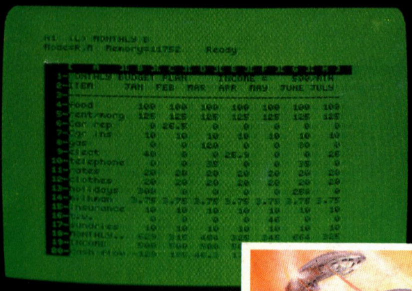
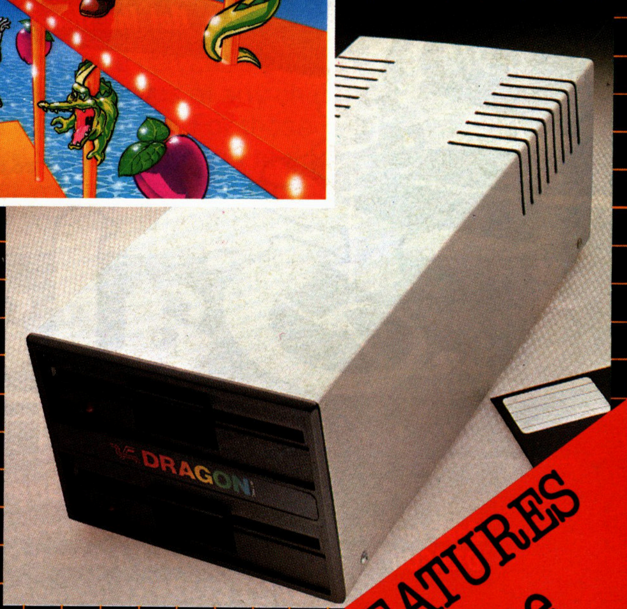
DRAGON Data Ltd.™

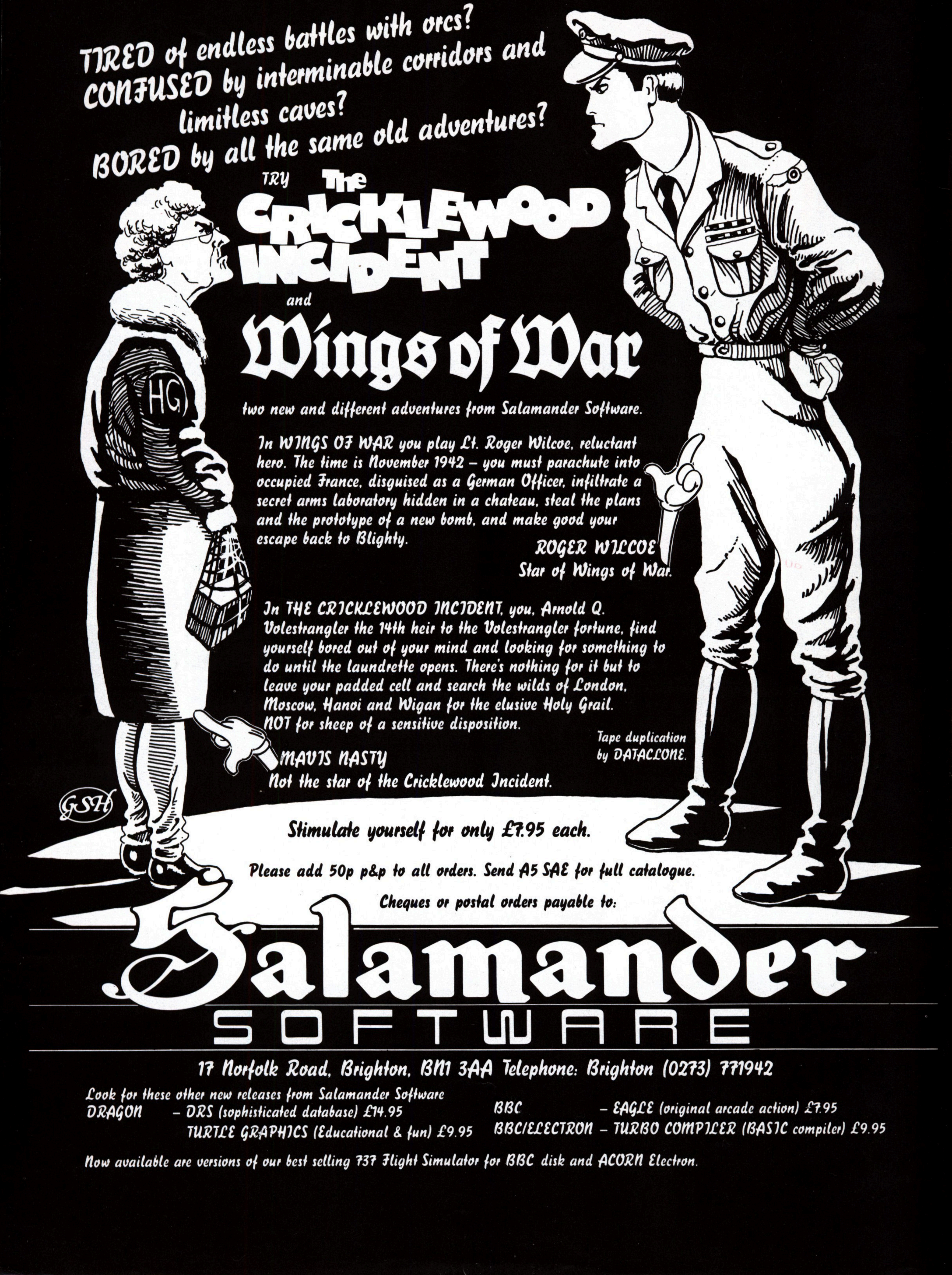
WORLD

No.2 February 84 60p

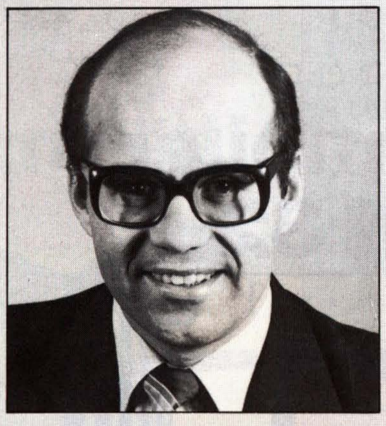




SPECIAL FEATURES
Software
Reviewed


TIRED of endless battles with orcs?
CONFUSED by interminable corridors and limitless caves?
BORED by all the same old adventures?
TRY The CRICKLEWOOD INCIDENT
and
Wings of War
two new and different adventures from Salamander Software.
In WINGS OF WAR you play Lt. Roger Wilcoe, reluctant hero. The time is November 1942 – you must parachute into occupied France, disguised as a German Officer, infiltrate a secret arms laboratory hidden in a chateau, steal the plans and the prototype of a new bomb, and make good your escape back to Blighty.
ROGER WILCOE
Star of Wings of War.
In THE CRICKLEWOOD INCIDENT, you, Arnold Q. Volestrangler the 14th heir to the Volestrangler fortune, find yourself bored out of your mind and looking for something to do until the laundrette opens. There's nothing for it but to leave your padded cell and search the wilds of London, Moscow, Hanoi and Wigan for the elusive Holy Grail. NOT for sheep of a sensitive disposition.
MAVIS NASTY
Not the star of the Cricklewood Incident.
HG
GSH
Tape duplication by DATACLONE.
Stimulate yourself for only £7.95 each.
Please add 50p p&p to all orders. Send A5 SAE for full catalogue.
Cheques or postal orders payable to:
Salamander
SOFTWARE
17 Norfolk Road, Brighton, BN1 3AA Telephone: Brighton (0273) 771942
Look for these other new releases from Salamander Software
DRAGON – DRS (sophisticated database) £14.95
TURTLE GRAPHICS (Educational & fun) £9.95
BBC – EAGLE (original arcade action) £7.95
BBC/ELECTRON – TURBO COMPILER (BASIC compiler) £9.95
Now available are versions of our best selling 737 Flight Simulator for BBC disk and ACORN Electron.

INTRODUCTION TO 'DRAGON WORLD'



Welcome, Dragon users, to our magazine, especially if you are one of the thousands of new users since our last publication in December.

1984 will be an interesting year, and starts with the news of a major order we have just secured to supply a customer in Spain. With this in mind, it would seem a particularly good time to extend a warm welcome to all our new overseas readers and to say how much we look forward to hearing from you and to receiving any contributions you may have for 'Dragon World'.

This month we are also featuring "AMPALSOFT", an educational software company, responsible for several interesting titles which we highly recommend to Dragon users.

I do hope you are enjoying our new magazine and if you have any ideas for improvement, please let us know.

May I wish you every success in 1984 and many happy hours with your Dragon computer.

SUBSCRIPTION OFFER

SUBSCRIPTIONS FOR DRAGON WORLD CAN BE OBTAINED DIRECT FROM DRAGON DATA FOR £4. THIS WILL ENTITLE YOU TO 6 BI-MONTHLY ISSUES MAILED DIRECTLY TO YOU AND THE FORM FOR THIS CAN BE FOUND IN A SEPARATE INSERT TO THIS MAGAZINE. IF YOU KNOW OF ANYBODY WHO MIGHT BE INTERESTED WHO CURRENTLY DOESN'T RECEIVE THE MAGAZINE, PLEASE LET THEM KNOW OF THIS FACILITY.

CONTENTS

EDITORIAL

Welcome to the first issue of 'Dragon World' for 1984. We would like to take this opportunity to wish all our readers a happy and prosperous New Year, and to welcome all the new users to this, their first issue of 'Dragon World'. Thank you to all the users who have sent in letters and contributions which were, and still are, gratefully received.

In this month's issue you will find a complete software list; if you are having difficulty obtaining software from the dealers and wish to buy direct, please post to: Mail Order, P.O. Box 40, Port Talbot SA13 1ZG West Glamorgan.

We will accept cheques, Access, American Express or Diners Card numbers. Orders may be taken over the telephone on 0656 744700, ext. 235, quoting relevant card number. We regret that no Barclaycard numbers can be accepted at present. The poster offer is still available, and we will continue to include both these offers in future issues of the magazine.

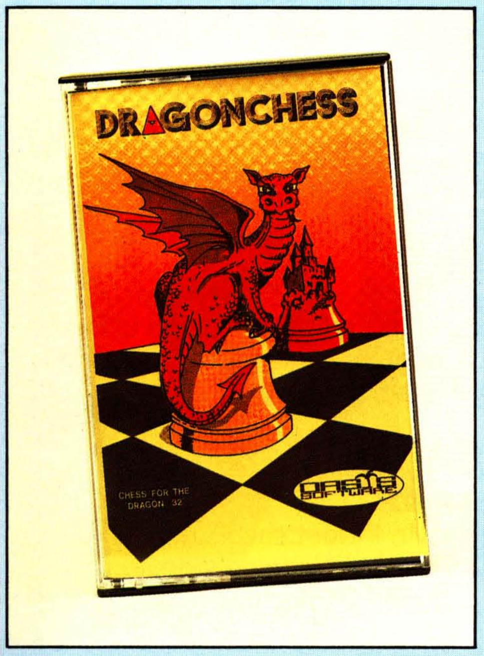
We should like to thank all the users who entered our "Dragon User of 1983" competition. There were many entries and it will obviously take time to decide on a winner from the many varied applications. We are at present compiling a short list and the results of the competition should be announced in the next issue of 'Dragon World' in April.

In order that we can keep our mailing list up-to-date, please inform us of any change of address or machine replacement. For those readers who have sold their Dragon, or intend to do so, please inform us so that we can discontinue your 'Dragon World'. If you do sell your Dragon, the new owner may wish to receive this magazine, so do please inform them of our subscription offer so that they do not lose out.

Editorial Team,
Dragon World,
Dragon Data Ltd.,
Kenfig Industrial Estate,
Margam,
Port Talbot, SA13 2PE,
West Glam.
Tel: 0656 744700

DRAGON 32

DRAGON CHESS £9.95



- ★ Six levels of play.
- ★ All legal chess moves including en-passant, castelling and pawn to queen promotion.
- ★ List of previous moves stored which can be displayed or printed.
- ★ Loading and saving of games positions from tape.
- ★ High resolution graphics which can be flipped round to make black or white play from either end.
- ★ Simultaneous text and graphics.

