other fatal dangers. Once all these have been negotiated, Wolf enters face-to-face with the Mirror of Death itself, which is not feared much for nothing. It has many properties, but its most famous, and arguably its most lethal, is the fact that it can produce carbon-copies of its enemies. Not just any old reproduction mind you, but one that has evil intentions. It will take more than a quick slash to vanquish this fellow.

Lone Wolf - The Mirror of Death boasts great graphics and good action. The only reservation that I have is that the play may not be varied enough, thus giving it a short life.

It is available for the PC, Amiga, and C64 at varying prices.

Other news comes out way that *Mirror of Death* will not be the last Lone Wolf excursion into the world of home computers. Analogic has many more adventures planned for

1990.

Last, but certainly not least, in the new Lone Wolf releases is the inter-



active telephone line, written by Joe Dever. This is surely in direct competition with Steve Jackson's *Fist*.

The company involved, *Broadway*, has invented a new label under which they hope to release many games of this kind, *Forces of Doom* being just the first. *Phonquest* (TM) games will all use the same system of operation, although at this stage it is hard to determine whether they'll just be sticking with Lone Wolf, or if they may branch into other areas of interactive 'quests'.

Unlike *Fist*, which uses a tone method to determine actions, *FOLD* breaks new ground with voice-interactive technology. This allows the caller to make 'real-time' decisions during play, and increases the tension created by the stunning sound-effects.

The way in which this works is simple. When at a point in the game where a passive decision is to be made, you will be given a number of options (like the Lone Wolf gamebooks), each separate option carries a different number which must be guessed after a flip, thus choosing the action to be performed. Aggressive moves are similar, although they must be guessed when in battle, and a sticky and is almost a certainty.

The sound-track was recorded at a top London studio using a very able team of professional actors, with many other effects added to make the game an all-round fantasy experience. Separate sections of the adventure are narrated by different voices (male and female) and they all do a great job in the atmosphere stakes.

One right I have of any game like

this is the prices which must be charged. CD peak rate is 23p per minute, whilst peak is 39p per minute, costing a mammoth £13 for a hour's play (all peak), which will add up to £103 per week, if it is played for an hour a day, quite a sum of money.

This is most worrying for the younger children who become addicted to such things (hence the reason for *Charlie's* demise). My advice to anybody worried about the costs, buy one of these gamebooks, at around £2.50, and get a friend around to read it out to you, it's not that much different and certainly a lot cheaper.

I'm not here to condemn the line, I think it's rather good fun, and if played in moderation it's a very worthwhile venture.

All the Lone Wolf products have something going for them, and the fact is that they'll be popular for a long while yet, thanks to *Joe Dever's* persistence, and *Broadway's* money, for when both are flowing legends can be born.

Lone Wolf Bibliography

Lone Wolf Gamebooks

1. *Flight From The Dark* - Joe Dever & Gary Chalk.
2. *Fire on the Water* - Joe Dever & Gary Chalk.
3. *The Caverns of Kulte* - Joe Dever & Gary Chalk.
4. *The Chasm of Doom* - Joe Dever & Gary Chalk.
5. *Shadow on the Sand* - Joe Dever & Gary Chalk.
6. *The Kingdoms of Tantor* - Joe Dever & Gary Chalk.
7. *Castle Death* - Joe Dever & Gary Chalk.
8. *The Jungle of Horrors* - Joe Dever & Gary Chalk.
9. *The Cauldron of Fire* - Joe Dever.
10. *The Dungeons of Torgar* - Joe Dever.
11. *The Prisoners of Time* - Joe Dever.
12. *The Masters of Darkness* - Joe Dever.

The World of Lone Wolf Gamebooks

1. *Greyfax the Wizard* - Joe Dever & Ian Page.
2. *The Forbidden City* - Joe Dever & Ian Page.
3. *Beyond the Nightmare Gate* - Joe Dever & Ian Page.
4. *War of the Wizards* - Joe Dever & Ian Page.

Legends of Lone Wolf

1. *Eclipse of the Kai* - Joe Dever & John Grant.
2. *The Dark Desert Opens* - Joe Dever & John Grant.

Others

- The Magus and Companion - Joe Dever & Gary Chalk.
- The Lone Wolf Poster Painting Book - Gary Chalk.



DATTEL ELECTRONICS

DATA RECORDER



- Quality Commodore compatible data recorder
 - Power control, eraser, etc.
 - Model for the 128.
 - Send now for quick delivery.
- ONLY £24.99**



SPRINT 128

- Full feature Commodore Master Recorder
 - General purpose 48,000 Hz range of full size Commodore Parallel Interface
 - Easy to use - supports Commodore Sprints 128
 - Advanced Microprocessors (MIPS) means no preset programming to load
 - Works with most applications.
 - No noise to log
- ONLY £29.99**

EPROM ERASER

- Keeps up to 4 Eproms at once.
 - Beats other erasers by half the time.
 - Uses little to your pocket!
 - Always complete - no more to buy.
- ONLY £39.99**

ROBOTEK 64



MODEL 6 CONTROL MODULE 64-01

- 4 output channels - each with individual ring
- 4 input channels - each fully buffered 100,000 impedance
- Analogue input with full 4 bit control
- 3 input logic state control
- Reference features: test pointer, analogue output, analogue input, analogue output, digital control etc.

ONLY £39.99
 INCLUDING CARRYING CASE AND BATTERIES. DELIVERY TIME ON ORDER.



MIDI 64

- FULL MIDI INTERFACE FOR THE 6400 AT A BARGAIN PRICE!
 - WORKS WITH ANY OF THE 6400'S
 - Compatible with most leading software packages.
- ONLY £29.99**



DIGITAL SOUND SAMPLER

- The new sampler allows you to record and transmit digitally into memory & then replay it with astounding clarity.
- Replaces Commodore's interface with a fully functioning synthesiser
- Now with full control editing modes to produce astounding effects
- Full 128 Hz to 4.5 KHz reproduction
- 100% compatible with cassette interface (i.e. total cost for 200,000 samples)
- Uses efficient noise reduction and time sharing of resources
- Used to form 16, 32 or 64 channel systems
- Powerful responses with editing facilities
- Load/save sample. Up to 4 samples to memory at one time
- Complete software, hardware package. Free to view (over 1000)

ONLY £49.99

Cometium

- DIGITAL VOICE SYSTEM
- Now you can have your digital voice recorder into a digital voice system.
- 4-Digital voice channels to monitor all 480 times
- Complete with 4 voice files, dual Area module - not applicable
- Cometium supports 4800 digital bits per 4 voice files
- Full editing, Area module, Load save facilities.
- Adaptor to fit in through TV opening

ONLY £9.99
 21000 12000 00 0000

COM-DRUM EDITOR

- 14 drum sounds supplied - plus 4 to enable you to create your own drum kit
 - An accurate sounds supplied with a cross drum to make new kits
 - Load/save facilities.
- ONLY £4.99** OVER 1000

PARALLEL PRINTER CABLE

- Connects all dot printers to the parallel port of your 644128.
 - Many programmes and cartridges (Matrix Printer/Print/Cartridge etc.) will drive printers from this port
- ONLY £12.99** COMPLETE

A TOTAL MIDI MUSIC PACKAGE



SAVE OVER
£70

YAMAHA'S NEW VALUE PACK - THE YAMAHA 8MG 10 FM SYNTHESIZER KEYBOARD IS NOW COMPLETELY BUNDLED WITH THE 6400. THE SYNTHESIZER BUNDLE INCLUDES SOFTWARE & MORE FEATURES PLUS FREE HOME TUNING TO CONNECT EVERYTHING TO YOUR COMPUTER SYSTEM

FREE MIDI CABLES

- YAMAHA 8MG 10 FM SYNTHESIZER KEYBOARD
- Beautifully styled polysynthesizer keyboard with shiver stop. Top quality loudspeakers.
- 100 notes keyboard
- 120 built-in instruments and digital changes
- Free 480 speakers
- Full 128 samples
- Beautifully styled polysynthesizer keyboard with shiver stop.
- Keyboard & 70" features in 480/120 adaptor.

MUSIC SYSTEM

- A full feature music studio with more...
- Editing and expansion facilities
- More features - too many to list
- More. And only

MIDI INTERFACE

- With 128/120 ports.
- Simply plug in!

FOR ONLY
£99.99
 NO MORE TO BUY!



RESET CARTRIDGE

- Unbelievable speed factor. 50000 - over 1000 changes per second allows for ultimate and unobtainable accuracy
 - Complete reset on various "non-volatile" programs.
 - AMM (patented) interface etc.
 - Simply plugs in to cartridge port.
- ONLY £5.99**

3 SLOT MOTHERBOARD



SAVE SPACE & TEAR ON YOUR EXPANSION PORT

- With storage space available on 3 high grade PCB
 - Enables to tear any size
 - Fully buffered.
 - Send location and an instant safety line
- ONLY £16.99**

Toolkit IV

THE ULTIMATE DISK TOOLKIT FOR THE 1280/1281.

- 1** Disk editor - an absolute must for the serious disk hacker. Toolkit IV has some features that stand out from the rest.
 - 2** COPY CONTROL V2 - Used to write any track to sector including sector 0 (formatted tracks). Useful throughout software.
 - 3** DISK FORMATTER V2 - Formats & checks (MS) sector information including all types of transfer packages. The sector transfer is done only. (Removable disks). Also with full sector full gap.
 - 4** DISK LOCK - Add password. Monitor file size. Display file name. (and additional) (Remember) use the program directly from the disk in 6400000 or 64000000 including subdirectory operation. (and full)
 - 5** BEEP FILE COPY - Submits the copy. Works up to 4 times (normal) speed.
 - 6** BEEP BEEP COPY - Copy an entire disk in 2 minutes or less using single 1281.
 - 7** FILE CONNECTION - Can compare (compare) programs by up to 2000. (BEEP) (BEEP) up to 4 times.
 - 8** FORMATTER - An essential feature on every disk up to 4 times (normal) track or full track (disk) 0 to 12.
 - 9** BEEP BEEP COPY - (copy) file & compare all read errors including entire 0 (formatted) level or system & full track (disk) 0 to 12.
- ONLY £9.99**

Contributions



*So you have written a program? You want it published?
Follow these guidelines when you send it in to us!*

Your Commodore is always on the lookout for new programs, hints, tips, articles and even regular series. In fact, if you have something that you think could be of use to other Commodore owners, we want to hear about it.

Below you will find a list of guidelines that will help us to deal with any hints you send in. We don't expect everybody to be building William Shakespeares, although if you do follow these simple rules it will make our job a lot easier. (And simpler).

We are constantly striving to make our magazines more professional in their approach. Therefore, it is to this end that we are making some alterations to how programs should be submitted with a view to publication.

In future, whenever you submit programs or articles for possible publication, we ask you to ensure it is submitted according to the guidelines set out below. Please note:-

Any programs NOT submitted along these guidelines will automatically be rejected.

Submission Guidelines

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double spaced, so there should be a blank line between each line of text. There should be a margin of at least 10 characters around the text.

3) The very first page should consist of the following:

Name of the article/program
Machine that it is for
Any extra needed - drive/printer, etc.
Your name, address and telephone number.

4) The top of every page should have the following information on it:
Program title
Your name
Page number

For example:

Rasters/A. Hogg/ Page 1

5) Do not make additional marks on your text, especially underlining.

6) On the bottom of each page put the word MORE if there are more pages or END if it is the last one.

7) If possible enclose a listing of your program. If written in machine code, then a listing of the source code would be appreciated.

8) Programs should be on either tape or disk with TWO copies included. If problems appear then we have more chance of successful loads. If the program is less than 10 lines it can be included in the text.

9) If your article needs any artwork then supply clear examples of what is needed. We don't expect you to be a first class artist but we do need to see what is required.

An Introduction To Plus/4 Machine Language

Mark Everingham reveals the inner workings of his Plus/4

In future instalments of *FourGround*, we shall be printing many articles and programs which need a basic understanding of Machine Language programming. For this reason, over the next few months *FourGround* presents a brief beginners course to machine language. It is obviously not possible to present a full discussion of the subject within the limited space of the magazine, but the course aims to provide a foundation on which to build more advanced knowledge.

For most people, the phrase "machine-code" has horrible images attached to it - strings of binary, incomprehensible listings and degree level mathematics. However, the fears the majority have about machine-code are totally unfounded - the language can be easy and extremely rewarding to use. If you have tried to learn machine-code before and given up in frustration, don't run off now, most books on the subject tend to be extremely badly written, full of jargon and irrelevant information. This short course has been written to be compact and easy to understand while retaining all the essential information. For those who can already program in machine language, it should serve as a useful reference work.

What Is Machine-Code?

"Machine-code", "Machine Language" or "Assembly Language" is a computer language just like any other - Basic, PASCAL or C. Machine Language is the native language of the Plus/4 and is that the computer at its most basic level cannot understand anything else. Basic may seem like the computers natural language, but is really nothing more than a program written in machine language. At the heart of your Plus/4 is a silicon chip called the 7300 Central Processing Unit (CPU). When you write programs in machine language, you are sending instructions directly to this chip. When you write in Basic, your Basic programs have to be converted into machine-code as they are run, because the 7300 cannot understand them in their Basic form.

So why program in machine language? The reason most people use Basic is because it is so easy to learn. For example, to draw a circle in Basic takes just one command. In machine language, hundreds of individual commands are needed.

This might make Basic seem far more attractive, but the problem with Basic is that it runs incredibly slowly. Machine language runs commands at up to a million instructions per second, whereas with Basic you're lucky if you achieve 60 per second. Another advantage of machine language is that because you're working at such close proximity to the actual hardware of the machine, you have the ultimate in power over every individual function that the Plus/4 hardware can perform. Once you have begun programming in machine language, Basic will, I can assure you, seem horribly inefficient.

Number Bases - Decimal, Binary, and the Dreaded Hex!

Computers are number machines - Your Plus/4 computer knows nothing about anything but numbers. Everything it does concerns numbers, even the commands it executes are stored as numbers. We human-beings tend to use the number base called decimal or tenary, simply because we happen to have ten fingers on our hands, but computers don't like decimal, so instead they use "Hexadecimal" or "Binary".

The concept of number bases is really simple. Think of a decimal number, say 12345. The key to understanding the magnitude of the number is column headings. Look at the number 12345 written under column headings below.

Heading:	10000	1000	100	10	1
Value:	1	2	3	4	5

In order to work out what number is written down, we just add up the products of the values and headings: $1 \times 10000 + 2 \times 1000 + 3 \times 100 + 4 \times 10 + 5$. Each column heading is a power of 10, so the first one is 1×10^4 , the second one 2×10^3 , the third 3×10^2 and so on. The principle for hex and binary is identical, except that for hex, the base is 16 and for binary it is 2. This means that the column headings are 1, 16, 256, 4096 etc. for hex, and 1, 2, 4, 8, 16, 32, 64 etc. for binary. To convert a number to decimal, just repeat the process above using the new column headings. For binary, this is easy, as each column can only hold a number 0 or 1. The number 170 is shown here:

Heading:	128	64	32	16	8	4	2	1
Value:	1	0	1	0	1	0	1	0

The final value is calculated as $1x128 + 1x32 + 1x8 + 1x2$. The same method is applied for hexadecimal, except that if the base is 16, each column needs to hold a number between 0 and 15, whereas in decimal only between 0 and 9. This means we need 8 more digits. For these we use the first six letters of the alphabet, A=10, B=11, C=12, D=13. The number 4660 is shown below:

Heading	4096	256	16	1
Value:	1	3	3	4

Of course, if in a program we need some numbers which were binary, some decimal and some hex, we couldn't tell the difference - the number 100 could be 100 decimal, 5 binary or 257 hex. The answer is that binary numbers are preceded by a "B" (percent) sign, hex by a "H" (String/Decimal) sign, and decimal numbers have no prefix. This avoids all possibility of confusion. For most of the time, however, you will find you use hexadecimal, so the problem need not arise.

So why do we use hex and binary instead of decimal? Computers are electronic devices, made up of banks of transistor switches. A switch can only be On or Off, and these settings can be effectively represented by 1 and 0. These 1's or 0's are called bits - Binary Digits referred to by their column headings 1-7 (right-left), and collections of eight of them (2³ decimal) are called Bytes. Hexadecimal is used because it is much shorter to write down, while retaining a similarity to binary. The number 255 in decimal needs three digits, in binary, 8, but in hex it needs only two.

The Importance Of Numbers

So far we have only talked about numbers for numbers sake, and this is the original error which most machine-code users make. In machine language, nothing but numbers are used, but the important thing to remember is that a number can mean anything. In real life, numbers can be house numbers, telephone numbers, prices, times, or a host of other things. Machine language is identical, numbers can be colours, sounds, commands, speeds or anything you wish. All of the commands discussed in the section "The Instruction Set" perform operations on numbers. If you can remember that these numbers can be of any significance, machine-code programming will be elementary!

The Memory Machine

Your Plus/4 is described as a 64K computer. This means that it has enough memory chips to store 65535 byte long binary numbers. You might expect 64x1000, but 65535 is equivalent to 64x1024, 1024 being near enough to 1000 which is a nice binary number. Because of the binary system, each of these 65535 "locations" can hold a number in the range 0-255, and each location is referred to as "address". This is simply a number between 0 and 65535 which points to a byte of memory. In addition, your Plus/4 has two types of memory: ROM (Read Only Memory) and RAM (Random Access Memory). The ROM holds the machine-code program from BASIC. It lives at address 30000-3FFFF, and cannot be changed. The address space 30000-31FFF is the RAM, although the extra 32k is found at 30000-3FFFF, found in the ROM. For the moment we will only consider the lower 32k.

Each byte of RAM can be put to a different use. To find out which byte of RAM is used for what, you should

consult a manual such as ANCO's "CIB-PLUS/4 REFERENCE BOOK" which has a full list. Many locations are used for things like the screen, Basic calculations or Basic programs. When you write a program in Basic, the Basic program in ROM finds a place in RAM to store the program and puts it there. In machine language, you have to tell the Plus/4 where to put the program, and you must take care not to put it in a place which is used by another part of the computer!

The Tedmon Machine-Code Monitor

On almost all computers, writing a program in machine-code means that you have to go out and buy a piece of software called an Assembler, lucky for you, the Plus/4 has a simple one built in, called TEDMON.

Machine-code programs are written like Basic as a series of text commands, such as "TXA", "LDA \$8000,X" and "RRR". Each command is represented by a three-letter mnemonic, or abbreviation. To actually run these commands, they must be converted into a series of numbers and stored in RAM. This process is called Assembling.

The TEDMON program on ROM in your Plus/4 will carry out the simple assembly process for you to save you doing all the work. To assemble a program, type "ASSEMBLE" from Basic. In order to assemble a program, you have to tell Basic where to put the program. Consult a memory map if you have one to find a suitable place, but for most of the time, a good place to put programs is at \$800. This location in memory leaves you with 8k free for Basic users with a graphics screen in operation, leaving 8k for your machine-code program - an amount you are not likely to use up for a long, long time! The TEDMON assemble command takes the below form:

A \$aaa xxx opcmd

The "A" is the actual assemble command, and can be replaced by a "." (period) if you prefer. "\$aaa" is the address at which you wish the line of machine-code to be stored. It should always be expressed in hex and preceded by a dollar symbol. "xxx" is the three-character mnemonic of the command you wish to be assembled, and "opcmd" is the operand or parameter the command takes. When you enter a line and push (Return), the line is stored in memory, and a new address is printed, allowing you to enter the next line. All numbers entered into TEDMON should be hexadecimal, and preceded by a dollar symbol.

When you have finished entering a program, delete the address on the screen. You can exit to Basic by pushing "X" and (Return) for exit, or you can run your program by typing "O" plus the first start address of your program. The dollar symbol should not be entered.

Once your program is in memory, you can look at it or edit it by typing "D" plus the address at which you want to start looking. A series of lines will be displayed on the screen. This is called Disassemble, and the lines on the screen can then be edited and re-entered using the normal Screen Editor functions.

We have not covered here every function of TEDMON, but only the most important ones. Full coverage of the TEDMON monitor can be found in your Plus/4 manual.

7501 Processor Registers

As I have said, all of machine-code's commands operate

on numbers. These numbers can be held in three places - in a memory location, in the memory directly after the machine-code command, or in one of the 7501 registers, which are like fast, internal bytes of memory.

The Program Counter

The program counter (PC) is the only 7501 16-bit register. In it is stored the address where the next command to be executed can be found. When an instruction is executed, the PC is automatically incremented so that it points to the address of the next instruction. When you divert control to another location using a JMP, JSR or branch instruction (see "The 7501 Instruction Set"), the address in the PC is changed accordingly, so you don't have to worry about it.

The Status Register

The status register (SR) is an 8-bit (byte) register which is used to indicate the status of various options of the 7501 CPU. In fact, only seven of the eight bits are actually used, and the functions of these bits are discussed below. Each bit is known as a "flag".

Bit 7 - The N (Negative) Flag. The negative or N bit is set to 1 when the last numeric result had bit 7 set, and is set to 0 when the number had bit 7 cleared. Effectively, numbers over 127 set the N bit, and those below clear it. If bit 7 is used to show the sign of the number, the N bit reflects this sign accordingly.

Bit 6 - The V (Overflow) Flag. The Overflow (V) bit is set to 1 when the last operation resulted in what is known as a two's complement overflow. This is only of use in signed arithmetic, where bit 7 represents the sign (1 =, 0 =). Overflow is set when the status of bit seven has changed incorrectly, in such cases as two numbers yielding a result greater than will fit in one signed byte.

Bit 4 - The B (Break) Flag. The Break bit is set to 1 when the last operation was a "BRK", or is cleared if the operation was anything else.

Bit 3 - The D (Decimal) Flag. Setting the decimal bit enters the decimal mode of the 7501 processor. Clearing it reverts to binary arithmetic. For information on the decimal mode, see the SED instruction.

Bit 2 - The I (Interrupt) Flag. Setting the interrupt (I) bit in one causes interrupts to be disabled, as clearing enables interrupts. The function only works with IRQ (Interrupt Request) interrupts, and is covered fully in the section on interrupts.

Bit 1 - The Z (Zero) Flag. The zero bit is set to 1 when the result of the last operation was zero, for instance when subtracting a number from itself. If the result was any number but zero, the zero bit is cleared.

Bit 0 - The C (Carry) Flag. The carry bit is set to 1 when the last addition gave a result which could not be expressed in 8 bits. When subtracting, it is cleared if a borrow out

of eight bits was required. The bit is also used by the shift and rotate instructions, where the last bit is shifted into the carry flag.

The Accumulator (A)

The accumulator, or A register is the most important of the 7501 registers. It is an 8-bit register which is used for all arithmetic and bit manipulation operations, as well as being used for most data transfer functions.

The X Index Register (X)

The X register is one of a pair of "index" registers. It is like the accumulator, in that it is an 8-bit register, but only the most simple operations can be carried out on X. Its primary use is as an index for indexed addressing modes - See the section on addressing modes.

The Y Index Register (Y)

The Y register is identical to the X register, also being used mainly as an index.

The Stack Pointer (SP)

The stack pointer is another 8-bit register, which contains a pointer to the current byte on the 7501 machine stack. See the section which follows on the stack for an outline of its use.

The 7501 Processor Stack

In Basic, you have two instructions to alter the flow of a program - GOTO and GOSUB is simple - it just performs an absolute, unconditional jump to a line. GOSUB does the same thing except that before the jump is performed, the statement to which the program should return is first pushed into an area of memory called the Basic stack. Then, when a RETURN statement is encountered, the program pulls the return statement back off the Basic stack and jumps to it, machine language works in the same way using its own machine stack, except that the GOSUB command is replaced by JSR (Jump to Sub-Routine). When JSR is executed, the address of the next command is pushed onto the stack so that when a subsequent RTS (Return from Subroutine) is found, the address can be pulled back off the stack and transferred to the PC to continue program execution. The process of pushing and pulling is automatic and controlled by the Stack Pointer. The machine stack is in fact a 256-byte (1 page) area of memory located at \$0100 onwards.

In addition to being used for sub-routines, the stack can also be used as a temporary store for numbers, avoiding the need to use a specific area of memory. The PRA, PLA, PHP, PLP instructions push numbers onto the stack and pull them back off it again. When a number is pushed or pulled, the stack pointer is decremented or incremented accordingly. You can however control the stack pointer contents directly using the TMS and TSN instructions. The stack pointer is a single 8-bit index into the stack area at \$0100.

Note that when using the stack, you must of course match every JSR with an equivalent RTS, and push with equivalent pull. The stack can accommodate 256 bytes, so upto 128 levels of nested sub-routines are theoretically possible. It is certainly unlikely that you will ever need more than this.

We've had to cut Mark short here, but you'll find more in the second installment.

A Serious Case of Bondage



Crosby — Kevin Crosby takes a look at the latest Bond film and its accompanying game from Domark. After 27 years the world's best known secret agent — James Bond returns once again and, as the film posters state,

"This time he's out for revenge".

Licence to Kill is the 16th official Bond movie (not counting Casino Royale and Never Say Never Again) and the second for new boy Timothy Dalton (not the last as some television networks would have us believe!) Dalton himself is very pleased with "Licence", "it's much closer to the original character in the books" he enthused at the recent premier in London's Leicester Square Theatre. "It's tough, tense, exciting, everything a Bond film should be". It's certainly a lot tougher than previous films which has been the main reason for Licence to Kill's 15 certificate (the first time a Bond film has received anything other than a 'PG' or equivalent). This was only obtained after over a month of the more violent moments were cut.

However, this doesn't detract from the film's enjoyment, which is the main thing really. It's just a shame that our film censors in this country aren't a little more like their counterparts on the other side of the Atlantic who have given it a PG13, this means that anyone can see it with a parent but you have to be 13 or over to see it on your own. Seems reasonable enough to me!

The film itself centres around the twilight world of Drago Baron Frank Sanchez — an unsavory character who escapes imprisonment and gets his own back on those that caught him (Bond and Felix Leiter). Now Bond becomes a vigilante to get Sanchez back for what he's done, but he must do it without the backing of her Majesty's Secret Service. This time he's on his own.

All the familiar elements are there; Evil haddas, Beautiful Girls, spectacular stunt sequences and of course those ingenious gadgets courtesy of 'Q', played for the 14th time (he won't be Dr No or Live and Let Die) by Desmond Llewellyn. However all these elements have a much harder edge. One could almost be mistaken for thinking this was a Dirty Harry



film, particularly if one is used to the Roger Moore style of Bond. If that's what you've been brought up on then prepare yourself for a shock... for this is a Bond for the 90's. The action is that bit more violent, the villains are that much more realistic (which makes them all the more sinister) and the Bond girls aren't just the stereotypical beauties of the past, they now have quite an effect over the action. Is it welcome? Most certainly. As Dalton puts it "This is an Action, Adventure Thriller for adults that kids can also enjoy". A well written plot from the pen of Michael G. Wilson and Richard Maibaum, competently directed by John Glen, the fifth Bond

Directorship, makes for an excellent entertainment.

It is nice to see Demark has pulled out all the stops to release the game to coincide with the general release of the film. This will be Demark's fourth game based on a Bond film. The first was View to a Kill back in '85, followed by The Living Daylights and Live and Let Die. It'll be launched across all formats, including the Commodore 64, Amiga and PC formats, simultaneously. All three versions of interest to Commodore readers look and play really well, particularly the Amiga version.

The game comprises six parts, split up among three scenes from the film.



The first involves James and Felix flying a helicopter at low level over Crazy City. Avoiding tall buildings and gun emplacements - a little like Thunder Blade. Once through that, you (as Bond) get off on foot after Sanchez but you have to get through plenty of his henchmen first. Weaving your way through a dangerous maze of exploding oil barrels and bullets, you must get back to Felix in one piece. Then it's back in the air to capture Sanchez's plane. Here you are lowered onto the back of the plane where you must attach a towrope and bring Sanchez in.

In Scene Two 007 must foil a narcotics drop, but first he must survive in the water armed with nothing but a knife against divers and armoured boats. Bombs are added for destroying the Drugs caches as well as escaping with your life. Next, you attack yourself, via Harpoon's a scaphale's poisons for a high speed, barrelroll waterski chase, avoiding catamarans whilst you work your way to the scaphale and gain control.

In the final scenes you are in hot pursuit of Sanchez having just obli-



erated his processing plant. He's taken to the winding mountain paths as part of a conveyer of 18 wheel juggernauts. The end is near, the focus of your vengeance is just around that bend. You just have to destroy each of the tankers one at a time until you get to the one at the front of the conveyer with you know who in it, but because he's packing some serious heat in the form of Stinger missiles.

The whole game takes the form of a seriously scrolling, birds-eye view shoot 'em up and works very well as a game. Which as a movie tie-in is rare indeed.

Bond Filmography

Dr. No - (1962) Sean Connery - Dir Terence Young.
From Russia With Love - (1963) Sean Connery - Dir Terence Young.
Goldfinger - (1964) Sean Connery - Dir Guy Hamilton.
Thunderball - (1965) Sean Connery - Dir Terence Young.
You Only Live Twice - (1967) Sean Connery - Dir Lewis Gilbert.
On Her Majesty's Secret Service - (1968) George Lazenby - Dir Peter Hunt.
Diamonds Are Forever - (1971) Sean Connery - Dir Guy Hamilton.
Live and Let Die - (1973) Roger Moore - Dir Guy Hamilton.

The Man With The Golden Gun - (1975) Roger Moore - Dir Guy Hamilton.

The Spy Who Loved Me - (1977) Roger Moore - Dir Lewis Gilbert.
Moonraker - (1979) Roger Moore - Dir Lewis Gilbert.

For Your Eyes Only - (1981) Roger Moore - Dir John Glen.

Octopussy - (1983) Roger Moore - Dir John Glen.

View To A Kill - (1985) Roger Moore - Dir John Glen.

The Living Daylights - (1987) Timothy Dalton - Dir John Glen.

Licence To Kill - (1989) Timothy Dalton - Dir John Glen.



007

LICENCE TO KILL

128 Corner



Pin	Type	Note
1	VSYNC SYNC	Unidirectional SYNC output
2	GRND	
3	AUDIO OUT	
4	VIDEO OUT	
5	AUDIO IN	Composite signal output
6	COLOR OUT	
7	NC	On-chip signal output No connection
8	NC	No connection



Pin	Signal
1	Ground
2	Ground
3	Red
4	Green
5	Blue
6	Intensity
7	Monochrome
8	Monochrome Sync
9	Monochrome Sync

Dear 128 Corner:

You, I own a Commodore 128 and have done since I upgraded from the good old 64 back in 1986. I was pleased to see that you have decided to provide a feature in your magazine for this excellent machine.

I agree with your views regarding the poor availability of software. I suppose the reason for this is that the machine is too small to be a fully

dodged business machine and too large to be a game machine. However, by viewing the magazines it is possible to get hold of some magnificent software that is specific to the computer.

I was surprised that you missed out the C128 version of *GEORRITZ WORKSHOP* as an example of a reasonable word-processor. I was initially put off by the *GEOS* system

because it did not allow printing in NLQ mode. The fonts that were provided were too 'dotty' as my Star 5610-C for serious letters. This has been solved in *GEORRITZ WORKSHOP*.

The biggest advantage of the program is that you can take large sections of figures from *GEOSCALC 128* and paste them into letters without having to write them out again.

For straightforward word-processing I would think that *Superscript 128* is best. Its sister program *Superbase 128* is another program worth seeking out if you want to create a database for any reason. The Basic like programming language, included in the program, allows you to manipulate the data in your own applications programs and adds a new dimension to using a database.

The type surrounding the design makes everyone wonder if they shouldn't part with their old computer and buy a new one. I like playing C64 games on the 128 and am happy with spreadsheet, database and word-processor applications that I have for the machine. In the words of the Americans, if it isn't bust, don't fix it.

Colin Murrell, Bolton, Lancs

Have you tried Post Master 128 or Figurefile II, both are very excellent applications. I totally agree with your criticisms about the 128. I too would like my own personal design, but why buy a Bentley to learn to drive when a Morris Minor will do the trick???

Dear 128 Corner,

It's good to see that you intend to continue covering the C128 computer. I've had one for a few years now and use it quite regularly for word-processing and filing. I must say that I prefer using the C128 over the 'real' computers that we have in the office. It's a

lot easier to use and more friendly (I can also play games as it when I'm bored).

When I first purchased my C128 I was very disappointed to find that I couldn't use the 80-column video mode without purchasing an expensive colour monitor, which I couldn't afford. I was intrigued however when I saw people advertising switches that allowed you to use 80 column mode on a cheap monochrome monitor - not as good as colour but certainly better than using 40-columns all of the time. I investigated this further and careful examination of the 80 column video socket in the manual revealed that there is an 80 column composite video output on pin 7 of the connector. Converting a suitable plug to this pin and the earth (pins 1 and 2) means that the computer can be plugged into a suitable composite monitor.

If you want to use the same monitor to display 40 column video then this can also be achieved. Pin 4 on the 40-column video connector is

the video out and pin 2 is ground. Connect the video and ground to a suitable connector as before and you can use the monitor to display 40 columns as well.

I know of a number of other people who use a composite monitor with a lead as described above, in some cases they use pin 1 (LUM/SYNC) instead of pin 4 though on my monitor the results aren't as good.

I have taken my lead even further and placed a double throw switch in its lead. One pole of the switch goes to the 80-column video pin, the other pole the 40-column video pin. The central pin of the switch is wired to the video plug for my monitor and all of the earth pins are connected together. Switching between 80 and 80 columns is now a simple matter of pressing the 40/80 switch on my monitor and flicking my own switch.

One further advantage of using a video lead such as the one described above is that you can, contrary to popular belief, get 80-columns on your

television, as long as you have a video recorder anyway. Most video recorders have an input for a video camera. This input is a composite video one. Plug the lead into this socket, make sure that your TV and video are set up to display from the camera input and you can see your C128 in 80-column mode on your TV. The quality may not be as good as a monitor, but at least it's 80-columns! N. Garton, Rothham, N. Yorks.

Get In Touch

C128 Corner is a forum for all 128 users. If you have any comments, suggestions or questions do send them in. Without your contribution then 128 Corner will not be able to continue, so come on, write to:

128 Corner

Your Commodore

Argus House

Boundary Way

Hemel Hempstead

Herts.

HP1 1ST

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema.

Most of them comply with the rules contained in the British Code of Advertising Practice.

But some of them break the rules and warrant your complaints.

If you're not sure about which ones they are, however, drop us a line and we'll send you an abridged copy of the Advertising Code.

Then, if an advertisement bothers you, you'll be justified in bothering us.

The Advertising Standards Authority. If an advertisement is wrong, we've got to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

This space is donated to the interests of high standards of advertising.

FORM MASTER

Don't just follow an 8000 Form Master - see an example of our system.

"I have purchased hundreds of systems over the last forty years and Form Master is by far the most brilliant system. James Christman is an amazing pro at what he does. My profit from date of purchase is £1500." **Mr. Manchester, Liverpool 15, 11/89**

"Form Master does what I claimed for it and finds good priced adverts in companies having success." **P. James, Dorset 6, 11/88**

"This first time I used Form Master I had a 50% increase and I have never looked back. I am currently making a profit of £300. After twelve months I am making 80% increase and have other ripping companies in the field. I won't be without it." **P. Black, Shropshire 12, 11/88**

"I had a number of adverts within a week of using the system and in the last seven days I have not placed a single ad. You can imagine how delighted I am." **R. Newell, Isle of Man 12, 11/88**

Form Master has averaged a 10% success rate over the National Form Master 10 Day, Contained System. Don't be in the losing end for the last season - order Form Master Now. **SPECIAL OFFER** for anyone who registers training and 80 Advertiser packages. Only £28.00 inc VAT & P&P.

* Available for Commodore 800, 801, 4+, PC, C/16, C64 & C128.

ORDER FORM MASTER about from Data Systems using any major credit card or by cheque.
To order by Cash Card simply call our 24 hr. ORDERLINE on 0800 000000 or 026 26 including 10% Post and Packing.

Enjoy free

FORM BY CARD

Card No.

Signal

Name

Address

SPECIAL OFFER not valid

Data Systems Ltd, 10 Victoria Street, Hemel Hempstead, Herts. Postcode HP1 1SD. Registered No. 322294.

Waiting for

Dave Hughes and James Sewell explore the world of the masked avenger, who is considered by some as one of the five most recognised figures in the world.



The Batman, the Dark Knight, the Caped Crusader – call him what you will, his fictional persona has played a major role in our subculture since his conception in a 1939 issue of *Detective Comics*. Fifty years on the masked vigilante, cloaked and hidden in the dark shadows of a fake New York metropolis known as Gotham City, is about to embark on his second big screen adventure. And this time it's for real.

It is ten years since Warner Brothers, movie-making parent company of Batman's publisher DC Comics, first considered the idea of a new Batman feature film. Since that time, comics like *Watchmen* and *The Dark Knight Returns* have brought a new era to the poplite story genre, adding political, racial, sexual and social issues to a field previously associated with muscle-bound, costume-wild freaks of nature. And *Batman: The Movie* is suddenly big business. Unlike the Caped Crusader's previous screen image, Sam Hammett's original 1938 screenplay concentrated on keeping Batman true to the comics – like *How Keanu's* original character, he is a dark brooding vigilante forever seeking retribution for his dead parents, rather than the over-the-top Adam West/Burt Ward team of the camp mid-Sixties spoof series. This time, *POW!* and *ZAP!* are left redundant by a movie that pulls no punches.

For this movie, Batman trades in his blue-and-grey hooded for a matt black rubber outfit resistant more of Frank Miller's *Batman* than Adam West's. This time, no-one will be left smiling at the pranks of his arch-enemy the Joker. And we all know that Batman's boyish sidekick Robin isn't even in the picture – due mainly to the expression of the part of the Joker which followed Jack Nicholson's reluctant acceptance of the role ("It's like God created him for the role" producer Mike Ustin is quoted as saying). Beetlejuice director Tim

or a Dark Knight

Batman directs, and Beetlejuice himself, actor Michael Keaton, fills the Batman's neighborhood shoes. Aside from Jack Nicholson, Jerry Hall and Kim Basinger (both appear, but Robin the Boy Wonder is not set to appear until *BATMAN 2* (already being scripted), about which movie rumors are already flying. Including the possible casting of Robin Williams as rhyming rival the Riddler... Originally, however, Sylvester Stallone and Arnold Schwarzenegger were considered for the title role, both being rejected because they did not fit the image of Batman's millionaire alter-ego Bruce Wayne. Also, as Tim Burton says, "the idea of Arnold Schwarzenegger in a Batman costume is frankly ridiculous." Instead, Michael Keaton wears built-up shoes, a latex chin and fake hair inside his padded suit, further enhancing his transformation from playboy Wayne to Dark Knight.

The plot, like the rest lot, has also undergone major surgery. Sam Hammet's first draft featured The Flying Graysons and their young offspring (Robin) who, like the young Bruce Wayne, witness his parents' murder, this time at the white-gloved hands of the Joker. The broadly Series image of the Batmobile has been replaced by a Gotham motorcycle which wields a machine gun and is powered by a jet turbine—a fast and powerful reminder of the Batman's night. Gotham City, depicted in the new film as its biennial celebrations, is a grimier, darker vision of New York: Commissioner Gordon is fighting police corruption as well as the Joker-dominated underworld collaboration, and the street is filled with the dark and sinister images of washed cars, sex vids and drug pushers, providing the will-to-do areas and making for a violent and dark base for Batman's comeback.

The Joker is depicted as a victimizing madman, driven insane by his own deliriousness following a chemical accident, and seeking his vengeance on humanity. Batman is the only thing that stands between him and his evil destiny, and the film follows the interaction between two iconic myths, each infatuated with his role and each closer to the other than either

would care to admit—a platoon linked closely with British writer Alan Moore's recent novella *The Killing Joke*. Closer still to the movie's plot is Frank Miller's highly-acclaimed *The Dark Knight Returns*, but a crucial scene was (in true Hollywood style) cut from the movie following news that Warner had agreed to a sequel in the classic confrontation of Frank Miller's story. Batman only just stays short of throttling the Joker to death, at which the Joker, in his last and best punch line, breaks his own neck so that Batman will take the rap for his death; this scene was written out following Jack Nicholson's agreement to appear in a sequel... The high Gotham City set—the largest film set since Cleopatra—stands empty at Pinewood Studios in Buckinghamshire, waiting for the crew to return in February for more adventures of the caped crusader. The movie's success is already secured.

At the time of writing, *BATMAN* is two weeks away from its U.S.

release, but already scores of people have gotten their fix. Fans have been reported as seeing *THE DEAD POOL* just to catch the movie's second trailer for *BATMAN* shown along with it. Despite Jack Nicholson's opinion that the movie is "stagnant", the media coverage and the secrecy in which the movie has been shrouded have made a sure-fire winner for Warner. The *Batman* comic series The Curb has rocketed to 3,500% of its cover price virtually overnight. Illegal prints, scripts and unlicensed shoddy print video copies are changing hands at massive prices. The merchandising from the series is likely to cover the cost of the \$30 million movie in a matter of months, with Warner mounting the biggest merchandising campaign since *STAR WARS*, with novels, action figures, masks, model kits and even credit cards aping the *Batmania* of the Sixties.

So what is it about this masked hero that demands such slavish devo-





tion from his legends of lore? Perhaps it is that, unlike irradiated or miracle-endowed superheroes from other comics, Batman is one of us — an ordinary man with no super-powers, who spends two years of his life building himself up to fight crime and avenge the violent deaths of his

parents, ensuring to stamp out injustice to the best of his ability. The simplicity of the character's origin make him easy to identify with, and more believable in a fantasy universe where an orphan from space wears his underpants outside his tights, and where the mere bite of a spider

transforms a man into a superhuman... He is fallible, a haunted, emotional lover constantly at odds with his hidden identity, and the film follows his inner torment and endless harassment for his lost childhood.

After all, at the end, it's about time Batman came of age...





The Music

"I've seen the future and it will be..." is the first lyric of the first track on the second track album. And what a fitting and apt lyric to kick off the Batman bandwagon.

Although there are nine individual songs on the album, only four were to be written, and this record only encompasses half of the music in the film. Danny Elfman started his score in *Batman* and *The World Is Always* provided the "bad" music throughout. Prince doesn't seem to mind though, his presence was requested by Jack Nicholson (the Joker), and being signed to the Warner record label must be was the obvious choice anyway. It seems as though Prince has made the his album for 1992, putting more effort into each song than you would expect from a normal soundtrack composer.

The songs are well constructed, the album is cunningly arranged, with a mix of several different tones, and the flavor of each track is unmistakably Prince... With one exception. *Barbaree* is probably the weakest track, not being altogether funky nor working with Sci-Fi as

many Prince songs before it - just settling for plain old dance music. But, as Prince states; I have seen the future and it will be... more *Pebbles Park* soundtracks.

The Film

Suspicious Batfans may have dismissed the movie before its release, and who wouldn't with the amount of false rumours spread about it in the tabloids. But now, upon its release in America, there seems to be little alternative to congratulating the entire team involved.

Michael Keaton is superb as both Bruce Wayne and the Dark Knight Detective, and would fail to explain Jack Nicholson's performance as the crazy, but always dangerous, Joker. Kevin Costner (Victor Vale) offers one of the best roles.

On its American premiere, 10,000 dedicated Batfans appeared to see the stars parade before them. The lucky few who were able to purchase tickets could not shoulder their *Silverman* *Snakes* (among other stars), and cry and cheer their way through a truly stunning movie.

Once the film has broken all box office

records, as it promises to do, and hit music goes, young and old, staggering out of the cinema exhausted, we will have to contend with the fact that it will not be too long before *Batman 2, 3, 4* and many more will come our way to keep our addiction satisfied. Long live Bruce Wayne, long live the Joker, long live society, long live the Batman...

The Game

Games are constantly in the habit of being their fate, restricting their output, pooling their resources, and pulling a major license outside out of the bag. There can be no better licenses to deliver than *Batman* - the movie, and we are to be sure that it is a winner for the big toy store bank.

The game is split into four sections, all based upon scenes in the film. The first is based in a Gotham City chemical factory (where the Joker is created), which is a role-playing game, and is viewed from a side perspective. The feature in this incarnation of course is Batman's rope which he can use to swing between levels - something akin to *Street Commando*.

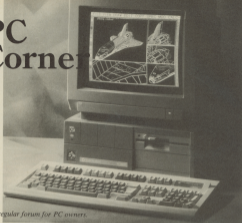
The second subgame revolves around the Batmobile, which Batman must drive in order to fend the Joker's van. Not only must you avoid other traffic, but you can use your batwing to burn corners at a high velocity.

Section three also features one of Bruce Wayne's modes of transport, this time the Batwing, supplied for. The Joker plans to wipe out Gotham using his idea of dead by night. As Batman, you must fly along and cut the wires keeping the balloons aloft to be ground, thus eliminating the danger.

The fourth section is very similar to the first, except the Dark Knight detective has to visit a church, for the final confrontation with the Joker.

Given prices that the game will be released at the same time as the film in this country - August 19th - so it should prove to be another step in the merchandising boom this autumn and almost certainly the biggest license in the software industry, ever.

PC Corner



Our regular forum for PC owners.

I have come to rely on my computer a great deal in the course of my day to day life. I have three machines, all of which are made by different manufacturers, all are "clones" of the IBM PC design and none of them were built by IBM. I use three computers to write software, to do graphics and text for manuals and adverts, control my recording studio, keep up to date with various things and people across three continents and keep my financial records in order for the pleasure of INSTAT and tax man. I even sometimes play games on one of them. Before the time in white coats come to take me away I'd like to explain why.

The great strength of the IBM PC family of computers is their flexibility, this is in part due to the availability of a great number of different options and expansion cards and the sheer volume of software available. The immense number of manufacturers in the PC market means that there is a competition in the price and power of the hardware. The wide range of expansion cards available means that you can use a PC to perform almost

any task by simply adding a relatively simple (and therefore always) printed circuit board to the computer. As with the hardware, having lots of software for the machine gives you the advantages that competition within any field will give you - quality and diversity of products.

Another important aspect of the strength of the IBM PC market is that it will have a long life, even IBM with the release of their PS/2 range of computers haven't been able to put a significant dent in the availability and innovation within the PC field. It's also to know that when you buy a PC that next month or next year you won't be left out on a limb with a computer that has no support or new software.

The speed at which prices are coming down is also amazing, for instance you can now buy a powerful 80386 based AT style PC from New Computing with 1 megabyte of RAM and a 40 megabyte disk for around £1400 (plus VAT) - which is what I paid for my 286/386 (XT) PC two years ago. There is also a pocket sized PC being released by Atari soon, which

uses memory cards instead of disks and has a serial port for RS232 communications and a lap-top PC from Yamaha designed for professional music applications.

Words for the Whys

The IBM PC is basically a "vanilla" computer, it doesn't have the design compromises that are a feature of a lot of home computers. Home computers have been tried and tested and good with a particular goal in mind, this means that they tend to be better on graphics, or extremely low cost but have some special features.

The upshot of this is that the basic PC doesn't perform any particular task exceedingly well, but can perform most tasks competently. The success of the PC design is really quite obvious. Just look at the large number of relatively cheap expansion cards in the market, while a lot these are aimed at the business or industrial market there are a few which are aimed specifically at the "hobby" market as 'home' use or can be used for amusement.

Possibly one reason for this success is the mechanical integrity of the PC system. While there is nothing particularly novel about the idea of having additional cards for a personal/home computer, the allowing of space inside the case for them means that the expanded system is robust enough to stand hostile environments (for example those involving pets and small children). The inclusion of the expansion bus in the PC's design also gives us an insight into the microcosm of the PC which in turn explains just why there is so much essential and public domain software available for the PC.

The History Bits

The disk operating system for DOS® used by the PC is based on one of the earliest systems designed for 8 bit microcomputer systems. This DOS was called CP/M, which stands for Control Program for Microcomputers, the major feature of this software was that it wasn't tied to any one computer. This meant that manufacturers - and even hobbyists - could build a computer for which there was a large amount of common software available. A number of popular PC packages were originally written for CP/M such as Wordstar, dBase II and Super Calc.

The main problem with CP/M was that it was plagued by the fact that there was no standard disk format, the main UK CP/M user group provides disks from its software library in over 120 different disk formats!

The Public Domain

The hobbyist involvement meant that there were a lot of dedicated technical people writing programs for their own use and/or amusement; these people got together in computer clubs in the USA and other parts of the world and released their software in the 'Public Domain' (or PD). These programmers were quite happy to let other people use their software and would often provide limited support, as long as the software wasn't sold commercially. Incidentally, the CP/M operating system isn't dated yet, the Amstrad PCV series uses a version of CP/M version 2.2.

The original IBM PC's operating system was very closely related to an early version of CP/M, which had two rivals, the first was that people who wanted to upgrade their computers to a 16 bit processor could move to a

fairly familiar operating and software environment. The second rival was that existing commercial packages could be easily converted (or ported) to the new machine, and since Microsoft, the suppliers of the PC's operating system, also released the DOS software as a separate product, other manufacturers could also build it in to their computers.

The User Groups that distributed the CP/M public domain software now also have MS-DOS programs in their catalogues which range from word-processors, games and CAD systems to the entire King James version of the Bible (6 disks). The sort of software available from the user groups can be very variable in quality, some of it is very good, some obviously not finished. I use a public domain communications package called Procomm, since it is superior to most commercial terminal emulation programs I've seen.

Operating System Enhancements

Mind you, you don't have to put up with the MS-DOS operating system, if you don't want to, there are various ways of 'improving' it. Like the PC hardware, the operating system can be enhanced by adding on what is sometimes called a 'Tweak-aid'. Examples of Tweak-aid used to improve the user interface of MS-DOS are the graphic environment managers such as Digital Research's GEM or Microsoft's Windows and the menu driven systems such as Xory and J.D.R.

These programs alter the look and feel of the operating system to make it easier to use, either by implementing the WIMP standard (Windows, Icons, Mouse Pointing device) or by displaying the disk files in your current directory on the screen with a menu of standard commands. Microsoft's Windows also falls into the category of programs that give your PC the ability to run more than one program at a time - the so-called multi-tasking. Other packages that give you the capability of DOSplus and Double DOS.