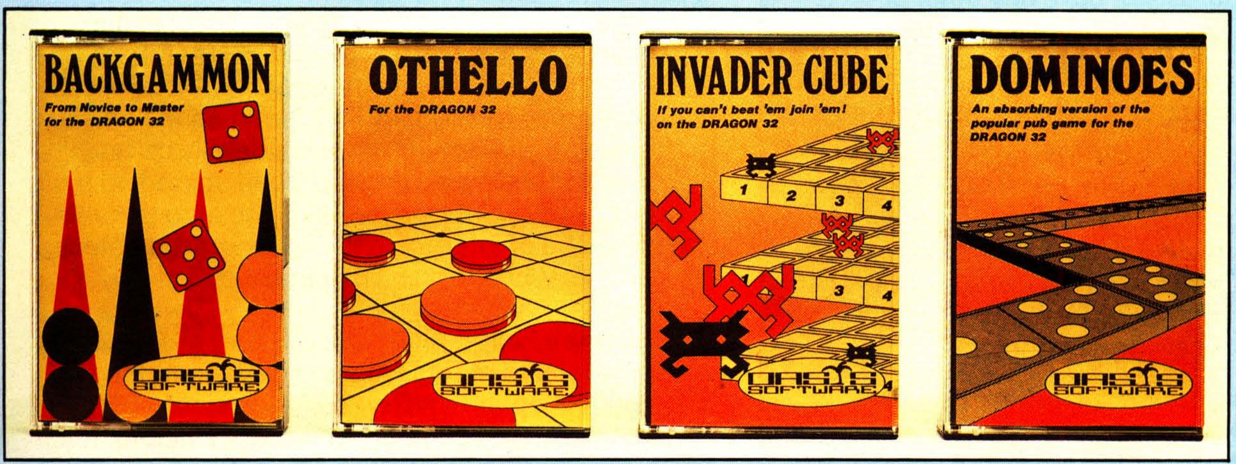
"Dragonchess is without doubt the best value chess on the market today . . ." (Microdeal)

Dragonchess is available from Boots and all good software stores.

- ★ Slides can be exchanged at any stage.
- ★ Best move hint.
- ★ Set up from any position.
- ★ Opening move library.
- ★ Moves may be taken back and play resumed from any point
- ★ Change level of play at any point in the game.
- ★ Will adjudicate games between humans.
- ★ Very high standard of play.
- ★ Professional packaging and life-time guarantee.

" . . . I thought this was an excellent version of the game. Well worth buying . . . "
(Home Computer Weekly)

BACKGAMMON £5.95 | **OTHELLO £5.95** | **INVADER CUBE £5.95** | **DOMINOES £5.95**



BACKGAMMON: The well known game of backgammon complete with full instructions and computer demonstration for beginners.

OTHELLO: At last. Othello or Reversi as it's sometimes known running on your Dragon 32. 2 levels of play, full instructions and computer demonstration for beginners.

INVADER CUBE: As well as being one of the best games of skill written for the Dragon 32 it also has some of the best pure machine code graphics we have ever seen.

DOMINOES: Two games in one with full instructions. Hours of frustrating fun this program has 6 levels of play and on the higher levels adapts it's play to its assessment of your ability.

Mind Games Compendium — All five games for just £19.95

ALL OASIS PRICES INCLUDE VAT AND P&P AND EVERY OASIS PRODUCT IS PROMPTLY DESPATCHED AND INCLUDES A LIFETIME GUARANTEE.

If your local dealer does not stock these products then let us know his address and we will contact him.

ACCESS ORDERS TAKEN BY PHONE 24 HOURS A DAY — 0934 419921

Access

OASIS SOFTWARE– Intelligence in code.

OASIS SOFTWARE
ALEXANDRA PARADE
WESTON-SUPER-MARE
AVON BS23 1QT
TEL 0934 419921

Please send me

DRAGON CHESS	£9.95 ☐
BACKGAMMON	£5.95 ☐
OTHELLO	£5.95 ☐
DOMINOES	£5.95 ☐
INVADER CUBE	£5.95 ☐
MIND GAMES COMPENDIUM	£19.95 ☐

I enclose cheque/PO for £
NAME ....
ADDRESS ....

USER CLUB NEWS

Firstly, I should like to welcome all new users to the User Club pages. The object of these pages is to give details of users' clubs and groups, as already stated in previous issues.

Secondly, I should like to apologise to Mr. J. W. Smith of West Yorks whose details I included in the last issue of 'Dragon World'. Mr. Smith no longer owns a Dragon 32 and does not wish any further contact from Dragon users. I should also like to apologise to all the people who contacted Mr. Smith, who will be forwarding your letters to me and I shall try to answer each one personally. In future, extreme care will be taken so that a similar misunderstanding does not occur.

Can I also include here a plea — if you sell your Dragon, move address, or have any other altered information from your registration card, please let me know so that I can adjust the mailing list accordingly.

The response to my request for information on clubs not already included on my list has been slow but with Christmas and the New Year, I am not too surprised. If you have written to me and are not included, please take into consideration the fact that this column is written over a month before you receive 'Dragon World'.

The new clubs I have heard from are:

SOUTH WALES

Anyone interested in forming a Dragon users club in the Llanelli area, please contact Mr. W. R. Collins on Llanelli (05542) 56917 (home).

ENGLAND

Cornwall

Mr. R. M. Starkie,
48 Old Roselyon Road,
Middleway,
Par,
Cornwall PL24 2LN
Tel: Par 4922

Mr. Starkie is interested in hearing from anyone in the area as he is proposing starting a Dragon Users Club.

Cheshire

Ellesmere Port Computer Club,
Chairman: Mr. G. Corcoran,
Tel: 051 327 3912

Meets alternate Mondays, 7.00pm, at Church Hall, adjacent to Ellesmere Port Golf Club.

Lancs

OCUS (Oldham Computers Users Society),
Meets at:
Coldburst Community Centre,
Rochdale Road,
Oldham,
every Tuesday, 7.30-9.30pm.

West Midlands

Pete Lucas (6809 West Mids)
Tel: Sedgley 72521

This is a small club that does not really wish to expand a great deal, but they are prepared to act as a 'springboard' for other users in the area to get together in a small geographical unit.

NORTHERN IRELAND

The last issue gave Mr. Peter Leach's details, and now he has requested that his telephone number is included. You can contact Mr. Leach on Antrim 65345.

On the Air

There is a club, possibly to be called 'Dragnet', now operating on Amateur Radio Frequency allocations. The 'club' had its first try out in early January and immediately gained 10 'members'. The 'net' is held on radio frequency 144525 MHz each Sunday morning at 11.30am (soon to be altered to 11.00am). The net is officially called 'computer users net' and has been started up by an enthusiastic Dragon owner. I am sure that this club will be of interest to the many radio amateur operators who are also Dragon owners.

Christmas Fair

It was good to meet Doug and Dave from the Dragon Independent Owners' Association at the Christmas Fair, Wembley. It is always helpful to get exchanges of ideas and hopefully some of the suggestions discussed may be implemented in the near future.

Even if clubs are unable to visit the exhibitions we are present at, there are still other methods of communication and I should appreciate ideas and suggestions from the clubs as to what they would like to see on these pages and within the magazine.

Thank you to all those club members who gave up some of their Saturdays to help us out by demonstrating in larger stores. Please make sure that you send your reports in though!

Cathy Hyde.

Cathy Hyde

DRAGON CHRISTMAS DRAW PRIZE WINNERS

1st PRIZE: Mr. M. G. Myatt, Gelli, Rhondda

Pictured above is Kevin Stephens, a marketing executive at Dragon Data, presenting the first prize of a double disk drive to the lucky Dragon 32 owner, Mr. M. G. Myatt (extreme right). Also pictured are Mr. Myatt's wife, Carol, and his two children, Christopher (aged 9) and Teresa (aged 11).

The Editorial Team at Dragon World would like to congratulate the Myatt family on their success, and hope it enabled them to enjoy the festive season even more than they usually do.

In addition, Dragon Data offered 20 runners-up prizes of £40 free software to be chosen from our catalogue. The 20 lucky Dragon users pulled out of the hat were as follows:

Mr. P. Francis, Pontypool.
Mr. Valentine, Congleton.
Mr. S. C. Lowe, Llanberis.
Mr. E. Parry, Ashington.
Mr. J. A. Bulwer, Wakefield.
Mr. J. Bull, Mansfield.
Mr. D. Foster, Leicester.
Mr. B. Wallis, Hull.
Mr. G. Hession, Leeds.
Mr. J. Tupper, Grantham.
Mr. J. Griffin, Hull.
Mr. J. Wall, Dyfed.
Mr. A. L. Hemsley, Rugby.
Mr. A. P. Jennings, Harrow.
Mr. D. M. Leary, Wirral.
Mr. W. Mellins, Darwen.
Mr. G. F. Sprigg, Middlesboro'.
Mr. J. F. Greenwood, Belmont.
Mr. P. A. Dove, Leeds.
Mr. Van Loveren, Merksam, Belgium.

CONGRATULATIONS TO ALL OF YOU

DRAGON PUZZLE 4

Here is the solution to the Christmas puzzle complete with the revealing hint!

```
1Ø CLS:PRINT@1Ø,"DRAGON PUZZLE 4"
2Ø PRINT@259,"FRANKINCENSE"
3Ø PRINT@328,"GOLD"
4Ø PRINT@165,"CAKE"
5Ø PRINT@360,"SNOW"
6Ø PRINT@198,"STAR"
7Ø PRINT@99,"MESSAGE"
8Ø PRINT@229,"POST"
9Ø PRINT@291,"PRESENT"
1ØØ PRINT@133,"PEARTREE"
11Ø FORI=3 TO 11:FOR J=6 TO 8 STEP 2
12Ø X=32*I+J:P=PEEK(X+1Ø24):
    PRINT@X,CHR$(P+32);:NEXTJ,I
13Ø PRINT@4ØØ,""
```

Errata — Dragon World Issue 1

There were, unfortunately some typesetting errors in the first issue of Dragon World. Whilst we contine to investigate alternative ways of presenting programs which are both error free and readable, perhaps you will accept our apologies together with the following list of errata.

Fortunately, most of the errors were easy to rectify given some experience but we understand how frustrating it can be for newcomers to be confronted with 'SN ERROR' etc. after painstaking typing sessions!

It may be helpful to review some of the common problems due to typesetting. Spaces often are a problem as they are not as wide as a character. Try to leave spaces between variable names and BASIC command words such as 'TO' as in 'FOR I=X TO N '. Also note that the space bar is often used as a control under INKEY$ and then it is important to distinguish between quotes around nothing and quotes around a space.

Here then is a list of the mistakes.

TORNADO

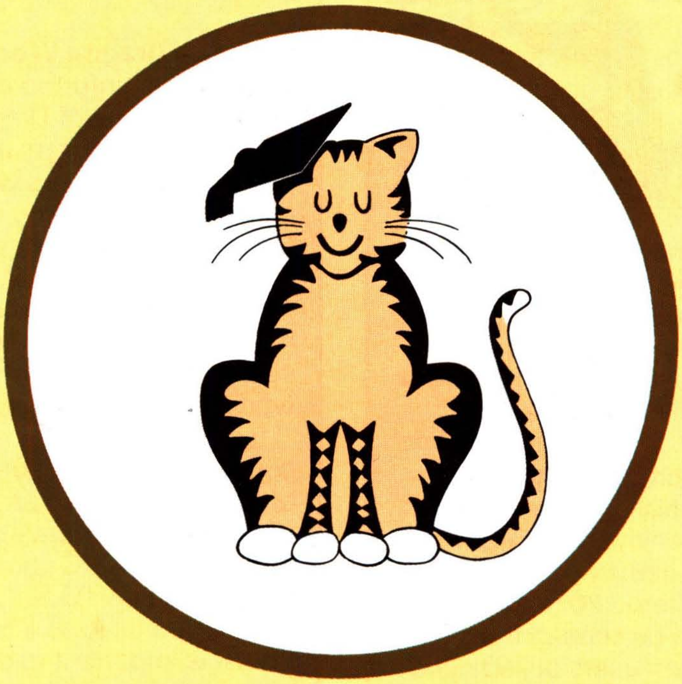
line 15Ø – a space between UI and THEN .
Line 2ØØ IF F<8 then F=8
Line 35Ø – the second semi-colon should be a colon
Line 38Ø – A final quote is required
Line 39Ø – The last two pairs of quotes should surround a space .