The End PC

This column has just scratched the surface of the IBM PC compatible computer world. The goal is to give an idea of what's out there, thereby explain why this stodgy little computer has become so popular. In future columns I hope to go into greater depth on how

to get the PC to do various things, some of which it was not designed to do. I shall also look at some specific programs and expansion cards that I have come across and actually used, not so much as a review, but as a user's comment and as solutions to specific problems which I might have some access.

Organisations & Products Mentioned

This section is far from complete, the prices are (where shown) exclusive of VAT and aren't necessarily the cheapest, they're just there as a guide.

Neo AT386-20, PC/AT with 300MB, 40 Mbyte Hard Disk, 1 Mbyte RAM Price - £1995.
Supplier - Neo Computing, 01-338 8818

YAMAHA CI - LCD Laptop PC/AT, 20 Mbyte Hard Disk, 2 Mbyte RAM Price - £995.
Supplier - YAMAHA Pulse, Condon Street, London

CP/M & MSDOS Users Group, 72 Mill Road, Hawley, Dartford, DA2 7E2

CompuLink User Group, Suite 2, The Sanctuary, Sartham, KT6 4DU, 01-898 84

Procomm v 2.4.2, communications software
Supplier - CompuLink User Group

Microsoft Windows, Graphics environment for PC Price - £99.
Supplier - PW Computer Supplies, 01-868 0548

Xory, DOS file and directory manager from Executive Systems Inc. Price - £44.
Supplier - PW Computer Supplies.

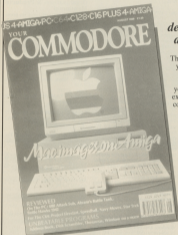
EDR, DOS and directory manager from Electronic Inc. Supplier - Quix, 01-894 2425

DESQview, multi-tasking environment Price - £49.
Supplier - CompaAdd, 0480 37315

Double DOS, Soft Logic's multi-tasking system Price - £24.95
Suppliers - Corporate Software, 07157 5461

Have

YOUR COMMODORE



*delivered to your door FREE!**

That's right, if you take out a year's subscription to Your Commodore we will make sure that it is delivered to your door each month at no extra charge*. Just fill in the coupon below and send it to the address given with a cheque, money order or credit card instructions to cover the cost of the subscription. We'll do the rest.

*Overseas subscription rates include postage.

UK	£13.00
EUROPE	£23.25
MIDDLE EAST	£23.50
FAR EAST	£26.40
REST OF THE WORLD	£28.00
in USA	\$42.00

Actual Rates on Request.

Send this form with your remittance to:
INFONET LTD,
5 River Park Estate,
Biller Lane,
BERKHAMSTED,
Herts, HP4 3HL,
United Kingdom

Please continue my subscription to Your Commodore with the issue.
I enclose my cheque/money order for £..... made payable to
ARCUS SPECIALIST PUBLICATIONS

Or Debit my Access/Visa number

Valid from to Signature

Name Address

Postcode

New subscriber Subscription renewal/extension (date as applicable)

Oops!



Make your programming easier with this package of extra commands for the dedicated Plus/4 user

By Mark Everingham

There are two problems with computers. The first is *temporal distortion*, and the second is *human fallibility*. If you are thinking to yourself "What the hell does that mean?" then don't worry - it's really very simple.

Temporal Distortion is the phenomenon by which time seems to become compressed in the presence of a computer. What this means in practice is that when you pop up to your Plus/4 for "five minutes before tea", you find that tea has long gone and that it's about time you were thinking about some breakfast. The answer to this is, of course, a computer alarm-clock, or OOPS! includes such a clock with advanced user-definable alarms and reminders so that you don't forget lunch or that vital doctor's appointment this afternoon.

The other problem with computers, which I mentioned before, is that although computers, cassette-

recorders and disk-drives are relatively reliable (Commodore's are notorious for being very reliable but equally slow), human beings are sadly not so dependable. It is all too easy to crash your Plus/4 or NEW the one and only copy of your latest million-seller article masterpiece. This event will send most rational human beings into a cry of OOPS! (Or similar four letter words).

In order to protect you from those awkward circumstances, OOPS! has an *oNEW*-type command and a unique automatic program-backup facility making it as difficult as possible to erase or lose an important program.

The features that OOPS! offers are now becoming more and more common as parts of programming languages on the latest generation of 16-bit micros such as the Commodore amiga, but have not until now been available for the BASIC users of 8-bit 16 home-computers like the Commodore Plus/4.

The OOPS! Basic Commands

OOPS! in practice consists of a program which adds thirty new commands to the normal Commodore Basic Operating-System. This may seem a bit of a daunting prospect at first, but they are all easy to use, and give an idea of how comprehensive the OOPS! system is. The commands can be split into six sections, namely *Clock Commands*, *Alarm Commands*, *Basic Commands*, *Saving Commands*, *Information Commands* and *Miscellaneous other commands*. In a moment I'll demonstrate the use of each of these, but first a short explanation of command syntax.

Each of the OOPS! commands is used in exactly the same way as any standard Commodore Basic commands. Each can be used in Basic Direct mode or in a Basic program, and all can be abbreviated as you please later. Because the commands behave like normal Basic commands, any syntax they have can be replaced with a string variable, for example both the below have exactly the same meaning (to set the 24-hour clock to 23:59:59).

```
SETCLOCK "23:59:59"
```

```
CLS="23:59:59":SETCLOCK CLS
```

This property of the commands is useful when writing your own programs, for instance if you can't remember how to work the *SETCLOCK* command, you could write a program which would ask you for the Hours, Minutes and Seconds, and then set the clock for you using something like *SETCLOCK HH:"MM" : "SS"* with the hours, minutes and seconds stated in *HH*, *MM* and *SS* respectively. This facility of the command means that you are not just limited to direct literal commands.

The OOPS! Clock Commands

The OOPS! system includes a 24-hour clock which does not interfere with the normal Basic 118 clock and can be made to appear on-screen all the time, whatever else you are doing with your Plus/4 so that you can run programs, print documents, display the directory or whatever you want - the clock will keep running. Below are the commands used to operate the clock.

```
SETCLOCK "HH:MM:SS"
```

The **SETCLOCK** command is used to set the initial time of the 24-hour clock, as you will obviously not always start using the OOPSI clock at the same time of the day. It's syntax is the time enclosed within inverted commas. **HH** is the time under the 24-hour system, **MM** is the number of minutes and **SS** the number of seconds. Each should be separated using a single colon. The command can yield the error message shown below which can all be trapped using the normal Basic TRAP command, and are displayed as normal.

SYNTAX ERROR - You have omitted the time from the command.

TYPE MISMATCH ERROR - You have typed a numeric parameter instead of the time string.

ILLEGAL QUANTITY ERROR - You have entered a time string which is either not of the form "HH:MM:SS" or has a number in it outside the valid range 0-23 for hours or 0-59 for minutes and seconds.

CLOCK ON/OFF

Once you have set the time on the 24-hour clock using the **SETCLOCK** command, you will want to be able to see what time the clock is showing all the time like a normal alarm clock. To do this just type **CLOCK ON**. From then on the clock is always displayed in the top-right corner of the screen. Note that the OOPSI system cannot be used with the high-resolution graphics screen, so the clock is always stable whatever you are doing. However, there may be times when you don't want the clock display to be visible, for example when writing a program that makes use of the full screen. To remove it from the screen type **CLOCK OFF**. Note that although the clock will now be invisible, it is still running and will still display the correct time when a subsequent **CLOCK ON** command is executed. The **CLOCK** command gives only one error message - **SYNTAX ERROR** indicating that you have left out, or possibly replaced the ON/OFF argument.

START and STOP

The **START** and **STOP** commands are used to respectively start and stop the OOPSI clock. Their main use is for accurate setting of the clock. For example if the time were showing 3p

to 10 o'clock you could use the program below to set the clock very accurately:

```
10 CSTOP
20 SETCLOCK "10:00:00"
30 PRINT "PRESS A KEY WHEN
TIME IS 10 O'CLOCK"
40 GETKEY K$
50 CSTART
```

Although it is not necessary to stop the clock while setting it, these commands are useful for setting the clock very accurately or taking the clock as a timer started by pressing a key like the button on a stopwatch.

The **START** and **STOP** commands require no arguments and so produce no error messages.

The OOPSI Alarm Commands

If you have anything important that you don't want to forget to do during the day, you can use the OOPSI alarm section of the clock to remind you to do this. As well as giving an audible warning, the alarm can also be made to display a message to remind you exactly what the alarm is set for.

SETALARM "HH:MM"

The **SETALARM** command is used to set up the time at which the alarm will sound. It works in the same way as the **SETCLOCK** command but the alarm time is expressed in only hours and minutes and not also seconds. The error messages the command gives are identical to those for the **SETCLOCK** command.

WARNING "Alarm Warning Message"

I have mentioned before that as well as making a noise when the alarm time is reached, OOPSI can be made to display a message as a reminder. This message appears inverted and flashing in the top-left corner of the screen to draw attention to itself. The **WARNING** command is used to set up this message (at which the alarm goes off, the message should be enclosed in inverted commas and be not more than 25 characters in length. The message may be made up of all alphanumeric characters except control codes such as **BYSON** and **BYSOFF**. The error messages that the **WARNING** command yields are shown below.

SYNTAX ERROR - You have left out the warning message.

TYPE MISMATCH ERROR - You have entered a numeric argument in place of the warning message.

STRING TOO LONG ERROR - You have entered a warning message that is more than 25 characters in length.

ALARM ON/OFF

Now that you have set up the alarm time and message, you can switch the alarm on using **ALARM ON**. If you wish to turn the alarm off to reset the time and message, use the **ALARM OFF** command. When the alarm time is reached, a warning sound is made and the warning message displayed on the screen. There are three ways of turning the alarm off. The first way is by typing **ALARM OFF**. The other two methods are described in the **OK** command section listed under Events. The **ALARM** command gives one error message:

SYNTAX ERROR - You have omitted the ON/OFF argument.

The OOPSI Event Commands

What are events? Well they are just what they sound like - events, occasions or happenings during the day. Still none the wiser? Well, most of us find that we have more than one thing that we want to remember to do on one day, or a single alarm is of limited use. For this reason, OOPSI has eight 'Event Alarms'. Each of these events works in the same way as the main OOPSI alarm and may be programmed with an event time and associated message. Within a single day you can get your Plus/4 to remind you of up to eight events, enough for those of you with the most densely packed social diaries!

SETEVENT EVENT #, "HH:MM"

The **SETEVENT** command works in the same way as the **SETALARM** command, but is used to set the time at which an event alarm will sound instead of the main OOPSI alarm. Its syntax consists of the number of the event you wish to set, in the range 1-8, followed by the event time enclosed in inverted commas and separated using a comma. The time is entered in an identical format to the **SETA-**

ALARM command. The command gives similar error messages, which are shown below:

Syntax ERROR - You have omitted the event number, event time or separating comma.

TYPE MISMATCH ERROR - You have entered a string instead of the event number or a number instead of the event time.

ILLEGAL QUANTITY ERROR - Either the event number you have specified is beyond the range 1-8, or you have entered an illegal event time.

MESSAGE EVENTS, "Event Message"

As mentioned before, each event alarm can be setup with a corresponding warning message in the same way as the GOPSI main alarm. The MESSAGE command is used to setup one of these messages. It takes as its arguments the event number in the range 1-8 and the message that you want to be displayed when the event time is reached. The message may be up to 23 characters long and should be enclosed in double-quotes and preceded with a comma as shown above. An event message is displayed flashing and inverted in the top-left of the screen in the same manner as the alarm warning on using the WARNING command. The MESSAGE command gives the error messages that follow:

Syntax ERROR - You have omitted the event number, event message or separating comma.

TYPE MISMATCH ERROR - You have replaced the event number with a string or the event message with a number.

ILLEGAL QUANTITY ERROR - The event number you have entered is outside the range 1-8.

STRING TOO LONG ERROR - The event message you have entered is longer than the allowed 23 characters.

ENABLE EV = 1, EV = 1, ..., EV = 8
Once you have setup an event with both event time and event message, you need some way of switching-on or enabling the event, so that when the event time is reached the correct sound is made and message displayed.

This is done by means of the ENABLE command.

The ENABLE command has a flexible syntax in that it can take one or more parameters rather like the Commodore Basic graphics commands. The basic argument is the number of the event that you wish to switch on, or enable. Thus to enable event number 4, just type **ENABLE 4**. If you want to enable more than one event, for instance events 1-4, it is a bit of a pain to have to type **ENABLE 1; ENABLE 2; ENABLE 3** etc. so instead you can simply type **ENABLE 1,2,3,4** saving down the amount of typing needed surely. Each of the event numbers should be separated from each other using a comma, and be in the normal range 1-8. Note that the event numbers can be specified in any order, so **ENABLE 1,2,3** has the same effect as **ENABLE 3,2,1**.

When an event has been enabled, at the specified time, the event message will be displayed in the top-left of the screen as explained before, and an alarm will sound. The event alarms are much less harsh, and gentler than the main alarm sound as they are far reminders that are not going to be such a matter of life and death as the main alarm. So if you are of a nervous disposition - use the event alarms instead of the main alarm as they are less of a strain on the heart! The event message that the ENABLE command yields are shown below.

Syntax ERROR - You have either left out all the event numbers, or finished the line with an unnecessary comma.

TYPE MISMATCH ERROR - You have replaced at least one of the event numbers with a string instead of a number.

ILLEGAL QUANTITY ERROR - At least one of the event numbers you have entered is beyond the allowed range 1-8.

DISABLE EV = 1, EV = 1, ..., EV = 8
The DISABLE command performs exactly the same function as the ENABLE command, but has the effect of switching-off or disabling an event. Its syntax is identical to that of the ENABLE command and the error messages it gives have the same meanings as for the ENABLE command.

Alarm Priorities and "Shutting Them Up!"

Some of you will have realized while reading the explanations of the GOPSI alarm and event commands that it is perfectly possible to set up all eight events and the main alarm to go off at the same time. What would happen if you did that? After all you can't make 9 inches and display 9 messages all at once, can you? The answer is NO, of course not, so a solution is reached in the form of alarm priorities.

Each event and the main alarm has a priority over the others. The alarm comes first, followed by the eight events in numeric order 1-8. To understand this, imagine that you had set the alarm and events 1,3,5,7 all to sound at 10 o'clock. Because the main alarm has the highest priority, the alarm noise will be made and the alarm message displayed instead of all the others. "OK," you say, but if I can only hear and see the alarm, then what about events 1,3,5 and 7 - they might be important, too! To find out, read on...

OK (CTRL-0)

When an alarm or event goes off, there are several ways you can turn it off:

- If it is the alarm, you can type **ALARM OFF**
- If you know which event is occurring you can type **DISABLE EVENT #**
- The **OK** command

Methods (a) and (b) are all right, but a bit messy and, if you don't know which event is sounding, then you will have to try each one of the eight in turn to find out. The solution to this is the **OK** command. This command simply switches off whatever is occurring at the moment - the main alarm or the current event. Using this command, you don't have to know which event is occurring, and it also saves you quite a bit of typing.

I discussed earlier the question of what happens when more than one alarm or event is occurring, meaning that you can only see the one with the highest priority - What about the others? Well, when you come **OK**, it switches-off the event which is currently displayed on screen, and then displays the next one down the priority scale. For instance imagine that the alarm and all the events were set for the same time. When that time was

crashed, you would see the alarm message. When you type **OK** return the alarm is switched off and then event 1 is displayed. If you type **OK** return again then event 2 is displayed and so on until all the events have been acknowledged, whereupon the message in the top-left of the screen is erased and the sound turned off again. If you only wanted one alarm, you could use each of the eight events to give you a message of up to 225 characters, using the **OK** command to read the next section of message, with all events set for the same time.

For those of you who find typing two characters and pressing **RETURN** a bit arduous, you can achieve the same effect as typing **OK** by pushing **CTRL-O**.

OOFS! Saving Commands

The third important section of **OOFS!** is its automatic saving facility. Using this function, you can get your Plus/4 to automatically save a backup copy of the current program onto disk, so that you don't have to interrupt your work too far if you make a disastrous mistake. You can also make **OOFS!** mark each file it saves so that you have running log of different versions of your program on disk.

SETSAVING MINUTE

As mentioned before, **OOFS!** can be made to automatically save backup copies of the program that you are working on, so you need to tell **OOFS!** when it is to save these backup copies. For instance, if you are writing an important program, you may want it saved every five minutes (just to make sure that you don't lose it, but, if you are writing a long program, the saving takes a long time so you may want it to be saved every fifteen minutes or half an hour instead. The **SETSAVING** command is used to specify how often **OOFS!** should save backup copies. Its only argument is the period of time between successive savings, expressed in minutes. So, if you want your program saved every ten minutes, **SETSAVING 10** will set this up. You can use any time period between every one minute and every hour (60 minutes). The **SETSAVING** error messages are as follows:

Syntax ERROR - You have omitted the **MINUTE** argument.

TYPE MISMATCH ERROR - You

have replaced the **MINUTE** argument with a string.

ILLEGAL QUANTITY ERROR - The time period you have specified in minutes is beyond the range 1-60.

DEVICE DEV#

For those of you lucky enough to have more than one disk-drive attached to your Plus/4, **OOFS!** includes the ability to cope with this. When you are writing a program, you might want to have a utility disk containing an Assembler or Font Editor etc. in disk-drive 8, and an **OOFS!** backup disk in disk-drive 5. To do this, you would simply type **DEV#DEV#** - the **DEV#** argument being the device number of the disk-drive on which **OOFS!** could save backup files. The device number can be in the range 0-11 for four disk-drives, but you cannot use device 0 (The Dataset) as this is really too slow for automatic saving. The **DEVICE** error messages are shown below.

Syntax ERROR - You have omitted the device number argument.

TYPE MISMATCH ERROR - You have replaced the device number with a string instead of a number.

ILLEGAL DEVICE NUMBER ERROR - The device number you have specified is outside the valid range 0-11.

FILENAME "Filename"

When a backup copy is saved onto disk, it will obviously need to be given a name on the disk, and the **FILENAME** command is used to specify this name. The name argument should be enclosed in inverted commas and may be of up to ten characters in length. It may be made up of any alphanumeric characters but may not include spaces. If you do include a space in the name, the actual name will be truncated by the space character. For instance, entering **FILENAME "HALLO THERE!"** results in a file-name of "HALLO". The **FILENAME** error messages are as follows:

Syntax ERROR - You have failed to enter a file-name.

TYPE MISMATCH ERROR - You have either replaced the file-name argument with a numeric argument or have not enclosed the name in inverted commas.

STRING TOO LONG ERROR - The file-name you have entered is longer than ten characters.

OOFS! Backup File Marking

It is a little pointless to keep on saving backup copies of a program onto disk unless you know what each file contains. For example, if you were saving backup copies every two minutes and each of them were called simply "PROGRAM", you might have some files which were saved two hours ago when you were just starting the program, and some which had only just been saved. Obviously you need to know which is which! **OOFS!** offers several ways of marking your programs, or you can just choose for the last backup copy to be erased every time a new copy is saved. This section discusses the various commands for using the file-marking facility.

VERSION VER#

The first way of marking a backup copy used by **OOFS!** is with a version number indicating at what stage of development that backup-copy is. You can tell **OOFS!** which version number of the program you want the first backup copy to be marked with by using the **VERSION** command with **VER#** being the version number. Thus if you want to start with version number 45, enter **VERSION 45**. **OOFS!** can cope with version numbers in the range 0-999, and this should be quite sufficient unless you are in the habit of saving 10000 backup copies of programs! The **VERSION** command gives the following error messages:

Syntax ERROR - You have omitted one of the **VER#** arguments.

TYPE MISMATCH ERROR - You have replaced the version number argument with a string.

ILLEGAL QUANTITY ERROR - The version number you have specified is outside the valid range 0-9999.

NUMBERMARK

This is the first of the commands to tell **OOFS!** how to mark backup copies of a program. The **NUMBERMARK** command makes **OOFS!** add the current version number to the end of the file-name. For example if the file-name were set to "BACKUP" and the version number to 39, the files saved on disk would be under these names:

"BACKUP V50"... "BACKUP V51"...
"BACKUP V52"...

As you can see, OOPS! automatically increments the version number each time it saves a backup copy. In the unlikely event that the version number should exceed 9999, it is reset to zero. The NUMBERMARK command takes no arguments, so needs no error messages.

TIMEMARK

As an alternative to marking backup copies with a version number, you can add the current clock time to the filename. Instead, this allows you to pinpoint the exact version of a program you want, to the nearest minute. For instance, if you had entered SETSAVING in FILENAME "TEST": TIMEMARK, the backup copies saved would have been the names shown below, assuming that saving started at two o'clock in the afternoon.

"TEST 14:00"... "TEST 14:05"...
"TEST 14:20"...

This function of OOPS! is most useful when you are making frequent changes to a program. The TIMEMARK command gives no error messages.

UNMARKED

If you don't want the backup copies of your program to be marked with version number or time, you can enter the command UNMARKED to disable these facilities. From then on, backup copies will have the name given to them in the FILENAME command, with no suffix. However, there is a fallacy in this - You cannot have more than one file with the same name on a disk, so after the first backup copy has been saved, all subsequent ones will not result in a file being saved onto the disk. To overcome this problem, see the next section on the REPLACING command.

REPLACING ON/OFF

As more and more backup copies of a program are saved on a disk, so the amount of free space remaining gets smaller and smaller until the disk overflows. If you are working on a long program there may not be room for more than three copies or so on one disk. The answer to this problem and also to saving backups with no suffix is to erase the last backup copy before you save the new one. In this way the

amount of room left on a disk will only decrease by the amount you add to a program between saves. For instance if you had entered VERSION 1: NUMBERMARK: FILENAME "OOPS": REPLACING ON, the following process would take place:

Save "OOPS" V1... Erase "OOPS-V1" and Save "OOPS" V1...

As you can see, this saves a large amount of space on a disk, but also means that your files are not quite so secure because the last copy has to be erased, and it is also slower than just saving new copies. When the first copy of a program is saved with REPLACING ON, OOPS! senses that this is the first save, and so does not attempt to erase a non-existent previous file REPLACING gives only the SYNTAX ERROR message, if you omit ON & OFF.

SAVING ON/OFF

When you have decided what to call your backup copies, how to mark them and whether or not to erase the last one etc, you can enable the OOPS! saving function using SAVING ON. To switch it off again just use SAVING OFF. Once saving is switched on, when it is time to save a backup copy (as dictated using SETSAVING) the following happens:

1. The line you are entering is erased.
2. The messages below are displayed and their corresponding actions carried out:
ERASING "Last File-name" if
REPLACING ON
Last backup copy is erased if
REPLACING ON
SAVING "In New File-name"
3. Your BASIC program is saved.
4. The Computer returns to Direct mode.

As you can see from the above, each time a program is saved the line that you are entering at that time has to be erased. This may seem a bit inconvenient, but when you think about it, the more you can lose is 80 characters of program, whereas if you hadn't saved your program you could have lost all a kilobytes of it!

The saving function of OOPS! only waits from Basic Direct mode (When entering programs). If it is time to save a backup copy while a program is running, OOPS! simply waits for the next time and then tries again. In this way it does not interfere with the

running of the program.

SAVING gives the SYNTAX ERROR message if the ON & OFF argument is left out.

STORE (CTRL-S)

Sometimes, you may want to save extra backup copies of a program separately from the timed saving function. It is annoying to have to wait ten minutes (or whatever) for the next automatic backup, but you may still want the program to be saved with a version number or the time etc. To save a program at any time, using the marking etc., you can simply type STORE. This has the same effect as if it were time to save a backup copy automatically. Additionally, the STORE command can be used even when automatic saving has been disabled using SAVING OFF.

Again, for those who consider typing five characters a major undertaking, you can achieve the same effect as typing STORE by simply pressing CTRL-S. Yes, I would have liked to use CTRL-S too, but Commodore beat us to it, for stopping printing!

OOPS! Information Commands

As you can see, OOPS! has a large number of functions which, when you first use the system, can be more than a little confusing. To help you when you are just starting to use OOPS! I have included in the program five commands to give you information on all aspects of OOPS! from its commands to the current settings of the alarm or the automatically-saving. It now discusses each of these in turn.

COMMANDS

OOPS! has thirty commands in all, and you may find it hard to remember all of them. If this is the situation, just type COMMANDS. This results in a heading and a list of all thirty commands in reference. You should then not find it too hard to remember which command does what.

CLINFO

Apart from the COMMANDS command, there are also four specialized information commands which cover the clock, alarm, events and saving functions of OOPS! The first of these is CLINFO.

On entering CLINFO, a list of information about the OOPS! clock is printed on-screen, which is as follows:-

TIME: HH:MM:SS - The current time of the clock.
DISPLAY: ON/OFF - Whether the clock display is on or off as set up by the **CLOCK ON/OFF** command.
STATUS: STARTED/STOPPED - Whether the clock is running or not, as dictated using the **START** and **STOP** commands.

ALINFO

The **ALINFO** command is similar to the **CLINFO** command, but is used to print information about the **GOOSE** alarm instead of the clock. It gives the information shown below.

TIME: HH:MM -The time at which the alarm is set to go off.
STATUS: ON/OFF - Whether the alarm is enabled or disabled using the **ALARM ON/OFF** command.
MESSAGE: "Warning" - The warning message to be displayed when the alarm time is reached, as defined using the **WARNING** command.

EVINFO

The **EVINFO** command displays a list of information about the current settings for each of **GOOSE**'s eight events. All are listed under a set of headings which may be explained as follows:-

- The event number to which this line of information refers.
TIME - The time at which this event will occur.
STS - The status of this event - whether or not it is on or off as defined using the **ENABLE** and **DISABLE** commands.
MESSAGE - The event message assigned to this event using the **MESSAGE** command.

Each line shows the same information, but for event number as shown on the left of the line.

SVINFO

The last of the **GOOSE** information commands is the **SVINFO** command which is used to display the current settings for the automatic saving function in **GOOSE**. The list of information the command gives is shown below.

TIMING: MM - Period between subsequent saves in minutes as defined by the **SETSAVING** command.
STATUS: ON/OFF - Whether or not the automatic saving is on or off, as set up using **SAVING ON/OFF**.

MODE: UNMARKED/NUMBERMARK - The current method for marking backup copies.
TIMEMARK

REPLACING: ON/OFF - Whether or not the replacing or saving function is enabled by the **REPLACING** command.

VERSION: VVVV - The current version number set up using the **VERSION** command.

DEVICE: DD - The current saving device number, as set up using the **DEVICE** command.

FILENAME: "Filename" - The current save file-name defined with the **FILENAME** command.

Miscellaneous Commands

As well as the specific function commands and information commands, I have provided **GOOSE** with a few extra commands to make life a bit easier when using the system. This section deals with these three remaining commands, namely **WINDOW**, **RECOVER** and **RESET**.

WINDOW MDW :

You may have noticed that when the clock display is enabled, scrolling the screen causes the clock to blink in an annoying way. You may also have seen that some strange effects can be achieved using the **Delete** and **Insert** keys on the top row lines of the screen. It would be better if all work took place in an area of the screen not affected by the clock display or flashing messages, which would remedy the problems outlined above. The answer is the **WINDOW** command.

The **WINDOW** command is used to reduce the usable area of the screen to an area below the clock display so that it does not interfere with any work you're doing. It is the equivalent of moving the cursor to the corners of the area and pressing **ESC/F** and **ESC/S**. Two windows are available which are similar to the **ESC-N** screens, but with the top edge below the clock display. The **WINDOW** command takes only one argument - **WINDOW #** enables the **ESC-N** equivalent window, and **WINDOW I** the **ESC-R** equivalent. These are best explained by trying them out yourself. The edge coordinates for the respective windows are as follows:

WINDOW R: Top-left Corner: (0,2)
 Bottom-Left Corner: (38,24)

WINDOW I: Top-Left Corner: (1,3)
 Bottom-Left Corner: (38,25)

The **WINDOW** command gives the error messages shown below.

INVALID ERROR - You have entered the window number argument.

TYPE MISMATCH ERROR - You have mistakenly replaced the window number with a string.

ILLEGAL QUANTITY ERROR -

The window number you have specified is neither 0 or 1 as it should be.

RECOVER

In addition to the automatic backup facility of OOPS!, to stop you losing valuable programs, OOPS! also has an *UNDO*-type command **RECOVER**. If you enter a program and then erase it from memory using the **NEW** command, you can get it back by typing **RECOVER**. The command will of course not work after you press the **RESET** button on your Plus 4, doing this erases the OOPS! program code. Some of the time you may be able to get your program back in this situation by reloading the OOPS! program; then typing **RECOVER**. You should not try to use the command when you've just switched your computer on, because it will give some funny results with no program data present.

The **RECOVER** command has no arguments so it yields an error message.

RESET

The final OOPS! command, **RESET**, is used to reset the OOPS! data to its default values. This turns the clock on, resets all the events and saving on, to the values that they have when you first use OOPS!. These default values are setup as typical values such as you might frequently use. These default settings are shown here.

OOPS! Abbreviations

Like the normal Commodore Basic commands, OOPS! commands can be typed abbreviated to save on typing. When typed into a program abbreviated, they are converted into their full form on listing so are just as easy to understand as typing them in fully. A list of the OOPS! command abbreviations is provided here. All abbreviations take exactly the same syntax as the full command.

OOPS! Kickstart File Generator

You should now have sufficient knowledge to be able to use all of the OOPS! commands competently. I'll now discuss the OOPS! Kickstart File Generator program (Listing # 2).

Most of the time when using OOPS! you will find that you always use the same saving setup, or the same event times and messages, so it is a

Command	Arguments	Minimum Abbreviation
SETCLOCK	"HH:MM:SS"	S SHIFT-E
CLOCK	ON/OFF	C SHIFT-L
SETALARM	"HH:MM"	SET SHIFT-A
WARNING	"Message"	W SHIFT-A
ALARM	ON/OFF	A SHIFT-L
SETEVENT	E# "HH MM"	SET SHIFT-E
MESSAGE	E# "Message"	M SHIFT-E
ENABLE	E#1..E#N	E SHIFT-N
DISABLE	E#1..E#N	D SHIFT-I
OK	-	OK
SETSAVING	SAV	SET SHIFT-S
DEVICE	DEV	D SHIFT-E
FILENAME	"Filename"	F SHIFT-I
VERSION	VER	V SHIFT-E
NUMBERMARK	-	N SHIFT-U
TIMEMARK	-	T SHIFT-I
UNMARKED	-	U SHIFT-N
REPLACING	ON/OFF	R SHIFT-E
SAVING	ON/OFF	S SHIFT-A
STORE	-	S SHIFT-T
CSTART	-	C SHIFT-S
CSTOP	-	CST SHIFT-O
RESET	-	RE SHIFT-S
RECOVER	-	RE SHIFT-C
WINDOW	0/1	W SHIFT-E
COMMANDS	-	C SHIFT-G
CLINFO	-	CL SHIFT-I
ALINFO	-	AL SHIFT-I
EVINFO	-	E SHIFT-V
SVINFO	-	S SHIFT-V

bit of a waste of time to keep on typing the list of SETEVENT commands or whatever to set these values every time you use OOPS! or every time you reset your Plus 4. It would be much easier to just write a short "Kickstart" program to set up the OOPS! functions so that all you need to do whenever you use OOPS! is to load and RUN this program. This is of course possible, and you can write such a program like any other Basic program using the OOPS! commands, but as an alternative, I have included the OOPS! Kickstart File Generator which can write such a program itself - A program writing another program!

When you run the Kickstart File Generator (Henceforth referred to as KICEGEN), it will ask you a series of questions. When it has done this, it will create a Kickstart program and then save it to disk or cassette. In this section I'll discuss all the questions KICEGEN asks, with the expected replies shown within parentheses.

(1) **WINDOW (0/1)** Enter which of the windows you wish to start using OOPS! within: The ESC-N type window 0, or the ESC-R window 1.

(2) **CLOCK TIME (HH/MM/SS)** Enter the time you wish the OOPS! clock to be set to when you first start using OOPS! Note that when you are entering times into KICEGEN, you must separate the Hours, Minutes and Seconds with any character EXCEPT a colon. For example 00/00/00, 10-00/00 and 10 00 00 are all valid, but 00:00:00 will not work as the BASIC input routine cannot cope with it.

(3) **CLOCK DISPLAY (ON/OFF)** Enter whether or not you want the clock display switched ON or OFF when you start using OOPS!

(4) **DO YOU WANT TO SET THE ALARM? (YES/NO)** If you don't want to bother with setting the OOPS! alarm, just enter NO and go on to section (5). If you do want to set the alarm, follow the steps below.

(5) **ALARM TIME (HH:MM)** Enter the time you want to set the alarm to.

(6) **ALARM WARNING (Message)** Enter the message that you want to be displayed on-screen when the alarm time is reached.

(7) ALARM STATUS (ON/OFF)

Enter whether or not you want the alarm to be switched ON or OFF when you start using GOPS!

(8) DO YOU WANT TO SET ANY EVENTS? (YES/NO)

If you don't want any events set up when you start using GOPS! enter NO and go to step (12). Otherwise, enter YES and follow the steps below.

(9) EVENT X TIME (HH/MM)

When you have answered that you do want to set some of the GOPS! events, you will be asked a series of questions for each of the events 1-6. The prompts displayed are of the same form for each event, with X being the event number to which the query refers. E.g. "EVENT 5 TIME" or "EVENT 7 STATUS". The first entry, EVENT TIME is the time you want this particular event to be set to, but if you don't want to set this event, just press RETURN without entering anything and go on to the next event.

(10) EVENT X MESSAGE (Message)

Enter the message you wish to be displayed when the time for event X is reached.

(11) EVENT X STATUS (ON/OFF)

Enter whether or not you want event number X to be enabled (ON) or disabled (OFF).

(12) DO YOU WANT TO SET THE SAVING? (YES/NO)

If you want to set up the automatic saving facility of GOPS! enter YES and follow the steps below. If not, enter NO and skip to the end of this section.

(13) SAVE TIMING (0-60)

Enter the period of time (in minutes) between subsequent automatic saves.

(14) DEVICE NUMBER (0-11)

Enter the device number of the disk drive on which you wish GOPS! to save its automatic backup copies.

(15) FILENAME (Name)

Enter the filename under which you want backup copies of a program to be saved.

(16) VERSION NUMBER (0-9999)

Enter the version number that you wish the first backup copy to be marked with when VERSIONMARK is executed.

(17) MARKING MODE (N,I,I)

Enter the first character of the mode

by which you wish backup copies to be marked. This should be one of (N)umbermark, (T)imestamp or (U)ppermark.

(18) REPLACING (ON/OFF)

Enter whether or not you wish the previous backup copy to be erased when a new backup copy is saved.

(19) SAVING STATUS (ON/OFF)

Enter whether or not you want the automatic saving function of GOPS! to be switched ON or OFF when you first start using the system.

When you have answered all the questions the screen window will clear and the message "COMPILING KICKSTART FILE..." will be displayed. After this, program lines will gradually be displayed which set up all the GOPS! functions as you have specified by answering KICKGEN's questions. When KICKGEN is displaying program lines, what it is doing is actually building a BASIC program into another area of memory from itself. When it has finished this process, KICKGEN displays the message "COMPILING COMPLETE - TAPE/DISK (T/D)?" If you want to save the Kickstart program to Tape, press "T" or if to Disk, press "D". KICKGEN then asks you to insert a disk or tape, and then when you press

RETURN it will save the Kickstart program onto the tape or disk.

When KICKGEN has saved the Kickstart program, from that time you can set up your preferred GOPS! functions simply by typing the following.

DLOAD "KICKSTART" if you are using a disk-drive.

LOAD "KICKSTART" if you are using a Datasette as a more secure media.

When the program has loaded, simply type RUN and hit RETURN. Of course, don't forget to load the GOPS! main program before!

That completes our discussion of the GOPS! commands and KICKGEN program. It's now discuss the most important aspect of GOPS! - Getting the thing started!

The GOPS! system comes in two parts which are the GOPS! Basic Loader (Listing # 1) and the GOPS! Kickstart File Generator (Listing # 2). The Basic Loader is simply used to create the machine-code GOPS! program, and is used as follows:

Enter Listing # 1 and RUN. The program will give messages to allow you to correct the data lines. Note that you should start the section of data with line 100. To help you, you can use the line below to set up automatic line numbering, and function key 1 to produce the DATA command with one keypress:-

KEY 1, "DATA": AUTO 10

When you have entered Listing # 1 and corrected all your mistakes, the program will ask you whether you want to save the GOPS! programs to disk or tape. Press "D" or "T" to choose which and then insert a diskette or tape and press RETURN to save the programs. The programs saved are: (a) BASIC Loader and (b) GOPS! Machine-Code Program. When these programs have been saved, to use the GOPS! system just reset your Plus/4 and type the following:

**For Tape Users: LOAD "GOPS!"
RETURN RUN RETURN**

**For Disk Users: DLOAD "GOPS!"
RETURN RUN RETURN**

When you type RUN, the GOPS! commands are loaded, a title screen displayed and the clock switched on. You can then write programs or use GOPS! commands. If you use the RUN/STOP-RESET combination to escape from a machine-code program, typing X RETURN from TED-MON will get you back into Basic and re-enable the GOPS! functions.

As mentioned before, the saving facility of GOPS! cannot be used with a Datasette, but all the clock, alarm and event functions may be used with either disk or tape.

The KICKGEN program (Listing # 2) can be entered just like any other normal BASIC program. It can be used as shown below.

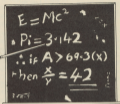
**For Tape Users: SAVE "GOPS!"
KICKGEN RETURN**

**For Disk Users: DSAVE "GOPS!"
KICKGEN RETURN**

The GOPS! KICKGEN program may be used either with or without GOPS! resident in memory.

Believe it or not, that concludes this article about GOPS! Now then, I'll just format that disk... there we go... hang on, what does that disk-label say? "GOPS! Master disk"... Oh ****!

Logically Speaking



Get your thinking caps on for this excursion into
Mathematics and Logic

By Pete Gerrard

I would imagine that many of you have dabbled with the game of Chess at one time or another, or perhaps a simpler game of Draughts on the same board. The complexities of Chess can be baffling for the beginner, the rules of Draughts are more easily understood, but regardless both of them remain fascinating games for expert and novice alike. The topic under discussion now revolves around the board rather than the pieces on it, so, if you have such a board, you may care to get it out and study it. If not, you will just have to imagine an eight by eight board; don't worry about black and white squares, just stick to the eight by eight board, and we can continue.

After a quick calculation you might say that there are 64 squares on the board, and indeed there are 64 squares that are just one square in width and height, but what about the squares that are two squares in width and height, or three squares, or even eight

squares? Of course, there is only one of the latter, but there are many more of the other sizes and all the intermediate ones. How many are there in total?

As if that were not puzzling enough, consider also that there are many, many more rectangles formed than there are squares. Rectangles can be two squares by one, or four squares by five, and there are far more of them than the more conventional squares.

So how many rectangles can be formed on a standard Chess board? and having discovered that one, what is the formula for the combined number of rectangles and squares?

Fortunately for us there are mathematical formulae for solving these particular riddles. However, if I were to simply tell you the answer for a square the size of a chessboard, that would not tell you the answer for a six by six board, or a square greater than six by six. Hence it is here once more in the form of the accompanying program. This will allow you to solve the riddle of how many squares and rectangles there are for a whole range of squares from the simple two by two to the much more

complex twenty by twenty.

Of course, you could use even larger numbers, although the square itself would not be drawn on the screen and the listing would need amending accordingly.

First, let us consider the formulae. If we have a variable 'A' equal to the size of the square (and thus 'A' would equal 8 for our Chess board problem), then the combined total of squares and rectangles that can be formed is given by the formula $(A^2 + A) * (A + 1) / 6$. This gives us the total number, and the individual number of squares is given by the formula $(2 * A^2 + A^2 * A) / 6$. From this we can of course work out the number of rectangles easily enough, but just for the sake of completeness, the formula that itself gives us the number of rectangles not including squares is $(2 * A^2 + A^2 * A - 2 * A) / 12$. A complicated one!

This brings us to the program listing, and a simple enough one it is to one once we know the relevant formulae. Lines 10 and 20 serve to identify it, while lines 30 and 40 select our colour scheme and clear the screen for neatness. Lines 50 to 60 then allow the

user to input a number in the range 1 to 30, and could be amended if bigger squares were required, although you would then be advised not to attempt drawing them and skip straight to line 140.

Line 78 clears the screen again, lines 80 to 130 define and draw the square in its entirety, before we get to the actual calculations. After a brief statement in line 140, line 150 then works out the total number of squares and rectangles using the formula given earlier. Line 160 calculates the number of squares and the number of rectangles including squares. That this Formulae do actually give us the correct answer can soon be verified by drawing a trivial square two by two, for instance, which is easily calculated by the eye.

The answers are displayed by lines 170 to 190, and line 200 rounds everything off and prevents the program falling through to line 210 and producing a RETURN WITHOUT GOSUB error.

That is the program, there are the formulae, and an interesting little mathematical diversion they form as well. However, an article of this nature would perhaps complete without giving you some work to do as well. As we have seen, there are formulae for squares, but what about triangles? Imagine an equilateral triangle that is divided into eight smaller triangles along its base, eight along its two remaining sides, and the interior filled in with triangles of the same size. Rather like an eight by eight chess board, we might be said to have an eight by eight by eight triangle. Remembering that triangles can be formed upside down as well as the right way up, what is the total number of triangles contained within the big triangle, saying that the big triangle as a whole adds one to the total, just as the center Chess board adds one to the number of squares. Easy if you know the formula, but if you don't....17

Switching The Bases

As human beings we are accustomed to counting in a numerical system that uses ten as its base, hence we arrive at what is called a decimal system. This is not particularly convenient for computers which are ideally suited to the so-called binary system.

As far as computers are concerned an electronic circuit can be either

turned on or off, it only has two possible values associated with it, and, if we give those values the numerical associations of zero and one, we have the basis for this binary system.

Binary and decimal do not mix happily though, and sometimes counting systems comes into the reckoning. This is apparent here, too, using the number 1234 as its base. Unfortunately the Arabic numerals from which we own derive only ascend in far as 9, using the symbols 0, 1, 2 and so on. Rather than inventing new symbols, hexadecimal uses letters of the alphabet. These letters are A, B, C, D, E and F, and this gives us hexadecimal numbers of the form 12FE for example.

In order to convert numbers from one numerical system to another, we need only know the base that is being used. For instance, the decimal number 1234 is easily translated as 4 plus 3 times 10 plus 2 times 10 to the power 2 plus 1 times 10 to the power of 3. The further left we go in the number, the greater the power of ten that is being used. To take our rather hexadecimal example of 12FE, knowing that E is equivalent to ten decimal number 14 and F is equivalent to the decimal number 15, the array at 14 plus 15 times 16 plus 2 gives 16 to the power of 2 plus 1 gives 16 to the power of 3, or 1790 in our more familiar decimal system.

Thus the basis for all counting systems rests in the base being used, and by multiplying that base number raised to the appropriate power by the number concerned, we can easily convert from one numerical base to another. To give another example, let us assume a numerical system that uses the base 7. An unusual choice, perhaps, before converting one numberless. If we look at a number of 1234 which was expressed using this base of 7, to convert it back into our own decimal system we have 4 plus 3 times 7 plus 2 times 7 to power 2 plus 1 times 7 to power 3. This equates to a decimal number of 466.

Now we arrive at the mathematical diversion. To begin with, we will be using a base of 8, a base of 5, and a base of 7. In other words, three different bases. What we want to find is a solution to the following equation, assuming the variable 'A' to be the number we are looking for: A (base 8) minus A (base 5) equals A (base 7). The same number, but using a

different numerical base for each time that it is expressed. What if, more, we want to find a four digit number that will obey the equation and, to complete matters will further, we would like to find a number which has its second and third digits equal. A number like 1233, for example, although that is not the answer.

Many such problems can be posed, and I shall leave you with one to ponder on at the end of this article. For now, let us consider the problem as it stands and the program below that enables us to solve it. With the rules of the problem clearly expressed, and the method of translating numbers from one base to another already defined, the listing should pose no problems.

Line 10 serves to identify the program and its author before line 20 sets our string values, while line 30 clears the screen. The loop in line 40 'breaks up' to cover all possible four digit numbers, and with line 50 will do us up a string variable so that we can check for the presence of two equal digits in the second and third positions within those four digit numbers.

If those digits are not the same then line 70 sends us straight to line 100 to continue the search, ultimately concluding with that line if no such number is found. However, if the second and third digits are the same then we go in turn to lines 110, 130, 150 and 170 in order to convert that number to base 8, 5, 7 and 10 respectively, before going to line 190 and displaying the numbers on the screen. So inform the user of the program being made.

If the equation now holds true then line 90 tells us of the relevant number before halting the program with an END statement, otherwise we carry on with the loop in line 40.

The first number that conforms with our equation is the decimal number 1532. That is, 1532 in base 8 minus 1532 in base 5 equals 1532 in base 7. The decimal number 2698 also obeys the rules.

That is a simple example of the kind of problem that can be set when using a mixture of bases for counting purposes. Now consider a different sort of problem, using the equation A (base 8) equals A (base 5) plus A (base 7). If the numeric variable 'A' is representing the decimal number 1717, what are the values of a_8 and a_7 in other words, what bases are we using?



PROGRAM: BAGES

```

00 10 REM BAGES BY PETE GERRARD
03 20 X=5000:10:POKE X,C:POKE X
+1,0:PRINT CHR$(150)
09 30 PRINT CHR$(147)
18 40 FOR I=1000 TO 9999
20 50 A=1
32 60 A$=MID$(STR$(A),2)
55 70 IF MID$(A$,2,1)>MID$(A$,
3,1) THEN 100
65 80 GOSUB 130:GOSUB 130:GOSUB
150:GOSUB 170:GOSUB 190
25 90 IF A$=B$=C$ THEN GOSUB 18
0:PRINT:PRINT B$, " IS ONE PD
SSIBLE ANSWER." :END
08 100 NEXT I:PRINT "NO NUMBER
FOUND." :END
74 110 REM BASE 6
87 120 A1=0:FOR J=1 TO 4:X$=MID
$(A$,J,1):X=VAL(X$):A1=A1*X*
5*(4-J):NEXT A1:A1:RETURN
30 130 REM BASE 5
00 140 A1=0:FOR J=1 TO 4:X$=MID
$(A$,J,1):X=VAL(X$):A1=A1*X*
5*(4-J):NEXT A1:A1:RETURN
48 150 REM BASE 7
02 160 A1=0:FOR J=1 TO 4:X$=MID
$(A$,J,1):X=VAL(X$):A1=A1*X*
7*(4-J):NEXT A1:A1:RETURN
41 170 REM BASE 10
38 180 A1=0:FOR J=1 TO 4:X$=MID
$(A$,J,1):X=VAL(X$):A1=A1*X*
10*(4-J):NEXT A1:A1:RETURN
88 190 REM DISPLAY NUMBERS
06 200 PRINT CHR$(147);
70 210 PRINT "BASE 6 =",A1
74 220 PRINT "BASE 5 =",B1
52 230 PRINT "BASE 7 =",C1
27 240 PRINT "BASE 10 =",D1:D1=
MID$(STR$(D1),2)
74 250 RETURN

```

PROGRAM: SQUARES

```

03 10 REM SQUARES BY PETE GERRARD
20 20 REM FOR THE COMMODORE 64
40 30 X=5000:10:PRINT CHR$(147)
+POKE X,C:POKE X+1,0:PRINT C
HR$(150)
80 40 PRINT CHR$(147)
20 50 INPUT "SIZE OF SQUARE (E.
S. S FOR 6 BY 6):SPACE(2)A
84 60 IF A=2 OR A=20 THEN 40
02 70 PRINT CHR$(147)
02 80 A$=1:A$="":B$="" :C$=""
06 90 FOR I=1 TO A:A$=A$+CHR$(I
70):NEXT A$:B$=CHR$(173)+A$+CHR
$(173)
74 100 FOR I=1 TO A:B$=B$+CHR$(
173)+NEXT B$:C$=CHR$(173)+B$+CH
R$(173)
58 110 FOR I=1 TO A:C$=C$+CHR$(
177)+NEXT C$:C$=CHR$(173)+C$+CH
R$(173)
58 120 REM STRINGS DEFINED
25 130 PRINT A$ :FOR I=1 TO A:PR
INT B$:NEXT:PRINT C$
08 140 X=50:Y=0:GOSUB 210:PRINT
" SQUARES & SQUARES"
74 150 A$=A1:Y=Y+3:THR=(A$*A$)
*(A$*A$)*1/4
26 160 B$=(2*AAAAA+3*AAA+2)*Y:Y
B$=(3*AAAAA+2*AAA+3)*AAA+2
74)*12
20 170 GOSUB 210:PRINT "TOTAL S
QUARES & Y*Y+1:GOSUB 210:PR
INT "RECTANGLES =",THR
22 180 Y=Y+2:GOSUB 210:PRINT "S
QUARES SPACE =",A$
70 190 Y=Y+2:GOSUB 210:PRINT "R
ECTANGLES =",NR
46 200 X=0:Y=20:GOSUB 210:END
210 PRINT CHR$(19);:FOR I=1
TO Y:PRINT:NEXT:FOR I=1 TO X
:PRINT CHR$(20);:NEXT:RETURN

```

Binders

Organise and protect your disk with Commodore Disk User disk binders and data disks.

Why not keep your Commodore Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder comes complete with 10 disk sleeves to organise and protect your program disks. Why not buy a disk binder to house all of your data disks? We can even supply Commodore Disk User data disks. The Commodore Disk User logo immediately identifies your disks and there's room to title them and document the disks details. Send for your disks and binders now!

Prices are as follows:

Commodore Disk User Binder £4.95, including 10 sleeves. Order code **BDYU1**

Commodore Disk User Binder with 10 sleeves and 10 disks, £7.95 Order code **BDYU2**

10 sleeves for insertion in binder, £1.50. Order code **BDSS0**

20 sleeves for inclusion in binder, £1.75. Order code **BDSS2**

10 Commodore Disk User data disks, £5.95. Order code **BDG10**

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST. Please allow 28 days for delivery.