CHRISTMAS TREE

line 3ØØ – the second comma in PAINT(X,Y,) should not be present.

HOLES line 1ØØØ should start with PRINT@448,MID$(N$(I),1+2*N(I)):

CHESHIRE CAT
EDUCATIONAL SERIES
from
AMPALSOFT



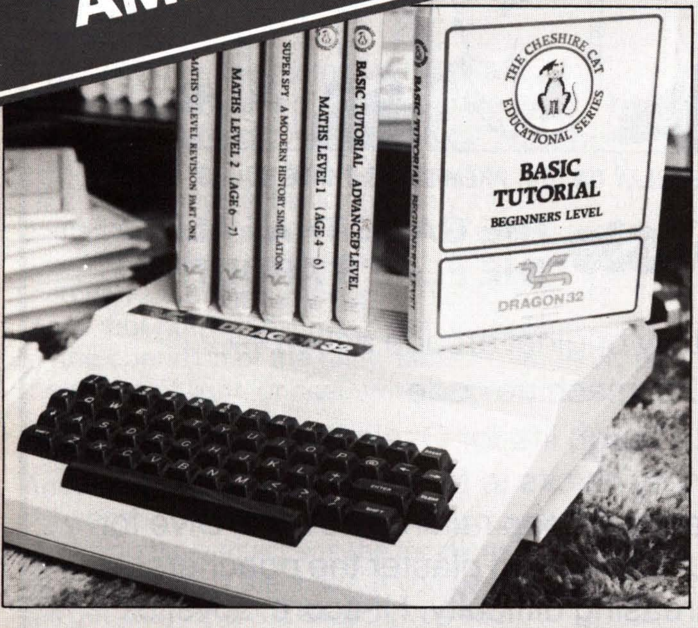
CHESHIRE CAT
The First name in Educational Software.

An exciting range of top quality programs covering all needs from pre-school to 'A' level.

Now available from Boots and other good stores.

Ampal Computer Services Ltd.
31 Woodbridge Road, Darby Green, Blackwater,
Camberley, Surrey.
Tel: (0252) 876677

AN INTRODUCTION TO AMPALSOFT



An exciting and rapidly expanding range of educational software for the Dragon 32 home computer is being produced by a novel co-operative of housewives and teachers.

This unlikely combination of talents has already shaken the software market with their high-quality, polished programs of genuine educational benefit, presented in the finest moulded plastic book-shaped packaging under a distinctive Cheshire Cat logo.

The mortar-board-sporting cat featured on the cover already commands pride of place in Boots' stores stocking computer software and other national retail chains and specialist software stores are showing a keen interest.

Yet 15 months ago Cheshire Cat was just the brainwave of three mothers with an active and enquiring collective brood of seven young children. Each family, all close friends, had bought Dragon 32 home computers because they realised the impact new technology would be having on their offspring's education and future employment prospects.

Naturally anxious to provide an early opportunity for their children to acquire some of the skills this micro chip revolution would require, they had sought the machine with the best keyboard for young fingers to operate. At the same time they also wanted to combine that with the largest memory, best colour, graphics and sound facilities available. At the price the Dragon 32 topped the poll.

But very soon the trio of housewives — Patricia Lansdowne, Ann Mortimer and Lynn Nixon, discovered there was a severe limit to the number of worthwhile educational programs readily available. Software shelves groaned under an avalanche of arcade-type games but genuine educational programs were rare. So Lynn, a skilled programmer, together with her two chums, had the bright idea of roping in a teacher friend to devise a bright, colourful and animated maths program suitable for their own four to six-year-olds.

Maths 1 was so good, neighbours and friends started clamouring for copies and Cheshire Cat was born.

Then a brilliant but simple Basic Tutorial program to teach beginners how to program their own Dragon 32 was produced — and that has now sold over 20,000 copies. "Basic Tutorial" has also featured in two major national advertising and promotional campaigns by Boots as part of a software pack sold with each Dragon 32.

Realising the tremendous potential this barren market place presented, the girls formed Ampalsoft at Knutsford, Cheshire, in October 1982, and devised their own Cheshire Cat logo and Ampalsoft trademark.

The talents of their husbands were swiftly drafted along with teachers and programming friends into the expanding co-operative fold. And it was the husbands who provided the sound business base on which Cheshire Cat's commercial success is firmly founded — Chris Lansdowne, 35, is a financial director; Colin Mortimer, 35, an electronics design engineer with a specialist interest in computers, and Tim Nixon is a top sales consultant. Now this multi-talented team includes more than 40 teachers and programmers working on new educational products for the Dragon 32 to add to the nine strong present range.

Cheshire Cat director, Patricia Lansdowne, credits their meteoric rise, with £1 million sales in their first year, to producing such top class products for a virgin market.

Patricia said: "We started Cheshire Cat to provide our own children with the educational software we could not buy. We have never cut corners on quality or detail and we never will. Each program takes up to four teachers and three programmers three months to complete and test. Our driving force is our children's future, not huge profits".

The ambitious target Cheshire Cat have set themselves is to eventually produce a comprehensive range of programs to provide every school subject from pre-school age ability right through to 'A' Level standard.

A wholly flexible program explores the 22 major areas of the current maths syllabuses of all the examination boards for 'O' level, CSE and Over 16 examinations.

Also available is the thrilling history simulation for 11-14 year-olds, "Superspy"; a first school reading aid, "Early Reading"; the top selling,

(continued on page 28)

BEHIND THE SCENES AT PEAKSOFT

WE DESPATCH BY POST UP TO THREE TIMES DAILY. AVERAGE DELAY ON OUR PREMISES IS THREE HOURS.

ossie THE GAME THAT ALL THE FLAP'S ABOUT!

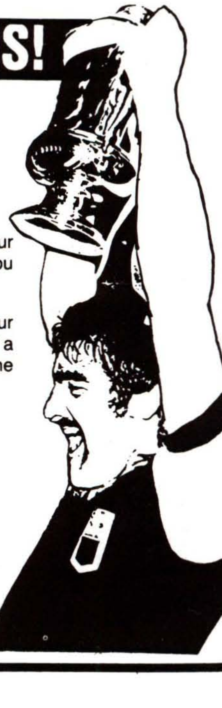
Totally original arcade action — 100% machine code

It's a tough life for Ossie the Osprey — five hungry chicks to feed, and those pesky poachers keep raiding the nest. Dive for your dinner and plaster the poachers! Increasing difficulty. Hi-score. Joystick optional. **£6.95**

PHOTO-FINISH: Superb new horse-race action. Terrific graphics, authentic race cards, odds based on actual form, REAL TIME races from 5f to 2 miles, 'Sporting Life' result cards with distances, times, etc, and an incredible photo-finish feature. 1-4 players. **£7.95**

£6.95 CHAMPIONS!

Take your team from the Fourth Division to the European Cup! Features four divisions, promotion, relegation, transfers, injuries, suspensions, weekly results and league tables, reserve team, crowd trouble, and all the fun, drama and frustration of managing a league club.



PEAKSOFT FREEPOST Burton-on-Trent DE15 9BR

£6.95

You're the last survivor - and it's a long way home. Can you survive the minefields, the helicopter gunships and the armed patrols? Who dares, wins! - Now with free lapel badge.

LIONHEART £5.45 — Unique, two-part romp. Arcade-style action as you assemble your army, pursued by the ruthless Prince John, switches to colourful graphics adventure as you land in the Holy Land to mount your Crusade against the Sultan Saladin.

DEATH'S HEAD HOLE £5.45 — The television crews are assembled at the entrance and your mother has sold her story to *The Sun*. The eyes of the nation are on you! Face the terrors of a Mendips pothole as you lead a mission of mercy to rescue a party of frightened cavers. The realism is astounding!

DON'T PANIC £5.45 — Two classic adventures at an incredible price. Features Towers of Death and The Ice Kingdom. Villains and hazards galore.

TEL (0283) 44904

Other departments and overseas orders:
7 Hawthorn Crescent
Burton-on-Trent
DE15 9QP

Yes, please rush me ....

....

Name ....

Address ....

....

DRAGON ANSWERS BACK

Question

I wish to use the motor control relay in my Dragon 32 by means of the cassette lead to perform switching operations on external devices. I find that using the 'MOTOR ON' and 'MOTOR OFF' commands in basic give only a tediously slow operation of the relay. You give the addresses for these operations as &H8015 and 8018 and I have tried POKES of all numbers from Ø to 255 into these, but nothing happens. Is it possible to speed up the operation of the relay, please, preferably to the speed of light, or near, with some simple POKES?

Answer

To speed up the operation of the cassette motor relay, you have to access the locations HEX 95 and 96 which control the delay. If you peek &H95, you will get 149, whereas &H96 gives 150.

If you POKE &H95 with Ø and &H96 with Ø1, it gives the optimum speed available.

The locations &H8015 and &H8018 just control the on/off routines and do not need to be altered.

Question

I have recently purchased a Dragon 32 and am very pleased with the operation of the BASIC and the quality of the software.

My major grievance is that you cannot, so I am lead to believe, mix graphics and text on the high-res screen.

Answer

It is possible to mix text and graphics in an alpha semi-graphics mode, PMODE 24, which gives a resolution of 192 x 64. Details can be obtained on request (please send sae).

Alternatively, you could type:

```
POKE &HFFCO,Ø
POKE &HFFC3,Ø
POKE &HFFC5,Ø
```

This will enable you to put text characters on the graphics screen starting at &H4ØØ, by poking the respective character codes at the desired locations.

Question

I recently purchased an Epson RS-80 printer to work in conjunction with my Dragon 32. Whilst the printer works perfectly using the commands:

```
LLIST
?#-2,"...."
?#-2,CHR$(-),
```

I am puzzled to find that there is no information in the printer manual on graphic screen dumps. Could you possibly give me any assistance in this matter?

Answer

We have available, free of charge, screen dumps for the Epson MX-100 and also for the two popular SEIKOSHA printers, the GP-100 and GP-250, one of which is included in this issue. The MX-100 dump should be easily adaptable for most Epson machines. These routines are available on request (please send sae).

Question

I am a radio amateur and am writing to ask if you market, or are aware of, a program which would allow my Dragon 32 to send and receive RTTY signals, hopefully on cartridge.

Answer

We are aware of a high quality RTTY program on cartridge and cassette — priced at £12.00 (cassette) or £21.00 (cartridge).

The program is written wholly in Machine Code and occupies approximately 4K of memory. The audio signals, (up to over 150 baud) are fed directly into the cassette input line. Alternatively an external terminal interface can be used for higher speed.

The program also includes a 4000 character "type-ahead" buffer which allows you to reply while receiving.

For full details apply to:
Mr. M. J. Kerry,
22 Grosvenor Road,
Seaford,
East Sussex
Tel: 0323 893378

INSTRUCTION COURSES

Dragon Data are considering offering the following courses:

2-Day 'DEALER SALES COURSES'
Cost: £260+VAT, price includes one night's accommodation and meals.
Aim: To help dealers maximise sales and service.

1-Day 'INTRODUCTION TO THE DRAGON'
Cost: £25
Aim: To enable new users to make the best use of their Dragon system and to become more conversant with BASIC.
Venues to be decided.

If you are interested in receiving further details, please write to Kathy Ashton at Dragon Data.