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			
			TOTAL



NEXT MONTH

NIGHTBREED

Clive Barker spills his guts in an exclusive interview about his new film and the game to follow.

LIFESTYLES 2000

On the spot report.

DESERT ISLAND DISKS

Tony Crowther is the first to contribute to our new regular spot.

Plus Role Playing Games, Play By Mail, Software in the Office...

Reasoning On The 128

*We continue with our expert system for the 128
computer*

By Paul Schofield

In the last installment of this series we introduced you to the idea of a simple expert system, from now on, we will look at the more sophisticated knowledge base which I introduced in that article. This system comprises of three programmes:

EDR - the source program editor
KBDEF - the knowledge base compiler
INQUIRE - the inference engine

Why three programs? The most important consideration is space available for data, which is increased by keeping the programs small. Speed of operation is also improved, as once the application has been debugged, it is only necessary to use the **INQUIRE** program which reads in the data in a compressed and preprocessed format. In this section, we will be concerned mainly with the editor, but first a quick overview of the whole system.

Defining a knowledge base

To create the knowledge base for the reason program, we first processed all the data that we had on the subject in question and arranged it into a tree structure. In the case of the very simple chemical analysis example, we ended up with a nice simple symmetric tree, which therefore took an efficient path to any of the solutions. If we were to extend this, we would have to start using tests to identify particular chemicals and so the tree would no longer be symmetric. This is similar to the car maintenance example, where

it was suggested that the first questions might be:

Is fuel tank is empty?
Is battery is flat?

In this case, we pick these as starting questions, because they are easy to check and the produce a simple solution.

In terms of our expert system however, they represent a fundamentally different approach. We pick a solution to the problem, namely, just petrol in fuel tank and ask a set of questions to support it. If the answers are affirmative we have an answer, otherwise we take the next possible solution and test that.

This is the approach that our new knowledge base system will adopt. It is probably the most widely used in this type of application and has a number of advantages. There are, however, attendant disadvantages, and so the system must endeavour to take account of these.

Incomplete data

One of the first rules one learns as a programmer is, garbage in means garbage out, and this is particularly true of this type of expert system. To define your knowledge base, it is necessary to define your targets (the possible solution) and the conditions which must be satisfied for a particular target to be the solution. The order in which you define the targets also implicitly defines the order in which they are evaluated. This is good, because at the outset you can start by just entering conditions, which you

know to be true about a target. The disadvantage is easily shown by an example. Suppose you write a knowledge base for animal identification. The first target is cat and the conditions are:

Has four legs?
Has a tail?

In use, the knowledge base will tell you that a wide variety of animals are cats. The real message is 'according to the supplied criteria a possible solution is a cat'. For the expert system to be useful it's not sufficient for it simply to stop at the first solution; it must provide to additional options:

Why - A justification of the solution.
Also - Continue to look for other solutions.

With the aid of these tools it is possible to identify sections of the knowledge base, which are incomplete and identify suitable conditions to apply. We will consider such problems in a little more detail when we consider the source definition language and the inference engine, but this should suffice as a general introduction to the system.

EDR - The source file editor

The obvious question is why not use the Basic editor? One reason of course is that the Basic editor produces tokenized program files, but in general it is more convenient to use either a general purpose text editor or one specific to the particular application. **EDR** is specific to this application, but

would not require much modification to make a general purpose text editor. To run the editor, simply enter: `run 'ED'`. The editor initializes various variables and then displays the main options menu. Most of these are self explanatory, so we will just consider option 1, create/edit RAL source file. When this option is selected, a help bar is displayed at the top of the screen showing the functions assigned to the function keys. The remainder of the screen is either blank (create) or displays the start of the currently loaded file. The files produced are sequential files, so no file numbers are needed, just enter the text you require. The following special function keys are also supported:

```
CURSOR : Move up/down/left/
KEYS    right one character
DEL     : Delete character left of
        cursor
INST    : Insert blank character at
        cursor
F1      : Insert blank line
F2      : Delete current line
RETURN  : Move cursor to start of
        next line
F6      : Delete remainder of line
        to right of cursor
F4      : Copy line to paste buffer
        and delete it
F3      : Paste saved line at cur
F5      : Move cur to top of file
F7      : Move cur to end of file
HOME    : Move cur to top left of
        screen
CTRL,U  : Move up 20 lines or to
        top of file
CTRL,D  : Move down 20 lines or
        to end of file
F8      : exit from edit mode
```

One automatic function of the editor is that when the bottom of the screen is reached, the text is scrolled up 10 lines.

Modifying The Editor

The functions supported by the editor are all that you are likely to need for entering knowledge base definitions. For more general use a few modifications are likely to be needed. New edit functions can be added very easily. Add a text for the key in question between line 10380 and line 10390 and insert the corresponding subroutine somewhere between line 2090 and line 4090.

Two additional modifications may also be required:

1) Currently the number of lines of text is written at the start of the disk files. If this is considered undesirable, the disk read/write routines must be changed to detect eof.

2) Commodore Basic has trouble with commas when writing text to disk files unless special precautions are taken. As the knowledge base definition language does not use commas, this has not been done.

We shall now look at the definition language RAL and the compiler program KINGEN.

Having experimented with the REASON program, and typing in the EDR Editor, we will now look at the knowledge base definition. Later, we will look at the compiler program, KINGEN. First of all though, we will look at the definition language. As these 3 programs together represent the package Reason Version 1, I have called the language, Reason Application Language (RAL).

Before going into details though, a quick look at some of the objectives. Although KINGEN is technically a compiler, I prefer to think of it as a simple preprocessor. It does not generate machine code, but simply takes a description of data structures in a readable format and translates them into a set of array variables for driving an inference engine. It was also considered important, that the compiler should be able to output the processed knowledge base on a different disk to the source file. This is administratively very convenient, but, if tedious disk swapping is to be avoided, requires that the entire source file is read before attempting to write the output file. This is sure to cause that large string storage requirements are provided and in order not to unduly restrict the size of applications the language becomes a little cryptic in places.

It is possible to change this approach, but I would recommend trying the existing system first, it's not really that difficult.

Structure of RAL

RAL is a block structured language like Pascal or C, but the syntax of the individual blocks is a bit more rigid.

Outside any of the blocks comment lines are permitted, so the basic structure of a RAL program is:

```
comments
RAL BLOCKS
comments
:
comments
RAL BLOCKS
comments
```

There are 3 types of RAL block-

HELP - Initial instructions displayed to user of base.

TARGETS - Define the set of target solutions.

CONDITIONS - Define the set of questions the user is asked.

RELATIONS - Define set of rules for evaluating targets.

IMPLIES - Defines how additional conditions may be inferred from a users response.

The structure of each block is as follows:

```
BLOCK-HEADER
Block statements
END[options]
```

The square brackets [] indicate optional elements.

Comments

As mentioned previously, comments may be included anywhere outside a block, they have the format:

```
[any text string]
```

It is important that the '`'`' is the first character on comment lines and these are simply ignored by the compiler. It would not be difficult to permit comments within blocks, but it would greatly slow down the compiler and is not very useful.

Help Section

The help section is optional, but should normally be included in all RAL definitions. It simply defines a number of lines of explanatory text that will be displayed to the user before any questions are asked. It may typically be used to display a title, take credit for creating the knowledge base and providing any necessary background information. It is not necessary to include any information on how to respond to reason questions as this is automatically output by the INQUIRE program. The format of the help section is:

HELP integer
help-text-lines
end[HELP]

where:

integer defines the number of lines of help information following.
help-text-lines are the help information in the format to be displayed.

Targets Section

The targets section defines the text strings describing the target solutions and is obviously mandatory. The format of the targets section is:

TARGETS integer
target-line
END [TARGETS]

where:

integer defines the number of entries to be reserved for targets.
target-line have the format:

[*index*] *target-text*

where:

index is an optional target number reference (1,2,3 etc.)
target-text is the text that will be displayed to describe the solution.

Notes

The purpose of *integer* is to define internal storage requirements and so must be the same as or greater than the number of following targets. The *index* is purely a comment, a bit like a BASIC line number. It is used in later sections to identify specific targets and is therefore useful in large knowledge bases. The first *index* is always 1 and they are allocated consecutively.

Conditions Section

The conditions section defines the questions which will be asked to the user and is obviously mandatory. The format of the conditions section is:

CONDITIONS integer
condition-line
END [CONDITIONS]

where:

integer defines the number of entries to be reserved for conditions.
condition-line have the format:

[*index*] *condition-text*

where:

index is an optional condition number

reference (1,2,3 etc.)

condition-text is the text of the questions displayed to the user.

integer and *index* are treated as for targets.

Relations Section

The relations section defines the rules for evaluating targets according to the answers to the condition questions. The results of the various conditions may be either:

irrelevant?

(can be true for target to be a solution) or *relevant* (must be false for target to be a solution).

By default, all conditions are considered irrelevant unless mentioned explicitly within a rule in the relations section. The format of the relations section is:

RELATIONS
relation-rule
END [RELATIONS]

where:

relation-rule have the format: *target-index operator condition-index-list*

where:

operator is '+' following conditions must be true, or '-' following conditions must be false.

condition-index-list is a number of condition indices separated by spaces.

Implies Section

The implies section is optional. It defines the results of further conditions, which may be determined from the result of the current condition. The object of this section is to avoid asking questions whose results should already be known and to a lesser extent, avoid investigating impossible targets as quickly as possible. For example, if during the investigation of one target the question is asked "IS OBJECT BLUE?" and the answer is true, it is quite obvious that a later question "IS OBJECT RED?" must have the answer false. The implies section allows upto three additional results to be implied for both a true or false answer to a particular condition. Its format is:

IMPLIES
implication-rule
END [IMPLIES]

where the format of implication rule is:

result-condition *implied-condition*
 [*implied-condition*] [*implied-condition*]

and the format of both *result-condition* and *implied condition* is:

"*i*" *condition-index*

As with relation rules + = true and - = false.

If that sounds a little confusing, then have a look at the example program in figure 1. Yes we are back with our mixed up chemical bottles again! With only 3 targets and 4 conditions this is very simple and it is easy to see how the relations and implies sections are used.

It is always best to work through the targets in the order they are defined as this is also the order in which they will be tested. In this case we check firstly for a strong acid. For this to be the case, blue litmus must be turned strongly red. This is a rather subjective question, so the simpler condition blue Litmus Turned Red is considered first.

In the relations section, we use 3 rules defined, the first saying that conditions 1 and 2 must be true and a second requiring conditions 3 and 4 to be false. It is quite obvious in fact that if 1 and 2 are true that 3 and 4 will be false, but this is defined in the implies section instead. The reason for doing it this way is that at some time the future it could be decided to reorganise to test targets in a different order. This becomes difficult if the rules of the relations section are incomplete.

KBBGEN program

The structure of this program is quite simple. The main program reads in the RAL source file line by line and looks for the sections in the expected order. Within each section the lines are validated according to the defined syntax. Any errors encountered are reported and if the error is fatal the processing aborts. Standard subroutines are used for skipping spaces and finding and extracting various elements in the source lines. It should not be that difficult to make changes to the program if desired.

The most obvious change to consider is to open the output file on the same disk as the start of the program. As the format of the help, targets and condition sections is very simple, these can be written directly to the output file after being validated so the large string arrays are avoided. This saving


```

3810 GOSUB 180
1000 GOTO 1
3820 IF LEFT$(B1,1)="" THEN G=4
3830 L4=RIGHT$(B1,LEN(B1)-1)-B05
G=180
3840 B0
3850 GOSUB 200:CALLBACK:PRINT
GOTO 1
3870 GOSUB 180
3880 LOOP UNTIL LEN(L4)=0
3890 LOOP UNTIL B=1 OR L4=0
1900 IF B=1 GOTO 1800
3910 PRINT "NO MATCHING * UNEXPECTED
END OF RELATIONS, INCOMPLETE
*":GOTO 2100
1920 FOR IMPLIED SECTION TO OPEN
1930 = NO ERROR IF POSSIBLE
1940 GOTO 1810
1950 INPUT L4:G=L4
1960 LOOP UNTIL LEFT$(B1,1)=""
OR L4=0
1980 IF L4="" THEN GOTO
1970 IF LEFT$(L4,1)="" THEN
GOTO 2000
1980 PRINT "AN UNSPEC ERROR * IN
FILE EXPECTED IN L4":GOTO 1
19
1990 PRINT "THE REMAINING LINES I
SEARCHED. *":GOTO 2100
2000 FOR PROBABLY APPLICATION LINES
DO = THE 3 / TRUE OR FALSE VALUE
0
2010 L=1:G1=1:G2=1:G3=1
DO UNTIL J=0 TO 1:G1=1:J:G2=1:
NEXT:PRINT:G=2:PRINT "REMAINING IMP
LIES SECTION AT L4":GOTO 1800
2020 B0
2030 INPUT L4:G=L4
2040 IF LEFT$(B1,1)="" THEN G=4
2050 GOSUB 180:G=4:GOSUB 200:G=10-
G:G=10
2060 GOSUB 200
2070 DO UNTIL LEN(L4)=0
2080 GOSUB 180:GOSUB 200:G=10-
G:G=10:GOTO 2100
2090 GOSUB 200:PRINT:CALLBACK:
GOTO 1810
2100 IF LEFT$(L4,1)="" THEN PRINT
"NO MATCHING * UNEXPECTED
END OF RELATIONS, INCOMPLETE
*":GOTO 2100
2110 FOR IMPLIED FOR CONDITION "G1."
G1=1:IMPOSED "0"
2120 LOOP
2130 LOOP UNTIL B=1 OR L4=0
2140 IF B=1 GOTO 2170
2150 PRINT "NO MATCHING * UNEXPECTED
END OF RELATIONS, INCOMPLETE
*":GOTO 2100
2160 FOR CLOSE SEARCH FILE
2170 GOSUB 180:PRINT:G=4:G=10-
G:G=10:PRINT "START"
2180 FOR IMPROVE FILE WORD, WRITE
ANALYZE WORD FILE
2190 PRINT:G=4:G=10:G=10-
G:G=10:PRINT "START"
2200 PRINT:G=4:G=10:G=10-
G:G=10:PRINT "START"
2210 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2220 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2230 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2240 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2250 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2260 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2270 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2280 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2290 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2300 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2310 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2320 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2330 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2340 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2350 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2360 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2370 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2380 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2390 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2400 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2410 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2420 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2430 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2440 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2450 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2460 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2470 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2480 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2490 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2500 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2510 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2520 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2530 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2540 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2550 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2560 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2570 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2580 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2590 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2600 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2610 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2620 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2630 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2640 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2650 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2660 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2670 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2680 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2690 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2700 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2710 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2720 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2730 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2740 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2750 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2760 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2770 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2780 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2790 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2800 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2810 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2820 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2830 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2840 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2850 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2860 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2870 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2880 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2890 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2900 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2910 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2920 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2930 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2940 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2950 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2960 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2970 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2980 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
2990 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3000 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3010 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3020 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3030 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3040 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3050 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3060 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3070 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3080 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3090 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3100 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3110 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3120 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3130 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3140 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3150 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3160 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3170 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3180 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3190 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3200 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3210 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3220 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3230 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3240 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3250 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3260 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3270 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3280 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3290 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3300 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3310 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3320 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3330 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3340 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3350 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3360 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3370 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3380 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3390 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3400 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3410 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3420 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3430 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3440 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3450 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3460 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3470 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3480 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3490 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3500 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3510 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3520 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3530 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3540 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3550 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3560 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3570 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3580 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3590 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3600 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3610 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3620 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3630 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3640 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3650 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3660 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3670 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3680 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3690 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3700 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3710 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3720 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3730 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3740 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3750 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3760 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3770 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3780 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3790 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3800 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3810 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3820 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3830 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3840 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3850 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3860 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3870 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3880 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3890 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3900 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3910 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3920 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3930 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3940 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3950 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3960 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3970 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3980 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
3990 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"
4000 PRINT "ANALYZE WORD FILE
*":G=4:G=10:G=10-
G:G=10:PRINT "START"

```



DON'T LET THE AUSSIE BLUES GET TO YOU

FOOLSBUSTER84 IS THE ONLY POOLS PROGRAM AT HOME "DOWN UNDER", AS WELL AS IN EUROPE. Every week-and it guides you through the tough terrain of the Aussie leagues to successful predictions over and over, again and again. It's a very successful conversion from the ATARI ST program. Go on, don't hesitate, join the ATARI winners. It works for any game, cup or league, either British or Australian or European. It has a massive database, it's user friendly, it gives you a choice at the final completion of the coupon. Perfect for pools, ideal for fixed odds. Are you superstitious?

Do you believe in the power of the stars? If you answer 'yes' to any of the above 4 questions then FOOLSBUSTER84 is for you. Its MATCH PREDICTION function will happily accommodate you. Do you believe in the power of the numbers? If the answer is 'yes', then FOOLSBUSTER84 has something for you as well. Its "PREDICTION BY ELIMINATION" function will fulfil your exact requirements. And for the rest of us mortals, it will manipulate all the required information and it will give you different forecasts: a normal results forecast, a best horses a best draw and a best away forecast. Are you into probabilities, percentages and bar graph representations? FOOLSBUSTER84 will give them to you. In fact it has everything that you need for a successful forecast. It is truly the ultimate and the state-of-the-art pools program. So hurry order your copy today and win the pools. Succession.

FOOLSBUSTER84 COSTS ONLY £40!

SPECIAL OFFER: THE FIRST 24 ORDERS WILL RECEIVE A BOY GAME FREE. SO HURRY, DON'T MISS YOUR CHANCE TO BECOME A POOLS WINNER. WE ACCEPT ACCESS AND VISA CARDS.

MAKE CHECKS / P.M PAYABLE TO 'APOLONIA SOFTWARE' AND SEND THEM TO 'APOLONIA SOFTWARE UNIT 11 PARK HOUSE, 148 BATTERSEA PARK ROAD, LONDON, SW11 4BB' PLEASE STATE IF YOU WANT DISK OR TAPE.

ORDER: 01 / 738-9400 FAX: 06 / 822-1261. ALL OFFERS ARE SUBJECT TO STOCK AVAILABILITY.

APOLONIA SOFTWARE THE COMPANY THAT CARES

Dos 6.1

We put everything you have learned in this series to some practical use

By Fergal Moane

This final article of the series presents a complete disk operating system, which puts into practice some of the theory already learned. If you are interested, disassembly of the code from SCDS-B onwards should be useful.

Dos 6.1 - The Utility

This utility is in the same mould as DOS 5.1 on the demo disk. DOS 5.1 is the most used Commodore utility in America. This system outstrips it in a number of ways:

- 1) DOS 6.1 actually techniques it's new commands. Tokens are one-byte values representing commands, and are used by ordinary Basic for all commands. This ensures compatibility, space economy, and no need for a prefixing character.
- 2) The above feature means that DOS 6.1 commands can be used in program mode, unlike most normal extensions.
- 3) DOS 6.1 provides more comprehensive commands in an easily understood form.
- 4) You are left around 3K free in the 4K block of memory from \$C000-\$D000.
- 5) Loading is under ten seconds, providing you save out the code separately, and the program is easily copied to your own disks.

Users Guide

DISK ERROR STATUS

DERR displays the status of the disk drive at the current position on the screen. See your manual for a full explanation of error messages.
Syntax: **DERR**

DOSS COMMAND

DOSS sends a command to the disk drive, opening up a range of around 50 disk commands. Again, see a manual for details of standard disk commands. If **DOSS** is used without a

command, the error status will be returned.

Syntax: **DOSS**"command"

DIRECTORY

Displays the directory of drive G or I (Usually drive G) without loading it into memory.

Syntax: **DIR**, drive number

MEMORY SAVE

Saves memory from start address to end address to the specified device. This is very useful with graphics or machine code data.

Syntax: **MSAVE**, start, end, "filename", dev

DEFAULT TO DISK

This sets device 0 as the default device for all **SAVE** and **LOAD** operations. **LOAD**'tokens will now have the same effect as **LOAD**"filename",0
Syntax: **DISK**

DEFAULT TO TAPE

Tape is now the default device for **SAVING** and **LOADING**.
Syntax: **TAPE**

NORMAL DEVICE NUMBERS

This returns to the normal Commodore device numbering system. Use this if you need to load with **ALL**
Syntax: **NORMAL**

LIST BASIC PROGRAM

DLIST allows you to list a Basic program directly off disk without loading it into memory. This is useful in checking a program, or grabbing lines from another program. **RUN**/**STOP** will stop the listing at any time. If you try to list a non-basic program, you will get the expected garbage. Press **RUN/STOP** **RESTORE** followed by **CLOSE** to recover.
Syntax: **DLIST**"filename"

LEAVE DOS 6.1

Leaves the DOS 6.1 system without disturbing Basic or any memory.

SY549152:SY549676 will return you to the DOS 6.1 system.

Syntax: **QUIT**

MEMORY USAGE

A memory map of the system could be useful:

C000 - C120	Program for adding new tokens
C130 - C180	Table of jump addresses for new commands
C200 - C53A	New Dos routines
D53B - CFFF	Free RAM for user programs

Some low memory and into page locations are also altered during the command processing, but this has no effect on Basic and the area around the tape buffer is untouched.

Loading DOS 6.1

Type in and save the Basic program. When run, you will be presented with a screen of instructions. The machine code is being **FORKed** into memory at this stage. On completion of the program **NEWS** itself and you are into the DOS system. Obviously, you will not want to keep having to load the basic loader each time you wish to use it. Therefore, I suggest you use the **MSAVE** command to save out the relevant two portions of memory.

USING DOS 6.1 IN YOUR OWN PROGRAM

As the commands can be used in your own programs, it would be inconvenient to have to load DOS 6.1 in every time, especially if your program is seen by others. Follow the following procedure to include it in your program.

Load the two machine code files that you have saved at the start of your program. Immediately afterwards **SY549152:SY549676** to initialize the commands.

Program Notes

When starting programs in the modified environment, **CH** Basic insists that you now have to enter colons after a **THEN** statement. There seems to be no way to override this, but it does not effect programs saved in the normal mode. These will work normally.

EG- **DO GET A\$: IF A\$: "" THEN: GOTO 10**

30 IFAS-77 THEN DERR

Also note that some commands will return to direct mode if used in

a program. A bit of FORKING of the keyboard buffer could solve this. Unfortunately, this is unavoidable.

I hope that this series of articles has been of use to you, and makes your disk drive easier to handle.