MACHINE CODE CORNER

This issue, we consider the problem of transferring the contents of a hi-res screen to a matrix dot printer. Most such printers these days can operate in 'graphics mode' whereby each byte of information sent to the printer is interpreted as a pattern of seven or eight vertical dots. A graphics screen can then be reproduced on paper by transforming the information content, pixel by pixel, into successive bytes, each byte to communicate to the printer a particular pattern of vertical dots.

Unfortunately, different printers require this information to be coded in different ways and so it is impossible to provide a single universal program for all printers. In this article we consider two popular printers — the Seikosha GP1ØØA (referred to as SK) and the Shinwa CP8Ø (denoted by CP).

First let's tackle the problem from BASIC, where communication to the printer is via the use of PRINT#-2, . A unit of information is one byte and this can be represented by any decimal number from Ø to 255. Thus PRINT#-2, CHR$(N); sends to the printer the byte whose binary form equals the decimal number N.

Some of the values of N are used by the printer as CONTROL CODES, and in particular, each printer uses a particular set of such codes to change to graphics mode. For SK, N=8 instructs the printer to accept all subsequent bytes as graphics bytes until the control code N=15 is sent. For CP the control codes are more involved and are N=27, followed by 49 (corresponding to "1") to control the line feed width, then N=27, followed by 1Ø7 ("K") plus two numbers L and M to instruct the printer to interpret the following L+256*M bytes as graphics bytes.

The differences between SK and CP do not stop here! SK uses a 7 bit column form and such a column whose first, third, fourth, sixth and seventh bits are 'set' would require the code 128 (always present) +1+4+8+32+64=237. In other words as we descend the column of dots we add to 128 a power of two for every dot to be printed using the powers 1,2,4,8,16,32,64 with 1 associated with the top bit and 64 with the bottom.

On the other hand CP uses an eight bit column with the top bit associated with 128, and then down through 64,32,16,8,4,2 and 1 for the bottom bit!

First then we construct a program for SK assuming that our graphics picture has been created in PMODE4 with black on a buff background. The pixel at X,Y is set to black if PPOINT(X,Y)=Ø. We construct a program that scans down each column of seven pixels starting with Y co-ordinate J=Ø increasing to 189 in steps of 7, and X co-ordinate I=Ø to 255, the column being scanned by variable K going from Ø to L (where L is normally 6 except for the last line). The summation of the N value corresponding to each column configuration is done in the K loop using P=P-(PPOINT(I,J+K)=Ø)*P(K). Note that we have taken the trouble to place the powers of 2 in an array — this is not just for neat programming but an important necessity where considerations of speed are relevant (see the article on DATA statements in the first issue of DRAGON WORLD). The subroutine is as follows.

```
1 TE=128:FOR I=1 TO 7 :READ P(I):NEXTI
2 DATA 1,2,4,8,16,32,64
1Ø PMODE4,1:SCREEN1,1:PCLS1:COLORØ,1:
2Ø LINE(Ø,Ø)–(255,191),PSET,B
1ØØ PRINT#–2,CHR$(8):L=6:
    FOR J=Ø TO 189 STEP 7
2ØØ IF J=189 THEN L=2
3ØØ FOR I=Ø TO 255:P=TE:FOR K=Ø TO L
4ØØ P=P–(PPOINT(I,J+K)=Ø)*P(K):NEXTK
5ØØ PRINT#–2,CHR$(P);
    :NEXTI:PRINT#–2:NEXTJ
6ØØ PRINT#–2,CHR$(15)
```

Since CP uses an 8 bit graphics code a different program can be written for CP using the graphics bytes as stored inside Dragon providing we agree to represent the screen on paper sidewards. To be precise, the first row of the printed picture will correspond to the left-most vertical strip of the screen. If we again use PMODE4, but this time printing BUFF on BLACK then each byte of the graphics memory corresponds exactly to the information required by CP. The program uses Dragon's own store of the address of the start of the current graphics screen (bytes Hex BA and BB).

```
1Ø PMODE4,1:SCREEN1,1:PCLS
2Ø LINE(Ø,Ø)–(255,191),PSET,B
```

```
1ØØ ST=256*PEEK(&HBA)+PEEK(&HBB)
11Ø PRINT#–2,CHR$(27);"1";
12Ø FOR I=Ø TO 31:PRINT#–2,CHR$(27);"K";
    CHR$(192);CHR$(Ø);
13Ø FOR J=191 TO Ø STEP –1:
    PRINT#–2,CHR$(PEEK(ST+32*J+I));
    :NEXTJ
14Ø PRINT#–2,CHR$(13);:NEXTI
15Ø PRINT#–2,CHR$(27);"2"
```

And so at long last to Machine code! We look in detail at the construction of a machine code program for CP and present a BASIC program to POKE in the equivalent program for SK. Since the speed of execution of these programs is now determined by the speed of the printer itself and not by the program design, both construct the screen dump horizontally and assume a black on buff format in PMODE4.

We need to be able to send bytes of information to the printer from machine code — this is done by loading A with the appropriate byte and then the instruction JSR $8ØØF uses a subroutine in ROM that sends the byte in A to the printer. We use this procedure for both control codes and graphics bytes.

The basic architecture of the program is constructed by considering those eight bytes in RAM corresponding to a current rectangle of the screen consisting of 8 by 8 pixels. These eight bytes will occur in RAM at the positions 'start', 'start'+32, 'start'+64 and so on. To assemble a byte of information the first bit of all the eight bytes must be interrogated, then the second bit and so on. When all eight bits have been assembled we move on to the next set of 8 bytes corresponding to 8 vertical pixels adjacent to the previous 8 and after 32 such sets we will have assembled 8*32=256 bytes of information corresponding to a complete horizontal strip of the screen 8 pixels deep. These 256 bytes are stored in RAM from hex 7FØØ to 7FFF.

```
1Ø          ORG    $7DØØ
2Ø          PUT    $7DØØ
3Ø   ST1    EQU    $7E2Ø
4Ø   ST2    EQU    $7E24
5Ø          LDA    #$1B      861B
6Ø          JSR    $8ØØF     BD8ØØF
7Ø          LDA    #49       8631
8Ø          JSR    $8ØØF     BD8ØØF
9Ø          LDD    $BA       DCBA
1ØØLOOP     STD    ST2       FD7E24
11Ø         TFR    D,X       1FØ1
12Ø LOOPØ   LDY    #$7FØØ    1Ø8E7FØØ
13Ø LOOP1   TFR    X,D       1F1Ø
14Ø         ADDD   #225      C3ØØE1
15Ø         STD    ST1       FD7E2Ø
16Ø         LDA    #$8Ø      868Ø
17Ø LOOP2   CLRB             5F
18Ø         TFR    X,U       1F13
19Ø LOOP3   LSLB             58
2ØØ         BITA   ,U        A5C4
21Ø         BNE    CONT      26Ø1
22Ø         INCB             5C
23Ø CONT    LEAU   32,U      33C82Ø
24Ø         CMPU   ST1       11B37E2Ø
25Ø         BLO    LOOP3     25F1
26Ø         STB    ,Y+       E7AØ
27Ø         CMPY   #$8ØØØ    1Ø8C8ØØØ
28Ø         BEQ    CHECK     27ØB
29Ø         CMPA   #$Ø1      81Ø1
3ØØ         BEQ    NEXT      27Ø3
31Ø         LSRA             44
32Ø         BRA    LOOP2     2ØDF
33Ø NEXT    LEAX   1,X       3ØØ1
34Ø         BRA    LOOP1     2ØD1
35Ø CHECK   LDY    #$7FØØ    1Ø8E7FØØ
36Ø LOOPC   LDA    ,Y+       A6AØ
37Ø         BNE    FOUND     26Ø8
38Ø         CMPY   #$8ØØØ    1Ø8C8ØØØ
39Ø         BLO    LOOPC     25F6
4ØØ         BRA    INC       2Ø21
41Ø FOUND   LDY    #$7FØØ    1Ø8E7FØØ
42Ø         LDA    #$1B      861B
43Ø         JSR    $8ØØF     BD8ØØF
44Ø         LDA    #$4B      864B
45Ø         JSR    $8ØØF     BD8ØØF
46Ø         CLRA             4F
47Ø         JSR    $8ØØF     BD8ØØF
48Ø         INCA             4C
49Ø         JSR    $8ØØF     BD8ØØF
5ØØLOOPF    LDA    ,Y+       A6AØ
51Ø         JSR    $8ØØF     BD8ØØF
52Ø         CMPY   #$8ØØØ    1Ø8C8ØØØ
53Ø         BLO    LOOPF     25F5
54Ø INC     LDA    #$ØD      86ØD
55Ø         JSR    $8ØØF     BD8ØØF
56Ø         LDD    ST2       FC7E24
57Ø         INCA             4C
58Ø         CMPD   $B7       1Ø93B7
59Ø         LBLO   LOOP      1Ø25FF87
6ØØ         LDA    #$1B      861B
61Ø         JSR    $8ØØF     BD8ØØF
62Ø         LDA    #$32      8632
63Ø         JSR    $8ØØF     BD8ØØF
64Ø         RTS              39
```

The finer details are as follows. ST2 is used to store the address of the first byte of a strip, and is initially supplied with the address from store $BA (lines 9Ø, 1ØØ). This is incremented by 256 before returning to 'LOOP' by loading D from ST2 and incrementing A (as A is the most significant byte of D). Given the address X of a current set of 8 bytes, U is used to control looping through the 8 bytes (hence line 23Ø increments U by 32) and ST1 is used to indicate when this procedure should stop by setting ST1 to X+225 (lines 13Ø-15Ø).

To access the individual bits of our current set of bytes, we use A to provide a 'mask' so that for example to examine the most significant (left-most) bits A is set to hex 8Ø (binary 1ØØØØØØØ) (line 16Ø) and subsequently changed to the other masks using LSRA (line 31Ø). LOOP2 controls this process with branching to NEXT when A=$Ø1 (mask ØØØØØØØ1).

To assemble the graphics information B is first set to zero. The command BITA ,U (line 2ØØ) checks to see whether the appropriate bit (determined by A) of the current byte (determined by U) is set — if so 1 is added to B. On re-entry to LOOP3, LSLB shifts this information one place to the left so that the first bit (highest on the screen) eventually ends up as the most significant bit. When a byte of information is complete it is stored in address Y and Y is then incremented. A check for the end of a strip is then possible by comparing Y with $8ØØØ.

Note finally that lines 350-390 provide a check to see whether the current strip is blank or not — if so all that is required is a line feed, otherwise the whole strip is printed by loading each byte from

(continued on page 23)

YOUNG USER'S PAGES

MOVING STRINGS

Welcome to all our new readers. Dragons will have appeared in many homes over Christmas and many of you will be programming for the first time. On these pages we offer you short programs which help you explore the possibilities of your Dragon. We usually concentrate on one or two commands and this week, for the new readers, we look at the PRINT@ command but then we will use it to make pictures that move. All the commands used are explained in detail in the programming book which came with your Dragon. Before you make something move you must be able to show it on the screen. I've chosen to use the PRINT screen. What can you print on this screen? Well if you look at pages 136-138 in the programming book you will see a list of characters and their code numbers. There are letters, numbers, punctuation marks and some special shapes which can be printed in different colours. If you haven't experimented with these yet just type:

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

then press <ENTER>. A letter A should appear below the command line. If you want to print it in a different place you use the PRINT@ command. Type

```
PRINT@272,CHR$(65)
```

When you enter this the A will appear in the middle of the screen at position 272. There are 512 positions numbered from Ø to 511. You can refer to page 14Ø of the programming book to find the positions on the grid there. The characters are numbered from Ø to 255. Some of the codes don't actually print anything — some give instructions for printing like 'go to the next line' or 'go back a space'. The characters which have codes above 127 are black shapes on a coloured ground. If you add 16 to a code you get the same shape with a different background. I'm going to use these characters to make a train.

First the carriage: CHR$(129) is shaped like this ▘ and CHR$(131) like this ▀. We will write them as CHR$(129+N) and CHR$(131+N) then we can change the value of N until we have a colour we like. We can put the characters together to form a STRING and give the STRING a name. A string is just some characters tied together and a name allows us to refer to them without writing the whole string. A string variable must have a name which ends with $. I have called this C$,C for carriage. Try this tiny program.

```
1Ø CLS6
5Ø C$=CHR$(129+N)+CHR$(131+N)
1ØØ PRINT@1ØØ,C$
GO TO 1Ø
```

There is C$. But the program stops and you get OK on the screen. To prevent this type in a line which sends the program round in circles until you press the <BREAK> key:

```
11Ø GO TO 11Ø
N=32:GOTO1Ø
```

Line 11Ø goes round in circles and the next line gives N a value and starts the program off. See the colour change? Use <BREAK> to stop the program.

Three C$s in a row look like three windows but we need a CHR$(130+N) to finish it off and something for a coupling. I chose an equals sign but you may prefer an aitch. Now type these lines:

```
6Ø C$=C$+C$+C$+CHR$(13Ø+N)+"="
N=112:GO TO 1Ø
```

there — a carriage top with orange windows.

As you can see, if the character is a keyboard symbol like "equals" we can add it to the string in quotes. In fact CHR$(61) and "=" are equivalent.

What about the base of the carriage? How can we make the wheels? Well there are quite a few round things on the keyboard, Ø,O,* and @. I chose @ and I've filled in the gaps with =s so that they can't get lost in printing! (But you'll see later that they have another purpose.) We can make a string, B$, for the base using the quote marks, and we can print it under the carriage — that means starting at a position 32 more. Add these lines:

```
7Ø B$="@@===@@="
11Ø PRINT@132,B$
N=64:GO TO 1Ø
```

To get the carriage moving we should have an engine, but I'll come to that later. For now let's get that carriage to move across the screen. To do this we must print it at a different positions. If we call the print position for the carriage top P and the position for the base P+32 we can alter P in a loop. We will overwrite lines 1ØØ and 11Ø. At the same time we can sort out another problem. If you RUN your program now you will see that the whole line to the right of the carriage turns green. If we finish the PRINT command with a semi-colon this will stop. Here are the new lines:

```
1ØØ FOR P=1ØØ TO 12Ø
11Ø PRINT@P,C$;:PRINT@P+32,B$;
12Ø NEXT
RUN
```

Well it moved but grew as well because we printed each carriage on top of the last without rubbing it out. We can get over this by putting a blank character behind the carriage when we print so that it rubs out as it goes along. The new line 11Ø is:

```
11Ø PRINT@P,"   "+C$;:PRINT@P+32,
        "   "+B$;
```

Make sure you have a blank space inside the quotes. Now RUN. Success at last!

What about the engine? Well I'm sure you can design one for yourself but here are the lines for mine:

```
8Ø E$=CHR$(136)+STRING$ (5,128)
   +CHR$(123+N)
9Ø BE$="@@@@@@@"
```

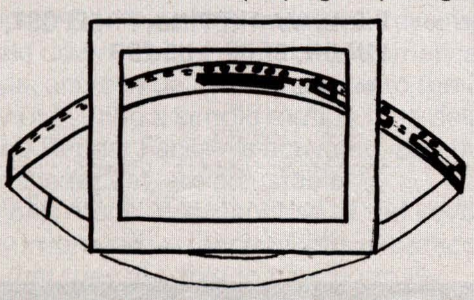
There I introduced a new command, STRING$(5,125). This made a STRING of five black rectangles — the first number is the length of the STRING and the second number is the character code. (That second number could be a one character variable or the character itself enclosed in quotes.)

Now you've seen how to make a STRING move you can put several carriages and the engine into one STRING and make them all move.

I decided to make my train go right across the screen and then come back on again as though it was going round and round on a track. So first I made a string of four carriages and an engine. Then I added a lot of blanks, CHR$(143) to the beginning. In fact there are 57 blanks so the whole STRING is 96 characters long. Then I did the same for the base with 57 =s for the track. (That was the reason for the =s between the wheels of the carriage.) It's not a perfect track — Zs or Is might be better.

```
1ØØ TT$=C$+C$+C$+
     C$+E$+STRING$(57,128)
11Ø TB$=B$+B$+B$+B$+BE$+STRING$(57,61)
```

I don't want to print all that at once. Each string is 96 characters long and I want only 32 at a time. I can cut these out using MID$. This is a very useful command which lets you chop bits out of a STRING. It goes MID$(string,start,length). For instance MID$("ABCDEFG",2,3) cuts out the string "BCD". If you don't give the length, all of the string to the right of the starting point is chopped out. I always print the last 32 characters but make a new string by chopping the last character off and putting it at the front. It's as though the string were printed on a strip of paper which is being pulled across the screen but the end has been glued to the beginning. This is a useful technique for those games to teach frogs the Green Cross Code. Why did the chicken cross the road? Because it was playing leap-frog!



```
16Ø PRINT@32Ø,MID$(TT$,65);
   : PRINT@352,MID$(TB$,65);
18Ø TT$=MID$(TT$,96)+MID$(TT$,1,95)
19Ø TB$=MID$(TB$,96)+MID$(TB$,1,95)
2ØØ GO TO 16Ø
```

The train will go round and round. To make it more interesting I've put in a signal which is controlled by the space bar. If the signal is red when the train approaches it stops. The variable F flips between −1 and +1 whenever the space bar is pressed. (Variables like this are often referred to by programmers as flags — how appropriate here!)

Whether you travelled with me through this page or have taken a short cut to the end, here is the complete listing of the train with the signal.

```
Ø REM      TRAIN : DEC 83
1Ø CLEAR1ØØØ:C=112:CLS6:F=1
2Ø F$=CHR$(134+16)
3Ø FOR I=249 TO 345 STEP32:
   PRINT@I,F$;:NEXT
4Ø S$=CHR$(188):G$=CHR$(131)
5Ø C$=CHR$(129+C)+CHR$(131+C)
6Ø C$=C$+C$+C$+CHR$(13Ø+C)+"="
7Ø B$="@@===@@="
8Ø E$=CHR$(136)+STRING$ (5,128)
   +CHR$(132+C)
9Ø BE$="@@@@@@@"
1ØØ TT$=C$+C$+C$+C$+E$+
    STRING$(57,143)
11Ø TB$=B$+B$+B$+B$+BE$
    +STRING$(57,61)
12Ø K$=INKEY$:IF K$=CHR$(32)THENF=F*-1
13Ø FL$=S$:IF F=1 THEN FL$=G$
14Ø PRINT@217,FL$;:IF F=1 THEN16Ø
15Ø IF T=45 THEN 12Ø ELSE 16Ø
16Ø PRINT@320,MID$(TT$,65);
    :PRINT@352,MID$(TB$,65);
17Ø T=T+1:IF T=96 THEN T=Ø
18Ø TT$=MID$(TT$,96)+MID$(TT$,1,95)
19Ø TB$=MID$(TB$,96)+MID$(TB$,1,95)
2ØØ GO TO 12Ø
```

'HAPPY BIRTHDAY' OFFER

Many thanks to all you Dragon owners who have bought the 'Happy Birthday Dragon' offer over the last few weeks.

It has come to our attention, however, that one or two of you have encountered problems with this cassette, one being with the game 'Hoppy'. In order to restart the game after the third frog has been run over, it is necessary to press the spacebar on the keyboard.

The other problem that has come to our attention is that if you try to load 'Santa Laverna' or 'Meson Raid' without first clearing the memory of 'Hoppy', this can cause a number of errors to appear while attempting to run the programs. Therefore, if you want to run either 'Santa Laverna' or 'Meson Raid', then switch off your Dragon to clear the memory, type in CLOAD 'Laverna or CLOAD 'Meson'. This will allow the programs to load without any difficulty.

DRAGON 64: LOW-COST PROFESSIONAL COMPUTING

With the launch of the Dragon 64, together with the OS9 Operating System, we have our passport to the world of truly professional computing. The price of the computer itself is less than is normally paid for the software it supports — so Dragon Data has slashed the prices of several well-known professional packages for use on the new machine.

In appearance, the 64 looks just like the 32, except that it is grey in colour. All the familiar input/output ports are there, with the addition of one marked S.I/O — this is the RS232 (serial) interface.

When you turn on the power you still find little difference between the 64 and the 32. In fact, the command ?MEM gives the usual value of 24871. This is because a great deal of care has been taken to make sure that most software for the Dragon32 can also be run on the Dragon64. So on power-up we enter a "32K mode". To move to the 64K mode, we need to type "EXEC" (or EXEC48ØØØ). A blue cursor reminds us that we are in 64K mode, and the command ?MEM now gives 41241. (The commands CLEAR and PCLEAR can, of course be used to increase this value.) The whole 64K of memory is now available as RAM — a re-assembled version of the BASIC interpreter resides in the top 16K, but this can be over-written if it is not required.

Apart from the obvious advantage of extra memory (16K if you are writing in BASIC, and 32K for machine code users) there are several other new features. Perhaps the most important is the RS232 interface, which allows communication with serial devices such as printers and graph-plotters or with other computers. New commands of the form DLOAD and DLOADM make it possible to use this port as easily as the tape interface, but it is also possible to send and receive single characters, using simple routines described in the "DRAGON64 Supplement" manual.

Another useful modification is the Keyboard Auto-Repeat Facility which makes all keys repeat when held down. This is available automatically in 64K mode, and can be incorporated into the 32K mode by executing a simple routine. The delay before the Repeat Facility is activated, and the speed of the repeat, are adjustable.

In the 64K mode, the response of the keyboard is greatly improved — touch typists in particular will find that they are able to type normally, without such problems as "PRINT" coming out as "PINT"!

Searching tapes for files is made considerably easier, in both 32K and 64K modes, by the fact that SKIPF, CLOAD and CLOADM do not give IO ERRORs when the tape is started in the middle of a file. Instead, the beginning of the next file is found before any attempt is made to load or skip.

A minor enhancement is that the reverse slash "\" is available directly from the keyboard, using the SHIFT and CLEAR keys together.

The DRAGON32's non-standard form of USR function operation is brought into line with normal Microsoft BASIC. With a 32, the USR function will default to USRØ unless an expanded form such as USRØ1 is used (when the function is called, not at the definition stage). This is "put right" in the 64, so that the usual form of the statement (e.g. X=USR1(5)) can be used. Unfortunately the form USRØ1 is now illegal, and so is a minor incompatibility between 32 and 64 — easily remedied by deleting the unwanted "Ø".

No "reverse BOOT" is available to return from 64K mode to 32K mode, but a cold-start can be forced by POKEing a Ø into address 113 and pressing RESET. This results in a return to 32K mode, but also "NEWs" any BASIC program in memory. However, the contents of the lower 32K of RAM are preserved.

No cartridges can be used in the 64K mode, although they will work as usual in the 32K mode. This means in particular that the DRAGONDOS Disk Operating System cannot by used in 64K mode. DRAGONDOS can, however, be used to BOOT a more powerful Operating System — OS9.

Its ability to support OS9 takes DRAGON64 out of the realm of "mere home computers" into the world of serious business computers. OS9 is a UNIX type Operating System, which is both versatile and easy to learn and use. With it you can load a variety of languages — BASICØ9 (a very much more powerful BASIC, with a large number of extra commands), PASCAL and "C" are already available. A 51 column by 24 row text screen is provided for your own programming, or for use with the professional packages, such as STYLOGRAPH (an advanced, but simple-to-use, Wordprocessor), RMS (a complete Record Management System), and DYNACALC (a computerised spreadsheet). OS9 and the packages it supports will be reviewed more fully in future editions of DRAGON WORLD.

What has the DRAGON64 to offer the home user — without disk drive and sophisticated operating system? If your main interest is in playing COSMIC INVADERS or BERSERK, then the answer is: not much. But if you want to make a little more practical use of your computer, then the extra 16K available under BASIC control can be very handy. A typical area in which 32K seems to get eaten up all too quickly is the Database Retrieval System.