PROGRAM CODE 8.1

```

20 10 REM *****
21 11 REM IF YOU WANT TO SAVE
22 12 REM TIME YOU CAN SAVE
23 13 REM THE PROGRAM CODE ON
24 14 REM THE USE OF A MONITOR
25 15 REM PROGRAM.
26 16 REM *****
27 20 PRINT"OUR SPEED-TESTING
28 21 OF DOS 8.1. PLEASE WAIT".
29 30 PRINT"OK, PLEASE WAIT".
30 40 PRINT"OK, YELLOW COPY".
31 50 PRINT"OK, YELLOW COPY".
32 60 PRINT"OK, YELLOW COPY".
33 70 PRINT"OK, YELLOW COPY".
34 80 PRINT"OK, YELLOW COPY".
35 90 PRINT"OK, YELLOW COPY".
36 100 PRINT"OK, YELLOW COPY".
37 110 PRINT"OK, YELLOW COPY".
38 120 PRINT"OK, YELLOW COPY".
39 130 PRINT"OK, YELLOW COPY".
40 140 PRINT"OK, YELLOW COPY".
41 150 PRINT"OK, YELLOW COPY".
42 160 PRINT"OK, YELLOW COPY".
43 170 PRINT"OK, YELLOW COPY".
44 180 PRINT"OK, YELLOW COPY".
45 190 PRINT"OK, YELLOW COPY".
46 200 PRINT"OK, YELLOW COPY".
47 210 PRINT"OK, YELLOW COPY".
48 220 PRINT"OK, YELLOW COPY".
49 230 PRINT"OK, YELLOW COPY".
50 240 PRINT"OK, YELLOW COPY".
51 250 PRINT"OK, YELLOW COPY".
52 260 PRINT"OK, YELLOW COPY".
53 270 PRINT"OK, YELLOW COPY".
54 280 PRINT"OK, YELLOW COPY".
55 290 PRINT"OK, YELLOW COPY".
56 300 PRINT"OK, YELLOW COPY".
57 310 PRINT"OK, YELLOW COPY".
58 320 PRINT"OK, YELLOW COPY".
59 330 PRINT"OK, YELLOW COPY".
60 340 PRINT"OK, YELLOW COPY".
61 350 PRINT"OK, YELLOW COPY".
62 360 PRINT"OK, YELLOW COPY".
63 370 PRINT"OK, YELLOW COPY".
64 380 PRINT"OK, YELLOW COPY".
65 390 PRINT"OK, YELLOW COPY".
66 400 PRINT"OK, YELLOW COPY".
67 410 PRINT"OK, YELLOW COPY".
68 420 PRINT"OK, YELLOW COPY".
69 430 PRINT"OK, YELLOW COPY".
70 440 PRINT"OK, YELLOW COPY".
71 450 PRINT"OK, YELLOW COPY".
72 460 PRINT"OK, YELLOW COPY".
73 470 PRINT"OK, YELLOW COPY".
74 480 PRINT"OK, YELLOW COPY".
75 490 PRINT"OK, YELLOW COPY".
76 500 PRINT"OK, YELLOW COPY".
77 510 PRINT"OK, YELLOW COPY".
78 520 PRINT"OK, YELLOW COPY".
79 530 PRINT"OK, YELLOW COPY".
80 540 PRINT"OK, YELLOW COPY".
81 550 PRINT"OK, YELLOW COPY".
82 560 PRINT"OK, YELLOW COPY".
83 570 PRINT"OK, YELLOW COPY".
84 580 PRINT"OK, YELLOW COPY".
85 590 PRINT"OK, YELLOW COPY".
86 600 PRINT"OK, YELLOW COPY".
87 610 PRINT"OK, YELLOW COPY".
88 620 PRINT"OK, YELLOW COPY".
89 630 PRINT"OK, YELLOW COPY".
90 640 PRINT"OK, YELLOW COPY".
91 650 PRINT"OK, YELLOW COPY".
92 660 PRINT"OK, YELLOW COPY".
93 670 PRINT"OK, YELLOW COPY".
94 680 PRINT"OK, YELLOW COPY".
95 690 PRINT"OK, YELLOW COPY".
96 700 PRINT"OK, YELLOW COPY".
97 710 PRINT"OK, YELLOW COPY".
98 720 PRINT"OK, YELLOW COPY".
99 730 PRINT"OK, YELLOW COPY".
100 740 PRINT"OK, YELLOW COPY".
101 750 PRINT"OK, YELLOW COPY".
102 760 PRINT"OK, YELLOW COPY".
103 770 PRINT"OK, YELLOW COPY".
104 780 PRINT"OK, YELLOW COPY".
105 790 PRINT"OK, YELLOW COPY".
106 800 PRINT"OK, YELLOW COPY".
107 810 PRINT"OK, YELLOW COPY".
108 820 PRINT"OK, YELLOW COPY".
109 830 PRINT"OK, YELLOW COPY".
110 840 PRINT"OK, YELLOW COPY".
111 850 PRINT"OK, YELLOW COPY".
112 860 PRINT"OK, YELLOW COPY".
113 870 PRINT"OK, YELLOW COPY".
114 880 PRINT"OK, YELLOW COPY".
115 890 PRINT"OK, YELLOW COPY".
116 900 PRINT"OK, YELLOW COPY".
117 910 PRINT"OK, YELLOW COPY".
118 920 PRINT"OK, YELLOW COPY".
119 930 PRINT"OK, YELLOW COPY".
120 940 PRINT"OK, YELLOW COPY".
121 950 PRINT"OK, YELLOW COPY".
122 960 PRINT"OK, YELLOW COPY".
123 970 PRINT"OK, YELLOW COPY".
124 980 PRINT"OK, YELLOW COPY".
125 990 PRINT"OK, YELLOW COPY".
126 1000 PRINT"OK, YELLOW COPY".

```

```

80 48000 POKE 1,4:1-1:0810 90
81 000
82 48100 DATA 100,21,100,100,70
83 0,4,3,100
84 48150 DATA 5,3,100,210,100,1
85 00,100,0
86 48180 DATA 3,100,7,3,000,000,
87 100,100
88 48190 DATA 4,100,10,000,0,0,
89 10,10
90 48210 DATA 100,000,000,00,0,0
91 0,000,00,000
92 48240 DATA 000,001,001,00,00
93 0,00,100,0
94 48270 DATA 001,00,000,00,00,
95 10,10,10
96 48300 DATA 001,00,000,1,000,
97 100,000,00
98 48330 DATA 000,00,100,1,000,
99 100,000,000
100 48360 DATA 000,00,100,1,000,
101 100,000,000
102 48390 DATA 000,00,100,1,000,
103 100,000,000
104 48420 DATA 000,00,100,1,000,
105 100,000,000
106 48450 DATA 000,00,100,1,000,
107 100,000,000
108 48480 DATA 000,00,100,1,000,
109 100,000,000
110 48510 DATA 000,00,100,1,000,
111 100,000,000
112 48540 DATA 000,00,100,1,000,
113 100,000,000
114 48570 DATA 000,00,100,1,000,
115 100,000,000
116 48600 DATA 000,00,100,1,000,
117 100,000,000
118 48630 DATA 000,00,100,1,000,
119 100,000,000
120 48660 DATA 000,00,100,1,000,
121 100,000,000
122 48690 DATA 000,00,100,1,000,
123 100,000,000
124 48720 DATA 000,00,100,1,000,
125 100,000,000
126 48750 DATA 000,00,100,1,000,
127 100,000,000
128 48780 DATA 000,00,100,1,000,
129 100,000,000
130 48810 DATA 000,00,100,1,000,
131 100,000,000
132 48840 DATA 000,00,100,1,000,
133 100,000,000
133 48870 DATA 000,00,100,1,000,
134 100,000,000
134 48900 DATA 000,00,100,1,000,
135 100,000,000
135 48930 DATA 000,00,100,1,000,
136 100,000,000
136 48960 DATA 000,00,100,1,000,
137 100,000,000
137 48990 DATA 000,00,100,1,000,
138 100,000,000
138 49020 DATA 000,00,100,1,000,
139 100,000,000
139 49050 DATA 000,00,100,1,000,
140 100,000,000
140 49080 DATA 000,00,100,1,000,
141 100,000,000
141 49110 DATA 000,00,100,1,000,
142 100,000,000
142 49140 DATA 000,00,100,1,000,
143 100,000,000
143 49170 DATA 000,00,100,1,000,
144 100,000,000
144 49200 DATA 000,00,100,1,000,
145 100,000,000
145 49230 DATA 000,00,100,1,000,
146 100,000,000
146 49260 DATA 000,00,100,1,000,
147 100,000,000
147 49290 DATA 000,00,100,1,000,
148 100,000,000
148 49320 DATA 000,00,100,1,000,
149 100,000,000
149 49350 DATA 000,00,100,1,000,
150 100,000,000
150 49380 DATA 000,00,100,1,000,
151 100,000,000
151 49410 DATA 000,00,100,1,000,
152 100,000,000
152 49440 DATA 000,00,100,1,000,
153 100,000,000
153 49470 DATA 000,00,100,1,000,
154 100,000,000
154 49500 DATA 000,00,100,1,000,
155 100,000,000
155 49530 DATA 000,00,100,1,000,
156 100,000,000
156 49560 DATA 000,00,100,1,000,
157 100,000,000
157 49590 DATA 000,00,100,1,000,
158 100,000,000
158 49620 DATA 000,00,100,1,000,
159 100,000,000
159 49650 DATA 000,00,100,1,000,
160 100,000,000
160 49680 DATA 000,00,100,1,000,
161 100,000,000
161 49710 DATA 000,00,100,1,000,
162 100,000,000
162 49740 DATA 000,00,100,1,000,
163 100,000,000
163 49770 DATA 000,00,100,1,000,
164 100,000,000
164 49800 DATA 000,00,100,1,000,
165 100,000,000
165 49830 DATA 000,00,100,1,000,
166 100,000,000
166 49860 DATA 000,00,100,1,000,
167 100,000,000
167 49890 DATA 000,00,100,1,000,
168 100,000,000
168 49920 DATA 000,00,100,1,000,
169 100,000,000
169 49950 DATA 000,00,100,1,000,
170 100,000,000

```

```

87 49980 DATA 000,00,100,1,000,
88 50010 DATA 000,00,100,1,000,
89 50040 DATA 000,00,100,1,000,
90 50070 DATA 000,00,100,1,000,
91 50100 DATA 000,00,100,1,000,
92 50130 DATA 000,00,100,1,000,
93 50160 DATA 000,00,100,1,000,
94 50190 DATA 000,00,100,1,000,
95 50220 DATA 000,00,100,1,000,
96 50250 DATA 000,00,100,1,000,
97 50280 DATA 000,00,100,1,000,
98 50310 DATA 000,00,100,1,000,
99 50340 DATA 000,00,100,1,000,
100 50370 DATA 000,00,100,1,000,
101 50400 DATA 000,00,100,1,000,
102 50430 DATA 000,00,100,1,000,
103 50460 DATA 000,00,100,1,000,
104 50490 DATA 000,00,100,1,000,
105 50520 DATA 000,00,100,1,000,
106 50550 DATA 000,00,100,1,000,
107 50580 DATA 000,00,100,1,000,
108 50610 DATA 000,00,100,1,000,
109 50640 DATA 000,00,100,1,000,
110 50670 DATA 000,00,100,1,000,
111 50700 DATA 000,00,100,1,000,
112 50730 DATA 000,00,100,1,000,
113 50760 DATA 000,00,100,1,000,
114 50790 DATA 000,00,100,1,000,
115 50820 DATA 000,00,100,1,000,
116 50850 DATA 000,00,100,1,000,
117 50880 DATA 000,00,100,1,000,
118 50910 DATA 000,00,100,1,000,
119 50940 DATA 000,00,100,1,000,
120 50970 DATA 000,00,100,1,000,
121 51000 DATA 000,00,100,1,000,
122 51030 DATA 000,00,100,1,000,
123 51060 DATA 000,00,100,1,000,
124 51090 DATA 000,00,100,1,000,
125 51120 DATA 000,00,100,1,000,
126 51150 DATA 000,00,100,1,000,
127 51180 DATA 000,00,100,1,000,
128 51210 DATA 000,00,100,1,000,
129 51240 DATA 000,00,100,1,000,
130 51270 DATA 000,00,100,1,000,
131 51300 DATA 000,00,100,1,000,
132 51330 DATA 000,00,100,1,000,
133 51360 DATA 000,00,100,1,000,
134 51390 DATA 000,00,100,1,000,
135 51420 DATA 000,00,100,1,000,
136 51450 DATA 000,00,100,1,000,
137 51480 DATA 000,00,100,1,000,
138 51510 DATA 000,00,100,1,000,
139 51540 DATA 000,00,100,1,000,
140 51570 DATA 000,00,100,1,000,
141 51600 DATA 000,00,100,1,000,
142 51630 DATA 000,00,100,1,000,
143 51660 DATA 000,00,100,1,000,
144 51690 DATA 000,00,100,1,000,
145 51720 DATA 000,00,100,1,000,
146 51750 DATA 000,00,100,1,000,
147 51780 DATA 000,00,100,1,000,
148 51810 DATA 000,00,100,1,000,
149 51840 DATA 000,00,100,1,000,
150 51870 DATA 000,00,100,1,000,
151 51900 DATA 000,00,100,1,000,
152 51930 DATA 000,00,100,1,000,
153 51960 DATA 000,00,100,1,000,
154 51990 DATA 000,00,100,1,000,
155 52020 DATA 000,00,100,1,000,
156 52050 DATA 000,00,100,1,000,
157 52080 DATA 000,00,100,1,000,
158 52110 DATA 000,00,100,1,000,
159 52140 DATA 000,00,100,1,000,
160 52170 DATA 000,00,100,1,000,
161 52200 DATA 000,00,100,1,000,
162 52230 DATA 000,00,100,1,000,
163 52260 DATA 000,00,100,1,000,
164 52290 DATA 000,00,100,1,000,
165 52320 DATA 000,00,100,1,000,
166 52350 DATA 000,00,100,1,000,
167 52380 DATA 000,00,100,1,000,
168 52410 DATA 000,00,100,1,000,
169 52440 DATA 000,00,100,1,000,
170 52470 DATA 000,00,100,1,000,

```

Dos 6.1

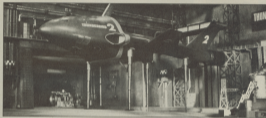
10	70720	DW14	107, 107, 25, 105, 23
11	70730	DW14	0, 105, 100, 0
12	70740	DW14	100, 0, 107, 0, 0, 10
13	70750	DW14	07, 173, 107, 107, 13
14	70760	DW14	3, 00, 0, 0, 0
15	70770	DW14	104, 0, 100, 0, 0, 0, 0
16	70780	DW14	0, 0, 0, 0, 0, 101, 0,
17	70790	DW14	0, 0, 0
18	707A0	DW14	3, 70, 000, 104, 100,
19	707B0	DW14	0, 103, 100
20	707C0	DW14	07, 170, 107, 100, 13
21	707D0	DW14	100, 0, 0, 0, 0, 0, 0, 0
22	707E0	DW14	100, 0, 0, 0, 0, 0, 0, 0
23	707F0	DW14	0, 0, 0, 0, 0, 0, 0, 0
24	70800	DW14	008, 7, 104, 0, 0, 10,
25	70810	DW14	000, 70, 100
26	70820	DW14	104, 104, 100, 13, 13
27	70830	DW14	020, 000, 170, 0, 0, 0
28	70840	DW14	000, 000, 100, 0,
29	70850	DW14	10, 100, 0, 0
30	70860	DW14	008, 7, 104, 0, 0, 10,
31	70870	DW14	000, 70, 100
32	70880	DW14	104, 104, 100, 13, 13
33	70890	DW14	020, 000, 170, 0, 0, 0
34	708A0	DW14	000, 000, 100, 0,
35	708B0	DW14	10, 100, 0, 0
36	708C0	DW14	008, 7, 104, 0, 0, 10,
37	708D0	DW14	000, 70, 100
38	708E0	DW14	104, 104, 100, 13, 13
39	708F0	DW14	020, 000, 170, 0, 0, 0
40	70900	DW14	000, 000, 100, 0,
41	70910	DW14	10, 100, 0, 0
42	70920	DW14	008, 7, 104, 0, 0, 10,
43	70930	DW14	000, 70, 100
44	70940	DW14	104, 104, 100, 13, 13
45	70950	DW14	020, 000, 170, 0, 0, 0
46	70960	DW14	000, 000, 100, 0,
47	70970	DW14	10, 100, 0, 0
48	70980	DW14	008, 7, 104, 0, 0, 10,
49	70990	DW14	000, 70, 100
50	709A0	DW14	104, 104, 100, 13, 13
51	709B0	DW14	020, 000, 170, 0, 0, 0
52	709C0	DW14	000, 000, 100, 0,
53	709D0	DW14	10, 100, 0, 0
54	709E0	DW14	008, 7, 104, 0, 0, 10,
55	709F0	DW14	000, 70, 100
56	70A00	DW14	104, 104, 100, 13, 13
57	70A10	DW14	020, 000, 170, 0, 0, 0
58	70A20	DW14	000, 000, 100, 0,
59	70A30	DW14	10, 100, 0, 0
60	70A40	DW14	008, 7, 104, 0, 0, 10,
61	70A50	DW14	000, 70, 100
62	70A60	DW14	104, 104, 100, 13, 13
63	70A70	DW14	020, 000, 170, 0, 0, 0
64	70A80	DW14	000, 000, 100, 0,
65	70A90	DW14	10, 100, 0, 0
66	70AA0	DW14	008, 7, 104, 0, 0, 10,
67	70AB0	DW14	000, 70, 100
68	70AC0	DW14	104, 104, 100, 13, 13
69	70AD0	DW14	020, 000, 170, 0, 0, 0
70	70AE0	DW14	000, 000, 100, 0,
71	70AF0	DW14	10, 100, 0, 0
72	70B00	DW14	008, 7, 104, 0, 0, 10,
73	70B10	DW14	000, 70, 100
74	70B20	DW14	104, 104, 100, 13, 13
75	70B30	DW14	020, 000, 170, 0, 0, 0
76	70B40	DW14	000, 000, 100, 0,
77	70B50	DW14	10, 100, 0, 0
78	70B60	DW14	008, 7, 104, 0, 0, 10,
79	70B70	DW14	000, 70, 100
80	70B80	DW14	104, 104, 100, 13, 13
81	70B90	DW14	020, 000, 170, 0, 0, 0
82	70BA0	DW14	000, 000, 100, 0,
83	70BB0	DW14	10, 100, 0, 0
84	70BC0	DW14	008, 7, 104, 0, 0, 10,
85	70BD0	DW14	000, 70, 100
86	70BE0	DW14	104, 104, 100, 13, 13
87	70BF0	DW14	020, 000, 170, 0, 0, 0
88	70C00	DW14	000, 000, 100, 0,
89	70C10	DW14	10, 100, 0, 0
90	70C20	DW14	008, 7, 104, 0, 0, 10,
91	70C30	DW14	000, 70, 100
92	70C40	DW14	104, 104, 100, 13, 13
93	70C50	DW14	020, 000, 170, 0, 0, 0
94	70C60	DW14	000, 000, 100, 0,
95	70C70	DW14	10, 100, 0, 0
96	70C80	DW14	008, 7, 104, 0, 0, 10,
97	70C90	DW14	000, 70, 100
98	70CA0	DW14	104, 104, 100, 13, 13
99	70CB0	DW14	020, 000, 170, 0, 0, 0
100	70CC0	DW14	000, 000, 100, 0,
101	70CD0	DW14	10, 100, 0, 0
102	70CE0	DW14	008, 7, 104, 0, 0, 10,
103	70CF0	DW14	000, 70, 100
104	70D00	DW14	104, 104, 100, 13, 13
105	70D10	DW14	020, 000, 170, 0, 0, 0
106	70D20	DW14	000, 000, 100, 0,
107	70D30	DW14	10, 100, 0, 0
108	70D40	DW14	008, 7, 104, 0, 0, 10,
109	70D50	DW14	000, 70, 100
110	70D60	DW14	104, 104, 100, 13, 13
111	70D70	DW14	020, 000, 170, 0, 0, 0
112	70D80	DW14	000, 000, 100, 0,
113	70D90	DW14	10, 100, 0, 0
114	70DA0	DW14	008, 7, 104, 0, 0, 10,
115	70DB0	DW14	000, 70, 100
116	70DC0	DW14	104, 104, 100, 13, 13
117	70DD0	DW14	020, 000, 170, 0, 0, 0
118	70DE0	DW14	000, 000, 100, 0,
119	70DF0	DW14	10, 100, 0, 0
120	70E00	DW14	008, 7, 104, 0, 0, 10,
121	70E10	DW14	000, 70, 100
122	70E20	DW14	104, 104, 100, 13, 13
123	70E30	DW14	020, 000, 170, 0, 0, 0
124	70E40	DW14	000, 000, 100, 0,
125	70E50	DW14	10, 100, 0, 0
126	70E60	DW14	008, 7, 104, 0, 0, 10,
127	70E70	DW14	000, 70, 100
128	70E80	DW14	104, 104, 100, 13, 13
129	70E90	DW14	020, 000, 170, 0, 0, 0
130	70EA0	DW14	000, 000, 100, 0,
131	70EB0	DW14	10, 100, 0, 0
132	70EC0	DW14	008, 7, 104, 0, 0, 10,
133	70ED0	DW14	000, 70, 100
134	70EE0	DW14	104, 104, 100, 13, 13
135	70EF0	DW14	020, 000, 170, 0, 0, 0
136	70F00	DW14	000, 000, 100, 0,
137	70F10	DW14	10, 100, 0, 0
138	70F20	DW14	008, 7, 104, 0, 0, 10,
139	70F30	DW14	000, 70, 100
140	70F40	DW14	104, 104, 100, 13, 13
141	70F50	DW14	020, 000, 170, 0, 0, 0
142	70F60	DW14	000, 000, 100, 0,
143	70F70	DW14	10, 100, 0, 0
144	70F80	DW14	008, 7, 104, 0, 0, 10,
145	70F90	DW14	000, 70, 100
146	70FA0	DW14	104, 104, 100, 13, 13
147	70FB0	DW14	020, 000, 170, 0, 0, 0
148	70FC0	DW14	000, 000, 100, 0,
149	70FD0	DW14	10, 100, 0, 0
150	70FE0	DW14	008, 7, 104, 0, 0, 10,
151	70FF0	DW14	000, 70, 100

C O M P E T I T I O N

ENTER



No Strings Attached



Win wonderful prizes, care of Grandlam, in our easy-to-enter Thunderbirds Competition.

Five first prize winners - drawn after the closing date - will receive a Thunderbirds video and the game for Amiga or C64. Pens and posters are available for sale

reason-up.

All you have to do is write the correct answers to the following questions and send them, on the back of a postcard or sealed envelope, to:

Thunderbirds Compo, Your Commodore, Argus House, Boundary Way, Hemel Hempstead, HP1 3ST.

1. What is Brain's real name?

- a) Horatio Haddockler
- b) Bartholomew Baker
- c) Kathleen Cooke

2. What does FAB stand for?

- a) Fantastic Automatic Breakthrough
- b) Nothing
- c) Thunderbirds Are Brill

3. What is Parker's nickname?

- a) Nobby
- b) Shifty
- c) Buzzer

Closing date: 31st August 1989.

The Rules

Entries will not be accepted from employees of Argus Specialist Publications, Grandlam, or Trigon. This

restriction also applies to employees' families and guests of the companies.

The entry restrictions form part of the rules and the editor's decision

is final. No correspondence will be entered into. In the event of a postal strike, we reserve the right to extend the closing date.

This Green and Pleasant Land!



Emma Norman dons her green outfit and, with banner grasped firmly in hand, campaigns for the organisation that everybody should know about

In a recent advertising campaign the organisation Friends Of the Earth gave five billion reasons why you should join them: The Human Race. These same five billion reasons could be given for joining the environmental organisation Greenpeace. Both Greenpeace and Friends of the Earth have the same object at heart:

To conserve the planet for future generations.

Most people have heard of Greenpeace and are aware of the work they do, but for those who do not know it is, in its own words: "An international Environmental Pressure group which maintains complete independence from all political parties anywhere



in the world. Its aim is to bring an end to the pollution of the planet and to protect wildlife." It operates throughout the world.

It began in 1971 in Canada, with a protest voyage into a nuclear test zone at Amchitka. The test was disrupted and the area is now a bird sanctuary. In 1977 Greenpeace opened its first branch in Britain where, to date, it has over 230,000 members. Today there are over 3 1/2 million members Worldwide with the majority of these coming from the USA. This figure is one of which they are very proud, but many members are always needed. Despite the successes, there are

many objectives still to be reached.

Why do we need Organizations like Greenpeace? What is so harmful with dumping waste? The wastes are concentrations of poisonous substances: metals like Copper, Lead, Zinc, Chlorine, Mercury and Arsenic. These build up in the environment and poison food sources. For example: in Wood Spring, North Devon people were warned not to eat more than 1lb of Shrimps or 4oz of shellfish due to the excess of cadmium in harmful poison) therein. Two million tonnes per year of such waste is dumped in the Irish sea.

Although a direct connection has

not been established between waste-dumping and the virus that killed 17,000 seals last year (and appears to be re-emerging amongst the remaining Common Seal populations), the possibility that pollution was the cause cannot be overlooked and emphasizes the urgency for action. There are many more examples of such atrocities caused by "Man's inhumanity to man", including Chernobyl, the Alaska oil spillage, the plight of the whales, elephants, too numerous to list, but what of the successes?

In the past five years Greenpeace has stopped hunts of seal pups; helped bring about an end to commercial whaling; stopped the burning of hazardous waste in the North Sea, US waters and the Mediterranean; stopped the dumping of radioactive wastes at sea; helped persuade the British Government to spend millions of pounds cleaning the beaches in Britain and filtering the waste from coal-fired power stations.

The majority of these achievements are brought about by sheer bravery. Greenpeace activists went out in dinghies and positioned themselves between the harpoons of the whaling boats and the living whales. As a result Commercial whaling is now banned. In the North Atlantic, Greenpeace dived its inflatable dinghies under barrels of radioactive waste, therefore dumping of such waste at sea has now been stopped. Volcanoes physically prevented baby seals being killed in the Orkney Isles and Newfoundland by throwing themselves between the hunters and the pups.

The level of the danger in which these people place themselves became apparent when one man was killed on the Rainbow Warrior when it was destroyed whilst trying to stop the testing of nuclear weapons off the coast of New Zealand. These people are risking their lives? What can you do to help?

You can join them in their fight to protect the planet by becoming a member, information and membership forms are available, from Greenpeace 20-31 Blenheim Green, London N7 8BN. (Subscriptions cost £12 for a single, £17.50 for the family, £6 arranged, and £20 for overseas).

Greenpeace is at present campaigning for Nuclear free seas in Germany, Nuclear free Irish sea, stopping toxic waste dumping in Spain, working to protect the seals and dolphins around the British coast, and trying to get Australia declared a Waste Free.

The Game



ENTITLED 'Rainbow Warrior', the Greenpeace game is set to raise a few eyebrows. It has already provoked a Conservative MP to condemn it, believing that many computer gamers may not about industrial pipelines and use other forms of violence in order to 'practice' the Greenpeace methods shown in the game.

The gameplay is split into several sections, all of which are based on separate ecological problems. You play a Greenpeace supporter who is determined to protect wildlife and stop pollution of the natural world, and each section is a mini-game all in itself. Among other tasks you have to prevent whaling ships from catching their prey by manœuvring the Greenpeace boat

between the whale and the ship, save programs from straying into a beam that has broken through the ozone layer, and you must stop sea-calling by ruining the pelts of seals, this is done by spraying them with green paint.

Although the game features very strong opinions, Microprose has made the graphics slightly humorous in tone, so as to appeal to a wider market.