A Database is the computerised equivalent of a manual "card index". Each Database contains a number of RECORDS (cards) and every record has entries in each of a number of FIELDS. For example, an address/telephone list will contain a record for each person. The first field could be the surname, the second field the forenames, the third field the address, the fourth field the postcode, and the fifth field the telephone number. A record may be retrieved by specifying any of the fields — normally we would probably specify the surname, to obtain the address and telephone number, but we could specify the telephone number to discover whose number it is!

The following BASIC program is a simple Database — for the sake of brevity it contains only a small number of options, but it can be extended into a more advanced system.

```
1 REM DATABASE
2 REM A.D.MAYER, 1984
1Ø PCLEAR1:CLEAR19ØØØ:B=185ØØ:
    C=16ØØ
2Ø DIMB$(1,5)
3Ø DATACREATE DATABASE,
    LOAD DATABASE,ADD RECORDS,
    DELETE RECORD,FIND RECORD,
    KILL DATABASE,SAVE DATABASE
4Ø S$(Ø)="CL":S$(1)="ADFKS":
    N(Ø)=2:N(1)=5
5Ø FORL=ØTO1:FORJ=1TON(L)
6Ø READB$(L,J):NEXTJ,L
7Ø L=Ø:GOSUB1ØØØ
8Ø ONX GOTO2ØØ,3ØØ
9Ø L=1:GOSUB1ØØØ
1ØØ ONX GOTO4ØØ,5ØØ,6ØØ,7ØØ,8ØØ
2ØØ GOSUB12ØØ:
    INPUT"NUMBER OF FIELDS";F
21Ø INPUT"AVERAGE FIELD LENGTH";AV
22Ø GOSUB13ØØ:N=Ø
23Ø FORJ=1TOF
24Ø PRINT"NAME OF FIELD";J:
    INPUTF$(J-1):NEXT
25Ø GOTO4ØØ
3ØØ GOSUB12ØØ
31Ø GOSUB14ØØ:OPEN"I",#-1,N$
32Ø INPUT#-1,F,AV,N:GOSUB13ØØ
33Ø FORJ=ØTOF-1:INPUT#-1,F$(J)
34Ø FORK=ØTON-1:
    INPUT#-1,A$(J,K):NEXTK,J
35Ø CLOSE#-1:GOTO9Ø
4ØØ IFN=NZ THEN17ØØELSEN=N+1:
    CLS:PRINT"RECORD NUMBER";N:J=Ø
41Ø PRINTF$(J)
42Ø INPUTX$:IFX$=""THEN44Ø
43Ø A$(J,N-1)=X$:J=J+1:IFJ<F
    THEN41ØELSE4ØØ
44Ø N=N-1:GOTO9Ø
5ØØ J=Ø:GOSUB16ØØ
51Ø GOSUB63Ø:IFJ=N THEN9Ø
52Ø GOSUB15ØØ:PRINT
"THIS RECORD?":GOSUB11ØØ
53Ø IFX$><"Y"THEN58Ø
54Ø N=N-1:J=J-1
55Ø IFJ=N THEN9Ø
56Ø FORK=ØTOF-1:A$(K,J)=A$(K,J+1)
57Ø NEXTK:J=J+1:GOTO55Ø
58Ø IFJ=N THEN9ØELSE51Ø
6ØØ J=Ø:GOSUB16ØØ:FL=Ø
61Ø GOSUB63Ø:IFJ=N THEN9Ø
62Ø GOSUB15ØØ:GOSUB11ØØ:IFJ<N
    THEN61ØELSE9Ø
63Ø IFT$=A$(FS-1,J)THENRETURN
64Ø J=J+1:IFJ<N THEN63Ø
65Ø IFFL=ØTHENPRINT"NOT FOUND":
    GOSUB11ØØ
66Ø RETURN
7ØØ PRINT"KILL?":GOSUB11ØØ:
    IFX$><"Y"THEN9ØELSERUN
8ØØ GOSUB14ØØ:OPEN"O",#-1,N$
81Ø PRINT#-1,F,AV,N
82Ø FORJ=ØTOF-1:PRINT#-1,F$(J)
83Ø FORK=ØTON-1:PRINT#-1,
    A$(J,K):NEXTK,J
84Ø CLOSE#-1:GOTO9Ø
1ØØØ CLS:FORJ=1TON(L):
    PRINTTAB(5);MID$(S$(L),J,1);
    TAB(1Ø);B$(L,J):NEXT
1Ø1Ø GOSUB11ØØ:X=INSTR(1,S$(L),X$):
    IFX=ØTHEN1Ø1ØELSERETURN
11ØØ X$=INKEY$:
    IFX$=""THEN11ØØELSERETURN
12ØØ INPUT"NAME OF DATABASE";
    N$:IFLEN(N$)<9THENRETURN
121Ø PRINT"TOO LONG":GOTO12ØØ
13ØØ NZ=INT(B/F/AV)-1
131Ø IFF*(NZ+2)>C THENNZ=
    INT(C/F)-2
132Ø DIMF$(F-1),A$(F-1,NZ):RETURN
14ØØ PRINT"PRESS SPACE WHEN READY"
141Ø GOSUB11ØØ:IFX$><"  "
    THEN 141ØELSERETURN
15ØØ FL=1:CLS:FORK=ØTOF-1:
    PRINTF$(K)":   "A$(K,J)
151Ø IFPEEK(136)*256+PEEK
    (137)<1472THEN1530
152Ø GOSUB11ØØ:CLS
153Ø NEXTK:J=J+1:RETURN
16ØØ INPUT"FIELD NUMBER";FS
161Ø INPUT"TARGET";T$:RETURN
17ØØ PRINT"DATABASE FULL":
    GOSUB11ØØ:GOTO9Ø
```

As it stands, this program will work with a DRAGON32 or a DRAGON64 in 32K mode.

Line 1Ø PCLEARs to 1 graphics page (the minimum allowed without "POKEing" to PCLEARØ), and then distributes the remaining memory between string and non-string categories, in an attempt to make best use of it. Lines 2Ø-6Ø set up strings for the menus. Lines 7Ø-8Ø and 9Ø-1ØØ display the menus and use subroutine 1ØØØ to interpret the user's response — line 1Ø1Ø uses INSTR on the appropriate S$ to decipher the response.

(continued on page 23)

(continued from page 20)

The first options are to Create a (new) Database or to Load an existing one. In either case, a name must be specified for the Database. If a Database is loaded, the program proceeds directly to the second menu. If the Create option is used, an average field length must be estimated, names must be given for all fields and then the data are entered. "End of data" is indicated by pressing <ENTER> on its own.

The second menu allows the Database to be Saved (on tape) — the file will be given the same name as the Database. Records may be Added — effectively a return to the "Create" mode. The Database may be interrogated, using the Find option — a field number is specified, then the particular record in that field (in our address/telephone Database, field number 1, followed by the surname of the particular person, for example). A record may be Deleted — it must first be "found" using a Find-type sequence. Finally, the Database may be "Killed", usually only after Saving, to allow a new Database to be Loaded or Created.

To convert to 64K mode, only line 1Ø need be altered. The maximum number of records allowed is calculated automatically by subroutine 13ØØ and is controlled by the parameter values in line 1Ø. In 32K mode, after the program is loaded, there are about 275ØØ bytes available for data. Each record requires 5 bytes of non-string memory (essentially a pointer to indicate where the record is to be found in RAM) and the relevant number of bytes (equal to its length) in string memory. If a reasonably accurate "average field length" is given, the program will calculate the number of records allowed and this number should nearly fill both memory areas. The numbers specified assume an average string length of about 12 for maximum efficiency. B is the number of bytes of string memory (reduced by 5ØØ bytes to allow for any slight underestimate of the average, and C is the total number of records*fields allowed — this is limited by the non-string memory available. So B+5*C must not exceed the available memory (275ØØ bytes). In this case, B+5*C=265ØØ, which allows 5ØØ bytes to spare in both types of memory. In 64K mode, however, the available memory increases to 42ØØØ bytes. So we need B and C to satisfy B+5*C=41ØØØ to allow the same margin of error. B=29ØØØ, C=24ØØ are suitable values, so line 1Ø should be replaced by

```
1Ø PCLEAR1:CLEAR295ØØ:B=29ØØØ:C=2400
```

To obtain the optimum values of B and C for a particular average length, AV, use the formulae

```
C = 265ØØ/(AV+5)
B = 265ØØ–5*C
```

(For 64K mode, replace 265ØØ by 41ØØØ.)

(continued from page 15)

$7FØØ to $7FFF into A and thence to the printer.

The assembly of this program was achieved using ALLDREAM situated in RAM from 23ØØØ onwards (see elsewhere in this issue for relevant details) and the listings were produced using DREAM and DREAMBUG.

If you wish to implement this program from BASIC then you must construct a BASIC program similar to the one below (which gives the equivalent version for the Seikosha). To do this, change line 1Ø to FOR I=&H7DØØ to &H7D8F, and then supply DATA statements using the listing of bytes from the assembly listing (i.e. bytes 86,1B,BD,8Ø,ØF,....39). Once run you may record such programs using CSAVEM or the equivalent DOS command.

Finally note that to use the programs, you must first make a CLEAR command such as CLEAR2ØØ,&H7DØØ, and then invoke the program with EXEC&H7DØØ.

```
5 REM MACHINE CODE SCREEN DUMP FOR
  SEIKOSHA GP1ØØA
1Ø FOR I=&H7DØØ TO &H7D89
2Ø READ X$:POKE I,VAL("&H"+X$):NEXTI
3Ø DATA 9E,BA,1F,1Ø,FD,7E,24,1F,1,1Ø
4Ø DATA 8E,7F,Ø,1F,1Ø,C3,Ø,C1,FD,7E
5Ø DATA 2Ø,86,8Ø,C6,8Ø,1F,13,7F,7E,22
6Ø DATA 7C,7E,22,11,93,B7,24,13,A5,C4
7Ø DATA 26,3,FB,7E,22,78,7E,22,33,C8
8Ø DATA 2Ø,11,B3,7E,2Ø,25,E8,E7,AØ,1Ø
9Ø DATA 8C,8Ø,Ø,27,B,81,1,27,3,44
1ØØ DATA 2Ø,CF,3Ø,1,2Ø,C1,1Ø,8E,7F,Ø
11Ø DATA A6,AØ,81,8Ø,26,8,1Ø,8C,8Ø,Ø
12Ø DATA 25,F4,2Ø,16,1Ø,8E,7F,Ø,86,8
13Ø DATA BD,8Ø,F,A6,AØ,BD,8Ø,F,1Ø,8C
14Ø DATA 8Ø,Ø,25,F5,2Ø,Ø,86,D,BD,8Ø
15Ø DATA F,FC,7E,24,C3,Ø,EØ,1Ø,93,B7
16Ø DATA 25,8Ø,86,F,BD,8Ø,F,39
```

ELITE* CALC

Features include

* 255 MAXIMUM NO. OF ROWS
* 255 MAXIMUM NO. OF COLUMNS
* VERY EASY TO USE
* INDIVIDUAL CELL FORMULAE
* COPY BLOCKS OF CELLS
* FULL CELL-EDIT CAPABILITY
* COMPATIBLE WITH ALL PRINTERS
* EASY 132 COLUMN PAGE WIDTH
* GRAPH FORMAT FOR BAR CHARTS
* COMPREHENSIVE MANUAL INCLUDED
* INTERNATIONAL USER GROUP
* SORT ROUTINE IN ASCENDING OR DESCENDING ORDER

* Single character commands
* Help display
* Enter text or formulae to 255 characters long
* Repeat text entries
* Available memory always displayed
* Rapid entry modes for text and data
* Selectable automatic cursor movement
* Insert, Delete, Move entire rows or columns
* Replicate one cell to fill a row or column with selectable formula adjustment
* All machine language for blistering speed
* >24K data storage space available in 32K systems.
* Basic style formulae
* Arithmetic operators: +, –, *, /, , (,), =
* Relational operators: =, >, <, <=, >=, <>
* Logical operators: AND, OR, NOT
* Conditional formulae: IF... THEN... ELSE
* Trig. functions: SIN, COS, TAN, ATN
* Log. Functions: LOG, EXP, SQR
* Misc. functions: INT, FX, ABS, SGN
* Range functions: SUM, AVERAGE, COUNT, MIN, MAX, LOOKUP
* Nine digit precision
* User definable constant table
* User definable printer set-up commands
* Individual column width settings (1 to 255)
* Adjustable row height to insert blank lines without wasting memory
* Hide columns or rows
* Alternate print font selectable cell by cell
* Display/Print formats set by cell, row or column
* £-format, comma grouping, prefix or postfix signs
* Scientific notation, fixed point and integer formats
* Left or Right cell contents justification
* Full page formatting
* All formats stored with worksheet on disc/tape
* Save/load disc/tape files in compact memory form
* Scan disc directories
* Output ASCII file for word processor input compatability
* Memory resident code... no repeated disc calls

*NOW... The worksheet calculator program you've been waiting for is waiting to work for you. ELITE*CALC is a powerful, full featured spreadsheet for the Dragon 32/64 and Tandy Color Computer. Answer 'what if...' questions, prepare reports and cash flow projections, maintain records and perform other tasks which, until now, required sophisticated business computers. ELITE*CALC is a serious tool for those who want to do more than play games on their micro.*

Availible from stock on cartridge for the Dragon 32 or 64 or Tandy Color Computer... please specify. Shortly available on disc for the Dragon Data or Cumana/Premier or Radio Shack systems. Also available in the $–format instead of the £–format if requested.

THE BEST FOR ONLY £45

Elite*Calc has had excellent reviews in the American press and an enthusiastic reception at the recent International Rainbowfest.

*"Elite*Calc is a great spreadsheet program"*
Stuart Hawkinson, Rainbow

"Truly one of the best programs I have seen"
John Steiner, Micro

"Very powerful program... essential to every serious user"
Mike Jarvis, M & J Software

ENQUIRIES INVITED FROM RETAILERS AND DISTRIBUTORS (HOME AND OVERSEAS)

MICROCARE
1 OAKWOOD ROAD,
RODE HEATH,
STOKE-ON-TRENT
☎ (09363) 5695

DRAGON 32/64
TANDY COLOR COMPUTER

The family favourites for your Dragon 32

NEW RELEASES

MYSTERY OF THE JAVA STAR: Four part (over 100K) educational adventure for the older child. 3 levels of difficulty. Packed with challenges, puzzles and information. £7.95

SHAPER: Comprehensive sound effect utility. Create, store and modify machine code sound effects, for use in your own programs. Comes with a store of 100 effects and a comprehensive manual. £9.95

NORTH SEA OIL: Simulation game. Run your own oil rig and see how fast you can make 100 million dollars! £5.75

PETTIGREWS DIARY: Epic 90K adventure for the whole family. Packed with features. £7.95
EMPIRE: Highly praised strategy game. £6.95
PUZZLER: Jigsaw puzzle game, 4 levels. £6.95
HOOKED: Entertaining fishing game. 1/2 players. . . £5.75
FUN TO LEARN: Menu based education for 6-12 year olds. £6.95
MONSTER MATHS: Menu based education for 8-14 year olds. £6.95
FAMILY PROGRAMS: Games/education/utilities. . . £6.95
FUN & GAMES: Games compendium for all ages. . . £6.95
LIVE & LEARN: Educational compendium for older children. £6.95
QUIZ PACK: Two quiz games. 500 words. £3.95
INFANT PACK: Pre-school education. Counting and alphabet. £3.95
JUNIOR PACK: Primary school education. £3.95

DEALERS! Phone us now on 01-514 4871 for our comprehensive catalogue

SHARDS

Available from selected branches of Boots and all good stockists, or by telephoning Access/Barclaycard orders to 01-514 4871 or by sending cheque/PO to: 189 ETON ROAD, ILFORD, ESSEX IG1 2UQ.

THIS MONTH . . .

SOFTWARE REVIEWED

VENTURE FORTH

The unusual computer language FORTH is now available from DRAGON DATA on a cassette for use on DRAGON 32 or 64. A FORTH program will normally run faster and use less memory than an equivalent BASIC program — although slower than machine code, it is exceptionally efficient for a high-level language.

At first sight it is an odd looking language. It employs reverse Polish notation (RPN), which is the same as that used by some Hewlett Packard calculators. Thus the BASIC expression 5+3*7 becomes 3 7 * 5 + in FORTH.

The main reason for using this unusual notation is that FORTH uses stacks. This in itself is not unusual, since all languages use stacks internally, but FORTH interacts with the programmer via stacks. In fact, virtually nothing can be done in FORTH without using stacks. When a number is entered, it is PUSHED on to the stack. When an operator is entered, two numbers are PULLED from the stack, the operation is carried out, and the result is PUSHED on to the stack.

FORTH consists of a standard vocabulary of "words". A word is roughly equivalent to a subroutine in BASIC. DRAGON FORTH is particularly powerful, since it includes a number of extensions to the standard FORTH vocabulary. The real strength of DRAGON FORTH, however, lies in its direct access to most of the DRAGON's BASIC commands, including the high resolution graphics commands.

The basic principle of FORTH programming is the use of the existing vocabulary to define your own new words, which in turn may be used to define more words. New words are defined using the colon. Thus the FORTH statement

```
: POWER4 DUP * DUP * . ;
```

creates a "subroutine" which takes the number at the top of the stack, finds its fourth power, and prints out the result. The operator "DUP" duplicates the top number on the stack, so that when the * operator is applied, the result is the square of the original number. This process is repeated to create the square of the square, or the fourth power, then the "." operator prints out the result. The semi-colon indicates the end of the definition of the word "POWER4".

Once POWER4 has been defined in this way, any subsequent statement of the form **3 POWER4** will push the number (in this case 3) on to the stack, then calculate its fourth power (81) and print the result.

DRAGON FORTH allows you to access BASIC via commands of the form **B[..BASIC statements..]**

One-line BASIC programs can be included, but commands which need line numbers (such as GO TO) cannot. Thus the command

```
B[ PRINT"THIS IS A STATEMENT" ]
```

will print **"THIS IS A STATEMENT"**, and the command

```
B[ PMODE3:SCREEN1,Ø:PCLS:CIRCLE(8Ø,8Ø),
   7Ø,4:PAINT(8Ø,8Ø),2,4 ]
```

will enter graphics mode 3, display the screen, draw a red circle and paint it yellow. It will also leave you in graphics mode, making it difficult to see what is happening on the text screen! A simple statement like **B[PRINT]**
can be keyed in, and will return you to text mode, but it must be done "blind". It is useful to note that the "?" form of "PRINT" also works.

DRAGON FORTH also includes a comprehensive line editor, which allows you to write FORTH programs as "pages" of text. These pages can be saved on tape, by accessing BASIC and using CSAVEM, so that they can be reloaded later. They can also be compiled into the FORTH dictionary, using the FORTH command LOAD, to be executed later.

To the programmer who has never used anything but BASIC, FORTH is something of a challenge. It doesn't use any of the "safety checks" that DRAGON's BASIC Interpreter performs, and this can make debugging difficult. But if you are keen to speed up your graphics animations, and don't want to go all the way down to the level of machine code, the challenge is worth accepting.

The first step is to get to grips with the stack system and the 'reverse-Polish' logic. A program that simulates these two aspects at least as far as numerical operations are concerned and displays the stack as it is manipulated is an invaluable aid to those about to 'venture forth'.

The program below is designed to do just that. It uses a stack of size 1Ø and allows you to enter integers and perform standard FORTH integer arithmetic using the operators.

+	adds together the top two numbers on the stack.
–	subtracts the top number from the second number.
*	multiplies together the top two numbers.
/	divides the second number by the first, ignoring any remainder.
/MOD	performs as /, but leaves both result and remainder on the stack.
DUP	duplicates the top number on the stack.
DROP	deletes the top number
ROT	"rotates" the top 3 numbers, so 1 2 3 becomes 3 1 2.
SWAP	swaps the two top numbers.
.	prints the top number.

The first five operators pull the numbers from the stack, operate on them, and push the result(s) on to the stack. The output operator "." pulls the number, prints it, and does not replace it.

Type in the program and RUN. The stack is displayed in the middle of the screen and your FORTH instructions along the top of the screen. Key in numbers as usual, and use the keys D for DUP, S for SWAP, R for ROT, +,–,*, and / for their FORTH counterparts. In addition use the downward arrow for DROP and ? for /MOD.

Although a single key is used for each instruction (allowing easy branching through an extensive use of INSTR) the equivalent FORTH commands are displayed. Unlike the version of FORTH produced by DRAGON DATA the period "." produces an immediate display. When your list of FORTH words is long (up to two lines is OK) you can clear the top of screen by pressing ENTER which clears the list of previous commands but does not alter the stack.

In FORTH you have to be careful to leave a space between each instruction. Our BASIC program uses the space to implement the operation and an error message is generated if you do not follow this cardinal rule. If you make a mistake in keying in, the last key pressed may be retrieved by use of the right arrow.

As a first attempt try the following sequence of commands (remember those spaces!).
23 32 5 + * calculates 23*(32+5)
5 23 S ? (standing for 23 5 SWAP /MOD which should give you 4 and 3 on the top of stack as 5 goes into 23 4 times with remainder 3.

It is possible to extend the program to allow other FORTH primitives but hardly worth it. The program is designed to introduce you gently to the essential syntax of the language and no more. Perhaps such a foretaste might tempt you to your local stockist to purchase the real thing.

```
5 REM TRY FORTH – A. M. SYKES
  DEC 1983
1Ø CLEAR1ØØØ:DIMX(1Ø),X$(12):
   CLS:GOSUB21Ø
2Ø A$="Ø123456789"+CHR$(8):B$=""
3Ø B$=CHR$(32)+"+–*."+CHR$(47)
   +CHR$(63)+"DSR"+CHR$(1Ø)+CHR$(13)
4Ø FOR I=1 TO6:X$(I)=MID$(B$,I,1):NEXTI
5Ø FOR I=7 TO 11:READ X$(I):NEXTI
6Ø DATA/MOD,DUP,SWAP,ROT,DROP
7Ø CC$=""
8Ø K$=INKEY$:IF K$=""    THEN 8Ø
9Ø A=INSTR(1,A$,K$):
   B=INSTR(1,B$,K$):IF A=Ø AND
   B=Ø THEN 8Ø
1ØØ IF B=12 THEN C$="":CC$="":
   PRINT@1,STRING$(96,32):GOTO8Ø
11Ø IF LEN(CC$)>Ø AND A=11
   THEN C$=D$:CC$=DD$:GOSUB23Ø:
   GOTO8Ø
12Ø D$=C$:DD$=CC$:IFB>Ø
   THEN C$=C$+X$(B) ELSE C$=C$+K$
13Ø GOSUB23Ø:CC$=CC$+K$:L=LEN(CC$)
14Ø IF L<2 THEN 8Ø
15Ø BC=INSTR(1,B$,DD$)
16Ø GOSUB23Ø:IF B=1 AND BC>1
   THEN ON BC–1 GOSUB 26Ø,27Ø,28Ø,29Ø
   ,31Ø,32Ø,33Ø,34Ø,35Ø,36Ø:GOTO7Ø
17Ø IF B=1 AND BC=Ø THEN X(Ø)
   =VAL(DD$):GOSUB24Ø:GOSUB25Ø:
   GOTO7Ø
18Ø IFB=Ø AND BC<4 THEN 8Ø
19Ø GOSUB3ØØ:GOTO8Ø
2ØØ IF B=1 AND BC=1 THEN 8Ø
21Ø FOR I=1 TO 9:PRINT@1Ø6+I*32,
   I;":";:NEXTI:PRINT@1Ø5+I*32,I;":";
22Ø RETURN
23Ø PRINT@1,C$:RETURN
24Ø FOR I=9 TO Ø STEP –1:
   X(I+1)=X(I):NEXTI:GOSUB 25Ø:RETURN
25Ø FOR I=1 TO 1Ø:PRINT@11Ø+I*32,
   X(I):NEXTI:RETURN
26Ø X(1)=X(1)+X(2):GOSUB 37Ø:
   GOSUB 25Ø:RETURN
27Ø X(1)=X(2)–X(1):GOSUB 37Ø:
   GOSUB 25Ø:RETURN
28Ø X(1)=X(1)*X(2):GOSUB37Ø:
   GOSUB 25Ø:RETURN
29Ø PRINT@69,"display   ";X(1):
   GOSUB36Ø:RETURN
3ØØ PRINT@1,C$;"error";:
   FOR I=1 TO 9ØØ:NEXT:CC$=DD$:C$=D$:
   GOSUB23Ø:RETURN
31Ø X(1)=FIX(X(2)/X(1)):GOSUB25Ø:RETURN
32Ø W=X(1):X(1)=FIX(X(2)/X(1)):
   X(2)=X(2)–W*X(1):GOSUB25Ø:RETURN
33Ø FOR I=9 TO 1 STEP –1:
   X(I+1)=X(I):NEXTI:GOSUB25Ø:RETURN
34Ø W=X(2):X(2)=X(1):X(1)=W:
   GOSUB25Ø:RETURN
35Ø W=X(3):X(3)=X(2):X(2)=X(1):
   X(1)=W:GOSUB25Ø:RETURN
36Ø FOR I=1 TO 9:X(I)=X(I+1):
   NEXTI:GOSUB25Ø:RETURN
37Ø FOR I=2 TO 9:X(I)=X(I+1):
   NEXTI:GOSUB25Ø:RETURN
```