The sprites are enormous and the animation as smooth as baby's bottom, they have even been described as Python-esque.

Rainbow Warrior should soon be available on Amiga and CD4, and it is to coincide with the new Greenpeace album, Rainbow Warriors, which features such artists as U2 and Simple Minds.



RAINBOW WARRIOR

RAINBOW WARRIORS

The world's first environment friendly software!

At last, an all action game that presents a solution to the environmental dilemma faced by the entire human race. Rainbow Warriors is an action game with real depth and meaning. It simulates seven campaigns of Greenpeace members over the last few years.



*Micro
Style*

GAMES FOR ADULTS

THERE have been many cult shows from television. By 'cult' I mean shows which have seen more repeats than a particularly bad curry. Examples are *The Wooden Tops*, *Bill and Ben - The Flowerpot men*, *Andy Panda* and *Thunderbirds*. All of these have something rather significant in common, their actors were more wooden than the cast of *Neighbours*, and have more strings than all of *Witchhick's* Ernest Thribben put together. They all featured puppets.

The latter of those repeatable shows is the one which we are to spotlight here, mainly due to the fact that *International Rescue* are their first pixalised exploits, thanks to software company *Granada*, whose previous credits include *Furiosa*, *Flamenco* and *The Running Man*, as well as the forthcoming *Livestock - The Computer Football Game* (which has been widely acclaimed). *Thunderbirds* is its first venture into the world of Gerry Anderson, although contacts have it that it will not be the last (look out *Mysterons*...!).

Released in the early sixties, *Thunderbirds* gained popularity at an alarming rate, due mainly to Anderson's previous successes - *Captain Scarlet*, *Fireball XL5*, and *Stingray* to name a few - but, there was something rather special about the Tracy family that enthralled viewers everywhere, and the plots were far more detailed than the simple rescue missions that they seemed to portray.

The characters created were not as shallow as one may expect. Each one had a more complex background and history than many at the forefront of soap operas, thus making for a slightly more involving program. The personnel of *International Rescue*



THUNDERBIRDS

runs as follows:

Jeff Tracy - Founder of *International Rescue*, and father of five sons (who are also in the 'family business').

Steve Tracy - Eldest of the brothers, pilot of *Thunderbird 1*, and second in command if anything happens to his father.

Plegl Tracy - Pilot of *Thunderbird 2*, also deputises on *Thunderbird 3* when Alan is not available.

Alan Tracy - Pilot of *Thunderbird 3*, also alternates with John on some of duty on *Thunderbird 5*. Is romantically involved with *Tia Tia*.

Gordon Tracy - Pilot of *Thunderbird 4*.

John Tracy - Spends most of his time on *Thunderbird 5*, due to his days as a radio ham. Occasionally replaces Alan on *Thunderbird 3*. **Braine** - Designer of the *Thunderbird* equipment and technical genius. Has a tendency to stammer.

Kyrano - The Tracy Island cook. Is half-brother of the villainous Hood.

Tia Tia - Kyrano's daughter. Is in love with Alan, and helps Braine maintain and repair the *Thunderbird* equipment.

Lady Penelope - Ex-British agent, famous for her unique pink Robin Hood FAB 1.

Parke - Failed sales cracker who was caught by Lady Penelope. Acts as her butler as 'cover'.

All of these except Lady Penelope and Parke were based on Tracy Island, which served audiences with its great technical sequences when the *Thunderbirds* lifted into the air, most memorable of which was the swimming pool which did back to allow *Thunderbird 1* to cut from its underground launching pad. The craft themselves were superbly designed, thus encouraging kiddies everywhere to request a home version of *Thunderbird 2* - the most ingenious of these all - courtesy of Dinky toys. *Thunderbird 2* was popular because of its ability to store other pieces of equipment - namely *Thunderbird 4* - in its stomach.

It is this depth, characterisation, gripping plots and technical brilliance that made *Thunderbirds* the most successful of Gerry Anderson's productions, and it is for this reason that the computer game should not go without notice. The game has many of its own merits, being the best arcade adventure to see



these shows for many months is just the first.

The game features four different missions. You must control two characters in each and perform certain actions in order to come out unscathed (although Thunderbirds can never die, and so unlike the lead Anderson) and acclaimed as a hero. The first - Mine Maniac - involves Alan and Brian who have to rescue some trapped miners. This is not a simple operation as priority lies in shutting off the mine's pumping station before it floods the whole mine, thus rendering all to be rescued by a final team, causing embarrassment and chaos, to seek other help - like I said, Thunderbirds cannot die.

The other three levels - Sub Crash, Bank Job and Countdown to Terror - respectively - are all in a similar vein, although time limits in which to complete your task get progressively more restrictive, and the problems connected to the mission become harder to figure out.

Graphics throughout are very pretty and true to Anderson's original ideas. Presentation throughout is of a very high quality, for instance, before each mission there is a small digitised animation sequence from the old episodes, for the four versions are all based on existing, single shows, although only loosely to those who own the videos



will find no faults in creating. The famous theme tune - 5... 4... 3... 2... 1... Thunderbirds are Go! - Boom... Boom... Boom... etc. - has been re-written to a rather good house version, which accompanies the sequences but leaves the game itself its original sound effects.

Another small thing that I think helps sway my attitude about this game is the small animated sequences which the graphics go through at certain stages (similar to Little Computer People) giving the game more character. Each mission has a password which once discovered allows passage straight into the section without the earlier waffle.



work I think. Roger Grandson, and Gerry Anderson will all be very pleased with the final product of some soft talking and hard work. I know I am as a lifelong Thunderbirds fan and I can see myself getting as gripped by this computer version as I was with the original series. Although it really is one of the older titles in the book this time, for me, Thunderbirds ARE gold!

Bill Henderson

Tape are the programming team behind Chubby Chase, Tommen, and the more recent Prices - proving that they are really ardent adventure experts - although Thunderbirds is the most complex of all, and probably the most complex.

Grandson - £24.95



INFO

Gameplay: 95%
Graphics: 95%
Sound: 95%
Usability: 85%
Overall: 91%



GILBERT - ESCAPE FROM DRILL

Again Again - £9.99/£14.99

Follow Gilbert on his snazzy escapades as he explores the planet Drill, whilst searching for the missing parts to the Millennium Dearth.

THE cute but snazzy alien, Gilbert, is bunkering once again for the bright lights of ITV. He has the contract, and the rig, now all he needs to do is find the missing parts of the Millennium Dearth and sign on the dotted line.

Gilbert has twenty-four hours to find the five missing parts, hidden by the discerning TV watchers of the planet, before some other, hopefully more talented, star signs the contract. No Drillies possess a full set of facilities and those would-be subscribers have actually left clues in various video games which should help Gilbert fulfil his quest in scotching the minds of the

Earthly young.

Gilbert starts his adventure in the middle of his home town, though why someone would wish to build a town in such a hostile place is beyond me. As you drifters - yes Gilbert has no feet - across town you are assaulted by all manner of beasts.

Fortunately Gilbert can dispatch these creatures with a flick of the head and a globin of meat. This formidable method of defense is finite but fresh supplies of the sticky green stuff can be found in milk bars, along with the video games which gave the missing parts. If Gilbert manages to shoot an entire wave of creatures, a flying

plate of wobbling jelly gains its way onto the screen. Shooting this either liberates another, a can of beans, or a piece of cake. Oh which one of these items can be carried at any one time. If the beans are eaten Gilbert blows and floats, enabling him to reach the upper levels of the forest and change levels when in the sewer system. Eating the cake brings Gilbert back to Drill with a bump, obviously made with the wrong brand of flour.

The five video games which Gilbert must complete offer a welcome

hour of time.

Unlike other games the attacking creatures never cease to arrive as they come forth in a never ending variety of guises. The playing area is not that small, and not only is there a map to contend with, but a jungle, sewer, tree top and underwater adventure to occupy the mind. Scouting good fun even if the gameplay is a little too repetitive.

Adrian Pampary

INFO

Graphics: 85%
 Controls: 82%
 Sound: 76%
 Usability: 88%
 Overall: 79%

break from the more mundane pleasures of shooting monsters in the street. Some of the better games include Earth

Invasion - in which a one-eyed monster blasts at rows of descending humans.

Brain Drain - play against the computer to see who can match the most tiles. And Spotlight at the OK Corral - spot 'em cowboy. Failure to complete a video game produces a loss of one

Enigma Variation, headed by Mark Garmahfield and Richard Bayle, are the people responsible for the latest TV presenter cognitive game and as a publicity stunt they sent all the A-list a free stick. Gilbert is throw around - what for their own.



SCRAMBLERS, in my opinion, could better be called because a graph of bumps on the back of one of those things it's scrambled brains for fun, dear. The only real damage passed here is one of less excitement coupled with even less pain than the pain thing.

Fifteen tracks of pure frustration lie before you, and failure to complete one within the time limit means a trip all the way back to the beginning. You start with the three dirt tracks, beginning with the over so easy track 'A', which is designed to allow you to save time by letting you complete at least one of the fifteen tracks. To complete the first track you simply have to ensure that the gear you are currently in (out of three) is correct for the slope you are travelling on. And that the bike you have the propensity of controlling

does not exceed one of the two extremes of speed - stalling, or going so fast you end up flying over the handlebars.

On the second dirt track you hit your first real obstacle, and find a use for the other controllable movements, like raising of the front and back wheels - just at the same time I further to add. This track also teaches you the precise times which the gear is hand, precise control over the bike's speed. When travelling over bumps and ripples you must ensure that your speed does not exceed a maximum set of limits. Failure to do so results in the action freezing and a message informing you whether you were too fast or too slow. You may note that you do not even have the satisfaction of seeing your bike hit the dirt.

By the time you have reached the third track you must have mastered the raising and lowering of the wheels - if not you might as well turn your computer off and start again. Here the main obstacle comes in the form of small holes, which require you to raise your bike wheels whilst passing over them. Next come three tracks with bottles, the car variety, strips and watery pits. On these courses the control over your speed becomes too critical for your joystick to handle with any degree of accuracy, and you inevitably crash.

If speed and gear selection were all you had to worry about, then life might seem approach something like simple, but you also have to ensure that you stay on the course as indicated on a small strip of the screen which shows sideways movement.

If you are a masochist who enjoys the sound of revving engines then you might enjoy this game - but I doubt it.

Adrian Pumphrey

Gremlin Graphics, once part of the Birmingham Software Exchange, are relocating to *Stoney Sheffield* where, hopefully, new ideas will ripple.

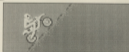
INFO

Category: 29%
Graphics: 83%
Sound: 88%
Language: 87%
Overall: 54%

SUPER SCRAMBLE SIMULATOR

Gremlin - £9.99/£14.99

Mount up and pit your sanity against the latest game to bring you all the thrills and spills of high risk sport without the risks and without the thrills and even less of the sport.





RED STORM RISING

Microprose - PC - £34.99
Tom Clancy's novel of World War III submarine warfare comes to life as you take the helm of a US Attack submarine. Sooner or later, it had to happen.

WHEN Islamic fundamentalists destroyed the Russian satellite, war broke inevitable as the Great Bear set its sights on the Middle East. The combat that followed as the super powers clashed would be partially decided by naval actions in the crucial North Atlantic. As a submarine commander your strategy and skill will be vital in hunting and destroying enemy subs, task forces and troop carriers.

Before you take on such responsibilities you can test your skills in a series of training missions, during which time you can learn how to use the sea's thermal layer to avoid enemy detection, the importance of keeping your speed low and a reasonable depth to ensure your propeller doesn't cavitate in the water (and alert enemy subs), and how and when to use the weapon systems. The toughest part of submarine warfare is tracking down your enemy without him finding you. This means you can't use active sonar, which would give away your position. You must find a quiet part of the ocean and listen for the signs that your passes near can find. Once you have a good sonar contact (above 90% - many torpedoes are fired in bare

at poor contacts) you can fire your torpedoes at a submarine, or missiles at the surface contact.

Although you can play individual scenarios, the real challenge is when you play the Red Storm Rising Scenario and play your part in World War III. Here your missions are carried out and directly affect the backdrop of the war. Graphic sequences set the scene in NATO battles against the Soviet onslaught. An example of this is when you're sent in to intercept a task force carrying troops. Not only do you have to track down and attack the convoys, your success at sinking the troop carriers will decide the success or failure of the Soviet attack. If you do badly, the map will rapidly turn red.

As an idea, Red Storm Rising can't be faulted. It's based on a best selling book by a bestselling author and the computer implementation provides all the options and problems that face a modern submarine commander, who must do his best to win battles that affect the outcome of the war. Unfortunately, the game falls down on presentation. Although the clips showing the progress of the war are good, the actual screens

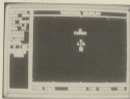
used in the battles (most of the game) are dull. In fact, there's no comparison to games such as Silent Service or 688 Attack sub. As a result this will deter many gamers from a challenging and fascinating game.

Tony Hetherington

Red Storm Rising was written, in conjunction with Tom Clancy, by Sid Meier who was the author of 1-2-3 Strike Eagle and Silent Service. He was also the co-founder of Microprose.

INFO

Gameplay: 84%
Graphics: 35%
Sound: 18%
Usability: 47%
Overall: 50%



Islamic religious radicals outwitted Russia's huge oil refinery complex at Kuwait's oil fields in Kuwait. This destroys one third of the nation's production capacity. The Soviet Union faces years of crippling energy shortages and an immobilised military.

STORMLORD

Hewson - £14.99 Disc, £9.99 tape.

AN evil Queen dominates the once beautiful and peaceful land. It was a land where fairies frolicked and bunch herons lay around being bunch. It was also incredibly dull, so it's just as well that the Queen arrived with her evil minions. Wicked creatures roam the land, man-eating plants have taken root, and frolicking fairies lie captured in cages. Now Stormlord must rescue the fairies and liberate the land by destroying the Queen.

The land is represented on screen by a series of levels, each consisting of a number of sideways scrolling screens.

Decile plants, statues, steps, and boulders form the platforms on which our hero can walk and jump. Control is somewhat

limited although you can build up power for a bigger leap by holding down the fire button. The screens also contain a very useful but inexplicable network of springboards that can haul our hero quite remarkable distances, and quite often land him by a durned-in distress. You can also be sure that there will be another springboard strategically placed for the return journey.

While Stormlord is running, jumping and leaping about the land various nasties are out to get him. Huge worms want to nibble him. Dragons want to fry him, fire want to roast him and Venus fly traps want to chomp him. However, he can hit back with a self loading, Dying word, but it's so ineffective that it's best to get out of the way.

Fairy folk and foul creatures clash in a game that aims to take your screen by storm, as the heroic Stormlord rouses rebellion against a malevolent monarch.

There are also a number of traps such as boulders and acid which drop down from above and doors which block your path throughout the game.

Luckily, there are useful objects scattered around the landscape, but our brave and incredibly bunch heron can only carry one at a time, which I suppose isn't that surprising when you consider that a key is almost as big as the door it opens! Apart from the keys,

there are other objects whose function is less obvious, which can make it difficult to always carry the one you need next, resulting in quite a bit of backtracking. This can be reduced by leaving them near springboards.

The result of all this is a game that's a cross between a sideways scrolling shoot-em-up and a platform game. Although I doubt this will appeal to either game's supporters as it is

INFO

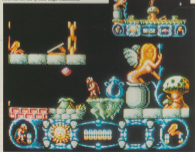
Gameplay: 77%
Graphics: 49%
Sound: 92%
Usability: 88%
Overall: 41%

rather silly and equally uninspiring.

Tony Hetherington

Stormlord was originally written by *Wagstaff Circus* (C&A conversion by Nick Jones), who also produced *Cyberoid II* and *Excelsis*.

Freedom man! Not only is this child-frolicking mauler, but check out the size of these magic mushrooms!



ROCKSTAR ATE MY HAMSTER

WITH all the sweat and toil of making it to the top, *Rockstar Ate My Hamster* provides a gritty insight into the topsy-turvy world of rock 'n' roll management. Through the eyes of a somewhat slimy lamp of no-talent

Codemasters - £9.95 Tape
Sex, drugs and Rock 'n' Roll - one of these is featured in Codemaster's latest full price game. Does the game live up to its rather peculiar name?

Gameplay: 90%
 Graphics: 85%
 Sound: 80%
 Usability: 85%
 Overall: 80%



arrangement, called *Crail Pitt*, you act as a manager, booking and driving and calling the shots on the band's future plans. But getting to the top is not so simple.

The ultimate goal of *Rockstar*, which has the intriguing feeling of a board game, is to go *Platinum* by topping the charts, the world and elsewhere. With your lazzish sidekick *Clive*, who insists on using groovy talk, your first task is to choose the band. You are allowed up to four members and can select from a list of about 50 stars who appear one by one on a video screen in front of you. It is probably best to

take so there is the option of second-hand gear, or even lower to the dodgy knock-offs.

This is where your climb to the top or low-life stall into the wilderness begins. To start with, you have four options: practice, gig, publicity or gifts. It is probably advisable to lock them away for a bit of a jam, give them some publicity and then hit the road. The practice sessions cost money, so you have to choose your studio time wisely. The publicity, which is dabbishly handled by the sceptic *Clive*, involves the aptly named *Star Newspaper* and includes a subtle reworking of that infamous headline concerning a certain *Star* eating a hamster. Publicity stunts, however, do not always get covered by the press, but that's a chance you have to take.

If you decide to gig you have to select a venue from the list of pubs, clubs, Universities, Concert Halls and stadiums. For the first time out it is probably best to play to cheap and easy-to-kill venues. A decision on the ticket price has to be made and the number of nights playing. After each gig *Clive* informs you of the number of tickets that were sold and consequently the amount of money made. At the start of the game money is the key to success especially if you have chosen band members with high salaries, they have to

be paid on a weekly basis.

If gigs are going well and you have succeeded in front page guitar news, the phone will start to ring with offers of record contracts, charity gigs and potential sponsorships. When a satisfactory record deal is reached you have the option of cutting the deal and shooting a promo to begin your claim on the record sales market and a position in the national charts. The prospect of shooting a video means more decisions on producers, locations and image details with every decision being of paramount importance for the band's future. With some suspiciously recognizable names the band under your guidance will enter the realm of the weekly charts and face the problem of staying at the top, paying taxes and using bootleggers.

Rockstar Ate My Hamster is an enjoyable game even though the graphics are not particularly good. It is addictive with reasonably good sound for its caliber and price, and for this reason alone it is worth considering.

Mark Jones



choose the maximum of four band members as somewhere along the road temptations creep in and the odd fellowist attack tends to deplete the ranks. As for selecting stars, which are loaded from a data cassette, it varies from player to player depending on who can handle the big money stars and tightie budgets.

Each star bears a distinct likeness to some of the top names in the pop world - including *Dumbo Ruggins* (Kylie Minogue), *Drew* (Bros) and many more. With your band together it's time to buy the gear. New equipment costs money but it's one step to true professionalism, with amplifiers that go to eleven for that extra push over the cliff. However, being a leader is not to everyone's

Colin Jones, the programmer of all the formats of Rockstar, is undoubtedly, and without a doubt in any way, Worth

Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the expert programs from *Your Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

... Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Your Commodore Software Service* makes available all of the programs from each issue on both cassette and disk at a price of \$6.00 for disk and \$4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFOSNET LTD, 3 River Park Estate, Berkhamstead, Herts, HP4 1BL.
Tel: (04427) 76661

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied in totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strip containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strip. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the programs, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

APRIL 1989

BASEX - Give your C64 new sound, graphics and toolkit commands as well as a machine code assembler with this Basic extension.

AUTOSCROLL - Professional text screen scrolling with this C64 utility.

BALANCE SHEET - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).

ICORN 64 - Add icon control to your own programs with this C64 utility.

80 COLUMNS - A suite of Hi-res graphic commands for C128 80 column users.

ZEUS - A disk file copier for Plus/4 users. Available on disk only.

MULTI-PRECISION ARITHMETIC - A suite of powerful mathematical routines expanding your C64's number crunching abilities.

STORAGE SPACE - Use the RAM behind the C64's Kernel and Basic ROMs for data storage (available on disk only).

ORDER CODE

DISK YDAPR89 \$6.00

TAPE YCAPR89 \$4.00

MAY 1989

ANTI-FREEZE - Protect your C64 programs from cartridge-based freezes.

MULTI-COLOURED LIST - Brighten up your C64 Basic listings by adding on-screen colour.

PRINTEX - A suite of powerful screen printing commands to add to your Plus/4's Basic. (disk only).

BANKER 128 - Money Management for C128 disk users.
TURBO TOUCH - A superb typing trainer for C64 disk users.

ORDER CODE
 DISK YDMA189 £6.00
 TAPE YDMA189 £4.00

JUNE 1989

RETURNER - A handy routine that allows your C64 programs to re-load a disk based menu. (disk only)
LINE INPUT - Improve your C64's INPUT routine
PROGRAM ANALYSER - A suite of programs to help you examine your C64 Basic programs in depth.
ASM ASSEMBLER - A powerful 6510 assembler for the C64

ORDER CODE
 DISK YDJUN89 £6.00
 TAPE YDJUN89 £4.00

JULY 1989

MOUSE 80 - Use a 1251 mouse in 80 column mode on the C128.
QUICK SEARCH - Add a FIND option to C64 Basic.
SPRITE LIBRARY - Add a collection of sprites to your growing sprite library (C64)

HEAD FOR HOME - An update on our popular C16 and Plus/4 game. Available on disk only.
WINDOWS ON THE 64 - Powerful windowing commands for the C64.
SPRITZ SPRITE - Give your Plus/4 or expanded C16 sprites - just like a C64. Available on disk only.

ORDER CODE
 DISK YDJUL89 £6.00
 TAPE YDJUL89 £4.00

AUGUST 1989

REASONING ON THE 128 - Programs for the first part of our expert system series (128 Disk only).
THESAURUS - Build up your own C64 thesaurus with this program for disk or tape.
WINDOZZ ON A MAZE - Generating mazes on your C64
ADDRESS BOOKS - A cassette based address book for C64 users
DISK SCRAMBLER - Protect your disk data from prying eyes (C64 disk).

ORDER CODE
 DISK YDAUG89 £6.00
 TAPE YDAUG89 £4.00

Cassettes or disks are available from March 1988. Please ring the editorial office (01-437 0825) for details of these.

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
SEPT 1989		TAPE (£4.00)	YCSEP89	
SEPT 1989		DISK (£6.00)	YDSEP89	
OVERSEAS POST £1				
			TOTAL	

NAME

ADDRESS

POSTCODE

I enclose a cheque/postal order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.

All orders should be sent to: YOUR COMMODORE.

READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP3 7ST

Please allow 28 days for delivery.

TERRAMEX

Grand Slam - £2.99 Tape

High in the icy vastness of space, something is headed our way. A giant asteroid is on collision course with Earth and only you can save the world.

WELL, actually the only person who can save the world is Professor Eyrstrain, who predicted the asteroid's coming some 30 years ago. The trouble was that no-one believed him, so he went off to a dark and dingy continent to sulk.

Taking the part of one of five stereotyped explorers you must search out the lager of Lemel, and then search out Professor Eyrstrain so he can build an asteroid deflector. Although if the world is saved then Grand Slam will be able to write more programs like this... Hmm, total destruction might not be so bad an idea after all.

When Terramex first appeared on the ST and Amiga it had the feel of an average 8-bit game, and now we see it in budget 8-bit form, which is better value for money. It doesn't make the game any more interesting though.

Having selected your character and waited for the rest of the game to load, you set off into the dark continent followed by your invisible bearers, in undramatic 3-D arcade adventure style. An irritating piece of music will have your teeth grinding by now, so go for the volume switch whilst concentrating on the smallest but colourful characters, and the generally dull

landscapes.

One of the first objects you encounter, and this game is object intensive, is a vacuum cleaner, which endows the power of flight. Yes, you'll believe a hovercar can fly, but not everywhere. One of the failings of Terramex then becomes obvious, you can hover along a certain distance and suddenly because it would make the game too easy, you stop in mid air, unable to progress further. There isn't actually anything there stopping you from going on, you just can't. One of the next objects needed is an umbrella which acts as a parachute. One good point is that you can ask your character for an idea when faced with a problem. If you're lucky the right object will flash up, and then it's down to you to use it properly.

Besides the object manipulation part, there's also the challenge of having to leap over snakes, crocodiles, and other obstructions, in an Indy Jones like manner. Should all efforts fail, then a quite humorous sequence follows whereby the asteroid plunges into the Earth. For sheer awfulness this rates some beating.

For aficionados of the arcade adventure, £2.99 represents good value, if unimpressive entertainment, but for those not converted to the cause, there is nothing here to convince that an expedition to find Professor Eyrstrain is a better idea than waiting at a bar for disaster to fall.

Duncan Evans

INFO

Collection: 60%
Graphics: 75%
Sound: 67%
Largibility: 71%
Overall: 68%

64

Grand Slam started life quite shakily, with Terramex being one of its poorer releases. Review offerings are generally much better.



The land has always been ruled by one of the four elemental magi. At the time of the great conclave, the Fire Wizard happened to be king, but news soon spread through the town that he has been humbly overthrown. The Earth Magi has fled, the Water Magi has disappeared and the Wind Magi has been grievously wounded.

Rumours of an mysterious magical beast abound as food raffles start to run dangerously low. In desperation, the mayor sends forth a hero who promptly disappears. Now, you have been summoned to help.

You have the choice to play one of six characters, each differing in their strength, armour and magical abilities. These attributes can be altered temporarily with the help of magic, or permanently by

finding relics - something that you will have to do if you are to get very far in the game.

Fire King follows the usual role playing ideas as so much as you have to wander round fighting monsters and finding key objects. There are books scattered around containing clues, although some information, especially in the Thieves Guild, has to be paid for.

The key to the game lies in manipulating the objects that you find and/or purchase. You have seven pockets each of which can contain up to nine similar objects, eg. keys, death spells, etc. The problem is that there are usually more than seven types of objects to carry, so you will need to plan ahead. On top of that, there is food to buy which will help to top up any energy loss in battle.



FIRE KING

Electronic Arts, Price £18.99 DvC
Role-playing down under is the name of the game. Gordon Hamlett casts a clairvoyance spell and determines if everything is fair dinkum or not.



Graphically, the game looks and plays a lot like Gauntlet with the top-down view and monster-producing monsters. On top of that, there is more of a feeling of adventure to the game rather than just one large shoot-em-up.

The problem with Fire King is the manipulation of objects. The pocket system is very cumbersome to use, especially in a 'real-time' situation. The monsters are everywhere and I suspect that fans of role playing games will find this too much of an arcade game for their liking. I must confess that knowing their previous track record, I expected something a bit more cerebral from this company.

Gordon Hamlett

INFO

Gameplay: 40%

Graphics: 51%

Sound: 41%

Usability: 50%

Overall: 50%

Written by an Australian Company called Micro Fairs and Licensed through SSG - their first venture away from carpentry.

PHOBIA

*Microsoft - \$14.95 Disc, £9.95 Tape
Transfer from Scribblephobia, fear of
horizontally scrolling shoot 'em ups, so
when the latest game from Microsoft
arrived I broke into a nervous sweat.*



INFO

Gameplay: 90%
Graphics: 90%
Sound: 85%
Usability: 80%
Overall: 92%

NOT another scary shooter I thought as the Duping Empire landed me the sealed package with a snarl. I could hold the sounds of shooting, see the disc writhing about in the light before I even opened it. I gripped my teeth as I removed the staples (and sent them back to Rick for a very mean zip-knife), and the disc leaped out, wobbling nervously, into the disc drive.

Surely the disc popped and I slipped into my groove room after a long, slow, deliberate, pondering look. My body bobbing from room to room, my eyes scanning in every hand. Scroll, shoot, kill, struggle!

But wait, hold onto your jackets! Phobia is different. Slightly more than not much different, but different enough.

There's strategy. The evil Phobos has captured the Galactic Emperor and you've got to rescue him. Why? Well don't ask silly questions. You need to navigate most of the Phobia planets to obtain a piece of his shield, with which you can attempt to rescue the Emperor from the sea where he's being held. Along the way, the minions of Phobos will be seeking to stop you, comprising horrors which obviously are resisted by sacrificial pods obtained from the mines. The trick is to stay one step ahead of the bad guys and avoid

having to collect pods every five minutes.

And here's the psychology. All the Phobia planets feature a specific distorted fear and reality based on human fears. Pools of water, spiders, confined spaces (you can guess how many this one is), dentists, and even death. I didn't see Psycho-phobia, fear of not being paid, so I hope the YC chaps don't slip that one to us shipped - Dep Ed.

Then there's the action. Hearing owlers and canyons, scrolling from right to left, very smoothly considering just how colourful and action packed the screens are.

There are power ups along the way, including pods, and a simultaneous two player mode, which can be slightly confusing at times, but guarantees an intense shooting experience. The graphics are very good, and are complemented by the excellent music and smooth crunchy bits.

Phobia isn't so difficult you have to start to play again. It's fun enough to play while eating a spaghetti sandwich (I was particularly impressed by the Death planet, with skeletons in coffins, and corpses swinging from ropes attached to the ceiling, and even the end of the level cut-scene) and have been designed to send a shiver up your spine.

With when you thought it was dull to go for a stroll. Phobos sees the ball hit in the hill, and laid the reviewer.

So, I decided to stop, by delivering a nose crumpling snoring thud. *Duncan "barking mad" Trapp*

Time Together is a legend of gutter strapping, which means the disc is almost certainly good for ages, but this is the big one.



JOURNEY TO THE CENTRE OF THE EARTH

US Gold - £19.99

Face danger, excitement and the unknown as you attempt to follow in the steps of 19th Century explorer Awe Salomson, that lead to the centre of the Earth.

If you were to imagine a game based on Jules Verne's *Journey to the Centre of the Earth*, programmed by the spirit that wrote *Jour of Awe*, you certainly wouldn't imagine this game. It's not the fact that the game encompasses the many features already noted from role playing, strategy or adventure gaming, the actual rate of plotted strategy and randomly generated events is incredibly disappointing.

The game begins quite well with an impressive display of the sort of music and graphics even a CGA PC is capable of, but if you persist the year, it's all downhill from here. You begin the game with a choice of four explorers and an expert guide, although after a rockfall you're on your own. The rockfall is the first of the "events", but more of this...

Rapidly, the game starts to go to hand-rim roadster level upon a screen showing the fall and severely limited map that

you have to navigate, displays showing your food, water, vitality and physical condition. There are also a score of icons, which must be selected to move, rest (to conserve energy), mine (to spread your food further) and one that leads to the medic screen. Have you got a diagnosis of your ills, and there's a first aid kit to patch them up.

Whenever you click the mouse icon you hear a few quiet footprints and something may happen, ranging from a silly event to an even sillier avoidance sequence. For example events include snakes that don't bite you and Apears; pebbles that trip you; and bumps on the head that turn into fractured skulls just because of a badly placed statistic. As for the avoidable sequences, try these two for examples. In the repetitive falling rock game you have to run up three stages while avoiding the rocks. Yes, you read this correctly, you run up even though you're going down into the Earth!

The second example is when you have to avoid a herd of stamping elephants. Exactly what a herd of elephants is doing beneath the earth's surface isn't explained, neither is how the thin tunnel you were in suddenly becomes large enough for a conglomeration of mammoths.

I find this the most annoying type of game as it promises to react, look and sound as good and delivers so little.

Tony Hetherington

INFO

Gameplay: 19%
 Graphics: 67%
 Sound: 63%
 Usability: 20%
 Overall:
 42%

*Journey to the Centre of the Earth is produced by French developers Chip, who also wrote *Jour of Awe* marketed by US Gold.*



Extending Basic

Get a printout of the variables of your Basic program with this latest routine in our series on extended Basic.

By Bernhard-Henry Lehmann

In this series of articles on developing extended Basic routines, we have already dealt, several times, with the way Basic stores its variables. Therefore, the next logical step is to write a routine which gives us a printout of the variables created by a Basic program and what they contain at any given time.

This is a very easy thing to do, because all that's required is that we loop through the Basic variable areas and print each variable name and content.

I wanted to make this facility possible as a direct command and in connection with the trace facility which we developed in the last article. I am including this in this installment for all you people who didn't buy last month's **YOUR COMMODORE** (shame on you!).

As a direct command, you simply

enter **VARS**, and all variables of the Basic program which you have just run are printed out. Of course, if you haven't run a Basic program or have given the direct **CLR** command, which clears all Basic variables, nothing will be printed out. The same is true if you have changed the Basic termlife. As you know, the moment you enter a Basic line by pressing return all variables are discarded.

In connection with the trace facility which we developed last month, you can call a printout of all variables after each line has been executed and the trace routine has reprised the line at the top of the screen. You will remember, that after each tracing step, the computer waits until you press any key. To get a printout of the variables simply press "V". If you want program execution to continue press any other key.

How Basic Stores Its Variables

Before we look closer at the program itself, let's briefly recapture how Basic stores its variables.

All Basic variables are stored directly after the end of the Basic termlife. To find the start of the variable area you have to push the zero page variable decimal 49/48, Hex 3D/3E. Each variable takes up seven bytes of memory.

The first two bytes contain the variable name. The way the variable name is entered shows what type of variable we are dealing with:

A floating point variable has both bytes of its name entered in simple ASCII. If the name consists of only one letter, the second byte contains zero.

An integer variable has 128 added

to both its name characters, and if there is only one character (which, as you remember, always has to be a letter), the second byte contains 128. (All this means, that in each case, bit seven of the eight binary bits is set.)

A string variable is distinguished by having the letter of its name entered in simple ASCII, while the second character has 128 added to the ASCII value, or 128, if there isn't a second character.

Finally, a function definition has 128 added to the first letter of its name, while the second character is simple ASCII or zero. (Incidentally, in the program given I have ignored function definitions. As always, this is partly due to my laziness and partly in order to encourage you to add this to the routine yourself...)

...The final five bytes of each variable contain the contents of the variable itself. This is correct in the case of the string variable: Here, the third byte contains the length of the string and the fourth and the fifth byte contain the base address of the string in the Basic textfile.

This is quite a clever way of going about things. Instead of storing the string itself in the variable area, which would of course take up copies of memory and make the whole variable area pretty difficult to manage, a pointer address which takes up only two bytes is stored in the variable area. The disadvantage with this method is of course, that the Basic textfile must not be changed. The moment it is changed, the pointer to a certain string in the variable area might not be correct any more. This is one of the reasons why you can't continue running a Basic program after you have changed the textfile.

The storage of an integer variable is also pretty straightforward. The value of the variable is stored in the third and fourth byte after the name, whereby the third byte contains the high byte of the value, plus the so-called sign bit, and fourth byte contains the low byte. You assign an integer variable by entering "N" after it, e.g. 87.

An integer ranges from -32768 to +32767. If you add these two numbers up you get 65535, which is 65536 short of one, because the zero is included as a valid number. And 65536 is the

maximum range of a 16-bit address. If bit seven, or the sign bit, of the high byte is set, the number is considered to be negative, if it is clear, the number is considered to be positive. For example, high 0 and low 100 represent +100, while high 255 and low 100 represent -100.

The most complicated variables of all are the floating point variables. This is a pretty long subject which I do not want to enter at this point. There is also no need to, because in our program we are using ROM-routines to convert the floating point number as it is stored in the variable area into an ASCII string which we then simply print out.

But, for the mathematician amongst you, here is how a floating point number is stored in the Basic variable area: The byte after the variable name contains the exponent, the byte after that contains the first mantissa, plus the sign bit, and the final three bytes contain mantissa two to four.

Printing the Variables

The routine which prints each variable name and the contents of the variable, is so straightforward, that it needs very little explanation.

The main loop, which I call VARS-LOOP, then of all prints the variable name and decides at the same time what type of variable has to be dealt with (lines 4180-4310). According to these tests the program flow then jumps to the appropriate routine: FLPOINT, if it is a floating point variable, INTEGER, if it is an integer variable, and STRING, if it is a string variable.

Before the value of each type of variable is printed, some spaces and an equal sign are printed, as well as the "V"-sign, in case of a string, and the "N"-sign, in case of an integer. This makes the printed more presentable.

In the case of a floating point variable, A is pointed at the low byte of the first byte of the floating point value and Y is pointed at the high byte (lines 4470-4540).

After the ROM-routine at EBBA2 is called to put the value into the first of the two floating point accumulators.

Next, the ROM-routine at EB0DD converts the contents of the

floating point accumulator into a printable ASCII string. On exit from that routine, A and Y point to the base of that string.

Finally, the string is printed in the usual fashion, with LABEL, which prints any string point at by A/Y and terminated with a zero.

An integer variable is printed by loading its high byte into A and its low byte into X. The ROM-routine at EB0CD, which prints any number contained in A/X, prints the integer value out.

To print a string variable, we first get the length of the string and store it in Z35 (line 3450-3470). Then we transfer the address which points at the string in the textfile into zero page Z31/Z34 (lines 3510-3560).

Finally, we use indirect-Y to print the string, letter by letter. This is done with a loop which is limited by the length value contained in Z35 (lines 3680-3690).

After a variable has been dealt with, VARSLOOP ends by incrementing the pointer address by seven (lines 4770-4780). The vital loop exit test, which tells when there are no more variables, is done at the beginning of VARSLOOP (lines 4070-4120). This is because the routine has to exit at once if there are no variables at all!

The end of the Basic variables area is contained in zero page Z3F/40. If this point is reached, there are no more variables. What follows are the Basic arrays with which we are not dealing at this point.

Final Notes

As I was developing the variables printing routine I discovered to my great puzzlement that certain variable names did not work properly. For example "CC" was stored as "C" and "CI" didn't work at all and resulted in a syntax error report.

Then I discovered the reason for these irregularities: The extended Basic commands we use starts with a letter, that is a normal ASCII character. If you declare a variable by entering the LET command (as one usually does), then the variable name too starts with a letter!

To overcome this I changed the main routine in the following way: First, I added two lines, which test for

a state character (lines 550-560). If a token has been found, which means that the line starts with a Basic command, the routine jumps straight to \$ATED which evaluates the Basic command in the usual manner.

Next, before starting the tests for our extended Basic commands, I save the Basic pointer \$QA, 78 (lines 600-610). If the tests are negative, that is, if none of our extended Basic commands has been entered, \$QA, 78 is recovered (lines 1070-1100). This resets the pointer to the first character in the line, resulting in a variable name being read correctly.



```

10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

```

800 NEXT
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990

```

```

0700
0710
0720
0730
0740
0750
0760
0770
0780
0790
0800
0810
0820
0830
0840
0850
0860
0870
0880
0890
0900
0910
0920
0930
0940
0950
0960
0970
0980
0990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990

```


COMMODORE REPAIR SERVICES

TO ADVERTISE YOUR COMMODORE REPAIR
SERVICES CALL: JAYNE HEIGOLD ON
0442 66551

RATE: £11.50 (+ VAT) PER SINGLE COLUMN CM
STANDARD SIZE 4cm x 1 col - £49.00 (+VAT)



TELEPHONE BOOKINGS WELCOME

VTS COMPUTER REPAIRS

01 - 427 8620

Simply send your unit to the address below
securely packed with full description and
any log.
COMMODORE ASSISTANCE, 80C ENGLAND,
+ MORE. £88.00

VTS COMPUTER REPAIRS

12 Lydy House, 65, Dorset Street,
London, SE1 8EG.
FAST, RELIABLE FORWARD

M.C.B. SERVICES

33, Albert Street, Mansfield, Notts
NG18 1BA Tel: 0523 650912

FIXED-PRICE COMMODORE REPAIRS

C64	£20.00
C65	£19.00
Plus1	£19.00
Amiga 500	£19.00
501	£20.00
502	£20.00
C64-20	£20.00

All prices incl. P&F (Labour, Parts etc.)
NO WORK TO DO Y. Please ring for details.

COMPUTER SYSTEMS

+ REPAIRS + SPARES
+ SALES + SUPPLIES



ESB
ELECTRONIC SERVICES
CHELSEA (20084)

COMMODORE SPARES & REPAIRS

REPAIRS

C64, C64 £25 inc.
C128, 1341 £32 inc.

Three month warranty, 1 week turnaround

SPARES

100114	£2.00	50-1042	£14.00
1010	£2.00	5001	£2.00
1020	£2.00	5002	£2.00
1021	£2.00	5003	£2.00
1022	£2.00	5004	£2.00
1023	£2.00	5005	£2.00
1024	£2.00	5006	£2.00
1025	£2.00	5007	£2.00
1026	£2.00	5008	£2.00
1027	£2.00	5009	£2.00
1028	£2.00	5010	£2.00
1029	£2.00	5011	£2.00
1030	£2.00	5012	£2.00
1031	£2.00	5013	£2.00
1032	£2.00	5014	£2.00
1033	£2.00	5015	£2.00
1034	£2.00	5016	£2.00
1035	£2.00	5017	£2.00
1036	£2.00	5018	£2.00
1037	£2.00	5019	£2.00
1038	£2.00	5020	£2.00
1039	£2.00	5021	£2.00
1040	£2.00	5022	£2.00
1041	£2.00	5023	£2.00
1042	£2.00	5024	£2.00
1043	£2.00	5025	£2.00
1044	£2.00	5026	£2.00
1045	£2.00	5027	£2.00
1046	£2.00	5028	£2.00
1047	£2.00	5029	£2.00
1048	£2.00	5030	£2.00
1049	£2.00	5031	£2.00
1050	£2.00	5032	£2.00
1051	£2.00	5033	£2.00
1052	£2.00	5034	£2.00
1053	£2.00	5035	£2.00
1054	£2.00	5036	£2.00
1055	£2.00	5037	£2.00
1056	£2.00	5038	£2.00
1057	£2.00	5039	£2.00
1058	£2.00	5040	£2.00
1059	£2.00	5041	£2.00
1060	£2.00	5042	£2.00
1061	£2.00	5043	£2.00
1062	£2.00	5044	£2.00
1063	£2.00	5045	£2.00
1064	£2.00	5046	£2.00
1065	£2.00	5047	£2.00
1066	£2.00	5048	£2.00
1067	£2.00	5049	£2.00
1068	£2.00	5050	£2.00
1069	£2.00	5051	£2.00
1070	£2.00	5052	£2.00
1071	£2.00	5053	£2.00
1072	£2.00	5054	£2.00
1073	£2.00	5055	£2.00
1074	£2.00	5056	£2.00
1075	£2.00	5057	£2.00
1076	£2.00	5058	£2.00
1077	£2.00	5059	£2.00
1078	£2.00	5060	£2.00
1079	£2.00	5061	£2.00
1080	£2.00	5062	£2.00
1081	£2.00	5063	£2.00
1082	£2.00	5064	£2.00
1083	£2.00	5065	£2.00
1084	£2.00	5066	£2.00
1085	£2.00	5067	£2.00
1086	£2.00	5068	£2.00
1087	£2.00	5069	£2.00
1088	£2.00	5070	£2.00
1089	£2.00	5071	£2.00
1090	£2.00	5072	£2.00
1091	£2.00	5073	£2.00
1092	£2.00	5074	£2.00
1093	£2.00	5075	£2.00
1094	£2.00	5076	£2.00
1095	£2.00	5077	£2.00
1096	£2.00	5078	£2.00
1097	£2.00	5079	£2.00
1098	£2.00	5080	£2.00
1099	£2.00	5081	£2.00
1100	£2.00	5082	£2.00
1101	£2.00	5083	£2.00
1102	£2.00	5084	£2.00
1103	£2.00	5085	£2.00
1104	£2.00	5086	£2.00
1105	£2.00	5087	£2.00
1106	£2.00	5088	£2.00
1107	£2.00	5089	£2.00
1108	£2.00	5090	£2.00
1109	£2.00	5091	£2.00
1110	£2.00	5092	£2.00
1111	£2.00	5093	£2.00
1112	£2.00	5094	£2.00
1113	£2.00	5095	£2.00
1114	£2.00	5096	£2.00
1115	£2.00	5097	£2.00
1116	£2.00	5098	£2.00
1117	£2.00	5099	£2.00
1118	£2.00	5100	£2.00
1119	£2.00	5101	£2.00
1120	£2.00	5102	£2.00
1121	£2.00	5103	£2.00
1122	£2.00	5104	£2.00
1123	£2.00	5105	£2.00
1124	£2.00	5106	£2.00
1125	£2.00	5107	£2.00
1126	£2.00	5108	£2.00
1127	£2.00	5109	£2.00
1128	£2.00	5110	£2.00
1129	£2.00	5111	£2.00
1130	£2.00	5112	£2.00
1131	£2.00	5113	£2.00
1132	£2.00	5114	£2.00
1133	£2.00	5115	£2.00
1134	£2.00	5116	£2.00
1135	£2.00	5117	£2.00
1136	£2.00	5118	£2.00
1137	£2.00	5119	£2.00
1138	£2.00	5120	£2.00
1139	£2.00	5121	£2.00
1140	£2.00	5122	£2.00
1141	£2.00	5123	£2.00
1142	£2.00	5124	£2.00
1143	£2.00	5125	£2.00
1144	£2.00	5126	£2.00
1145	£2.00	5127	£2.00
1146	£2.00	5128	£2.00
1147	£2.00	5129	£2.00
1148	£2.00	5130	£2.00
1149	£2.00	5131	£2.00
1150	£2.00	5132	£2.00
1151	£2.00	5133	£2.00
1152	£2.00	5134	£2.00
1153	£2.00	5135	£2.00
1154	£2.00	5136	£2.00
1155	£2.00	5137	£2.00
1156	£2.00	5138	£2.00
1157	£2.00	5139	£2.00
1158	£2.00	5140	£2.00
1159	£2.00	5141	£2.00
1160	£2.00	5142	£2.00
1161	£2.00	5143	£2.00
1162	£2.00	5144	£2.00
1163	£2.00	5145	£2.00
1164	£2.00	5146	£2.00
1165	£2.00	5147	£2.00
1166	£2.00	5148	£2.00
1167	£2.00	5149	£2.00
1168	£2.00	5150	£2.00
1169	£2.00	5151	£2.00
1170	£2.00	5152	£2.00
1171	£2.00	5153	£2.00
1172	£2.00	5154	£2.00
1173	£2.00	5155	£2.00
1174	£2.00	5156	£2.00
1175	£2.00	5157	£2.00
1176	£2.00	5158	£2.00
1177	£2.00	5159	£2.00
1178	£2.00	5160	£2.00
1179	£2.00	5161	£2.00
1180	£2.00	5162	£2.00
1181	£2.00	5163	£2.00
1182	£2.00	5164	£2.00
1183	£2.00	5165	£2.00
1184	£2.00	5166	£2.00
1185	£2.00	5167	£2.00
1186	£2.00	5168	£2.00
1187	£2.00	5169	£2.00
1188	£2.00	5170	£2.00
1189	£2.00	5171	£2.00
1190	£2.00	5172	£2.00
1191	£2.00	5173	£2.00
1192	£2.00	5174	£2.00
1193	£2.00	5175	£2.00
1194	£2.00	5176	£2.00
1195	£2.00	5177	£2.00
1196	£2.00	5178	£2.00
1197	£2.00	5179	£2.00
1198	£2.00	5180	£2.00
1199	£2.00	5181	£2.00
1200	£2.00	5182	£2.00
1201	£2.00	5183	£2.00
1202	£2.00	5184	£2.00
1203	£2.00	5185	£2.00
1204	£2.00	5186	£2.00
1205	£2.00	5187	£2.00
1206	£2.00	5188	£2.00
1207	£2.00	5189	£2.00
1208	£2.00	5190	£2.00
1209	£2.00	5191	£2.00
1210	£2.00	5192	£2.00
1211	£2.00	5193	£2.00
1212	£2.00	5194	£2.00
1213	£2.00	5195	£2.00
1214	£2.00	5196	£2.00
1215	£2.00	5197	£2.00
1216	£2.00	5198	£2.00
1217	£2.00	5199	£2.00
1218	£2.00	5200	£2.00
1219	£2.00	5201	£2.00
1220	£2.00	5202	£2.00
1221	£2.00	5203	£2.00
1222	£2.00	5204	£2.00
1223	£2.00	5205	£2.00
1224	£2.00	5206	£2.00
1225	£2.00	5207	£2.00
1226	£2.00	5208	£2.00
1227	£2.00	5209	£2.00
1228	£2.00	5210	£2.00
1229	£2.00	5211	£2.00
1230	£2.00	5212	£2.00
1231	£2.00	5213	£2.00
1232	£2.00	5214	£2.00
1233	£2.00	5215	£2.00
1234	£2.00	5216	£2.00
1235	£2.00	5217	£2.00
1236	£2.00	5218	£2.00
1237	£2.00	5219	£2.00
1238	£2.00	5220	£2.00
1239	£2.00	5221	£2.00
1240	£2.00	5222	£2.00
1241	£2.00	5223	£2.00
1242	£2.00	5224	£2.00
1243	£2.00	5225	£2.00
1244	£2.00	5226	£2.00
1245	£2.00	5227	£2.00
1246	£2.00	5228	£2.00
1247	£2.00	5229	£2.00
1248	£2.00	5230	£2.00
1249	£2.00	5231	£2.00
1250	£2.00	5232	£2.00
1251	£2.00	5233	£2.00
1252	£2.00	5234	£2.00
1253	£2.00	5235	£2.00
1254	£2.00	5236	£2.00
1255	£2.00	5237	£2.00
1256	£2.00	5238	£2.00
1257	£2.00	5239	£2.00
1258	£2.00	5240	£2.00
1259	£2.00	5241	£2.00
1260	£2.00	5242	£2.00
1261	£2.00	5243	£2.00
1262	£2.00	5244	£2.00
1263	£2.00	5245	£2.00
1264	£2.00	5246	£2.00
1265	£2.00	5247	£2.00
1266	£2.00	5248	£2.00
1267	£2.00	5249	£2.00
1268	£2.00	5250	£2.00
1269	£2.00	5251	£2.00
1270			

Emlyn Hughes



INTERNATIONAL

SOCCER



GAME

OF THE YEAR!

WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!

Quite simply the best football simulation available for the 64. Nothing short of superb.

ZZAP 64

**AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk**
Coming soon for Spectrum & Amstrad

Audiogenic

Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SL, England

Order by phone on 01 861 1166