ALLDREAM

As mentioned in STOP PRESS 5 the cassette based assembler program DREAM is complemented with DREAMBUG, a program designed to provide the user with the tools to dis-assemble machine code and de-bug programs. Together they form ALLDREAM but if bought separately, DREAMBUG comes complete with a program to combine both together ready for you to record a copy of the whole package.

Together they form a comprehensive toolkit for the machine-code user. In particular the printer options in DREAMBUG make it extremely easy to obtain listings of assembler mnemonics from machine code programs in RAM or ROM.

ALLDREAM when loaded resides in RAM from hex 6Ø8Ø to the end (7FFF), which conflicts with the practice in this magazine of constructing machine code programs in RAM from say hex 7DØØ onwards. Further, DREAM (under default instructions) assembles programs from 2ØØØ1 onwards. This means that it would be necessary then to protect 20000 onwards from BASIC by use of CLEAR and this would be unnecessarily wasteful of space, particularly for programs such as the INDEX program discussed in the last issue.

We prefer therefore when using ALLDREAM to load it into RAM further down leaving hex 7D00 onwards free for machine code. This may be achieved by using an OFFSET. To calculate the (negative) offset required let's assume that we wish to install ALLDREAM from 23000 (instead of 24704) onwards. The necessary offset will be 65536+ 23000 - 24704 =63832 and the command for cassette loading is

```
CLOADM"ALLDREAM",63832
```

(From disk the equivalent command would be LOAD"ALLDREAM.BIN",23000) To assemble a program from hex 7D00 onwards simply insert the two lines ORG $7D00 and PUT $7D00 at the beginning of your text file (see for example this issue's machine code corner).

Of course you have to adjust all the addresses supplied with ALLDREAM accordingly so that for example with our suggested position, EXEC 23000 takes you to DREAM whilst EXEC 27996 takes you to DREAMBUG.

DRAGON DATA SOFTWARE REVIEWS

This month 'Shaft' and 'Viking' have been examined by local schoolchildren. Descriptions of the software titles are given below, follow the scores and comments of the children.

Title: SHAFT — Price £7.95

The aim of the game is to move across the screen from one side to the other, avoiding the eight elevators moving up and down. you begin the game at the base of the screen and each time you successfully cross it, a special elevator will lift you up a level. After crossing the screen safely at every level, you will eventually arrive at the top, when you will be presented with a new, faster moving screen. Your reactions will have to get quicker and quicker for you to reach the top of the shaft.

Marks out of 10	User A	User B
Graphics	4	4
Skill Required	8	9
Ease of Use	7	7
Documentation	6	8
Value for Money	7	8
Overall Mark Out of 50	32	36

Comments

An enjoyable game with a lot of lasting appeal. Good value for money.

Title: VIKING — Price: £7.95

In 'Viking' you are given the chance to work your way up the ranks to become King or Queen. But first you must make decisions regarding what is to be bought and sold, what taxes to impose for the coming year, what food your people will need...all your decisions can have far reaching effects. For example, taxation will affect profits, food will affect the population. A disaster, such as a terrible plague, raid or a revolution may befall your kingdom, but you have the power to divert them or prevent them altogether. If you prove your worthiness and ability, you will rise to be the Viking Monarch.

Marks out of 10	User A	User B	User C
Graphics	3	4	4
Skill Required	8	8	8
Ease of Use	9	9	8
Documentation	8	9	9
Value for Money	8	8	9
Overall Mark Out of 50	36	38	38

Comments

A very interesting and enjoyable game. A lot of skill required.

(continued from page 9)

"Basic Tutorial", and its advanced complement, "Advanced Tutorial"; and ready later this month are, "Computer Science" and " 'O' Level Physics". Under production and on the drawing board are, "Geography", "Biology", "Astronomy", "French", and "German".

Superspy is a brilliant combination of every popular video game skill in a unique and thrilling teach-yourself history format. This fun way of acquiring knowledge of British history 1939-46 has been cleverly devised by a top history teacher for 11 year olds and upwards.

The full list of software now available from Dragon Data at £14.95 is:

Basic Tutorial	(2 cassettes)
Advanced Tutorial	(2 cassettes)
Maths 1 (4-6 year olds)	(2 cassettes)
Maths 2 (6-7 year olds)	(2 cassettes)
Maths 'O' Level	(3 sets of 2 cassettes)
Superspy	(2 cassettes)
Early Reading	(2 cassettes)
Computer Science	(2 cassettes)
'O' Level Physics	(2 cassettes)

Please see order form on separate insert

PREVIEW OF 'INSIDE THE DRAGON'

One of the major criticisms levelled at the Dragon 32 and 64 was the lack of technical information available for them. 'Inside the Dragon', written by Duncan Smeed and Ian Sommerville and published by Addison-Wesley, has been written specifically to fill that gap.

'Inside the Dragon' is 360 pages long and consists of nine chapters of text, a suggested further reading list, eight appendices and a complete index. The first chapter introduces basic computer principles using, obviously, the Dragon as a specific example. The second chapter explains the architecture of the M6809 microprocessor covering such topics as the register set, addressing modes and memory-mapped input/output. Chapter Three contains a detailed description of the M6809's instruction set. The next chapter introduces assembly language explaining the facilities that a typical assembler, in this case Dragon Data's own Editor/Assembler package, provides and how it is used to create machine code programs. The fifth chapter then goes on to explain how the various BASIC statements can be coded in assembly language and illustrates how good programming techniques can be used to simplify the writing of assembly language programs. This chapter concludes with a complete assembly language source program of a simple monitor. The sixth, and final chapter devoted to assembly language programming, describes the use of subroutines with examples of character string manipulation. This chapter is also used to explain advanced programming techniques in assembly language including parameter passing using the stack, recursive subroutines, position-independent code and combining assembly language with BASIC.

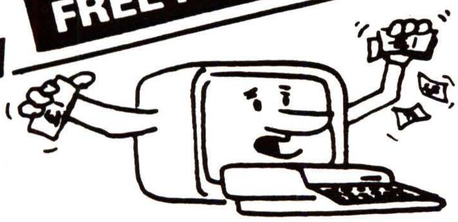
The seventh chapter is the first to explore the Dragon's potential in depth and describes the machines graphic capabilities and how they can be manipulated by the assembly language programmer. Among the topics discussed are graphics display hardware, integrating BASIC and assembly code graphics, display modes, graphics utilities and designing and implementing graphics programs. This chapter concludes with a listing of a complete animated graphics program. Chapter Eight is devoted to a description of I/O programming techniques and includes full details of the Dragon I/O hardware and how it may be used. The ninth chapter concludes the book with hints and tips which include details of the power-up and reset actions, how BASIC programs are stored, how BASIC represents strings and numbers, how parameters can be passed from BASIC to machine code programs and vice versa, how to extend the Dragon BASIC with new reserved words and facilities and finishes with a complete list of BASIC's system variables.

The final 100 pages of the book consist of the appendices which contain the data sheets of the 4 major chips in the Dragon, details specific to the Dragon 64 and Disk Operating System and miscellaneous information such as BASIC token values and I/O jump tables.

ACTIVE SOFTWARE

ANNOUNCE

FREE PARKING



ONLY ACTIVE SOFTWARE bring you a *unique* version of the all time classic board game, which upholds every tradition of the original,

BUT WITH ONE BIG DIFFERENCE . . . . .

YOUR DRAGON PLAYS AS WELL!!!

And it plays a MEAN, MEAN game — using cunning pre-programmed logic, it will match the toughest of opponents.

And that's not all! . . . You can wave goodby to your Dragon's green on black text! **FREE PARKING** uses 100% *SUPER SMOOTH, SUPER FAST,* Machine Code Graphics throughout the whole game.

And there's more!! . . . **FREE PARKING** is played with the whole board *permanently* displayed on the hi-res screen, so you can always see your opponents.

AND EVEN MORE!!!

- **PLAYER STATUS PERMANENTLY DISPLAYED**
- **DETAILED REPORTS AT THE PUSH OF A BUTTON**
- **SAVE GAME TO TAPE**
- **OUTPUT TO PRINTER**

WE DARE YOU TO CHALLENGE YOUR DRAGON! A MASTERPIECE! at £6.90 inc.

DUPLICATION SERVICE

We specialise in the reproduction onto cassette from your Master, to the highest possible standard. Using real time and high speed duplicating equipment specifically designed for the ultimate in fidelity and efficiency we consider ourselves the very best for any computer programme, or spoken work, if your standards are high and perfection is essential

Example: C5 both sides recorded **75p each** + VAT

BLANK CASSETTES

Example: C5's **30p** + VAT + **Postage**

Library cases **6p each** + VAT

Generous discounts available on duplication and blank cassettes.

Price List available on request.

CASSETTE LENGHTS IN MINUTES			BLANK CASSETTES A 1 – 9	B 10–24	C 25–49	D 50–249	E 250 +
C1	–	C5	0.30	0.28	0.22	0.20	0.19
C6	–	C10	0.34	0.32	0.24	0.22	0.21
C11	–	C20	0.40	0.38	0.29	0.26	0.25
C21	–	C30	0.45	0.43	0.33	0.30	0.28
C31	–	C40	0.52	0.49	0.38	0.34	0.32
C41	–	C50	0.58	0.55	0.42	0.38	0.36
C51	–	C62	0.64	0.61	0.46	0.42	0.40
C63	–	C72	0.70	0.66	0.51	0.46	0.44
C73	–	C82	0.76	0.72	0.55	0.50	0.47
C83	–	C92	0.82	0.78	0.60	0.54	0.51
C93	–	C100	0.87	0.83	0.63	0.57	0.53

EXTRAS

LIBRARY CASES — 6p each

GAMES TAPE 1

WITHOUT QUESTION, THE BEST SELLING COMPENDIUM AVAILABLE. 1000's OF SATISFIED CUSTOMERS.

***BEST BUY**

FOR ALL DRAGON 32 OWNERS, AN EXCEPTIONAL OFFER TO BUY THE HIGHLY ACCLAIMED GAMES TAPE 1
8 SUPERB FULL LENGTH PROGRAMS ON ONE CASSETTE, AN INCREDIBLE 200K OF PROGRAMMING SPECTACULAR COLOUR GRAPHICS AND SOUND EFFECTS

"The best buy must be the recent Games Tape 1 from a new source - Active Software which has taken some established ideas and implemented them beautifully using all of the Dragon's facilities cleverly. There are eight games, several of them as good as those offered elsewhere as individual cassettes." (YOUR COMPUTER – APRIL 1983)

Interplanetary Trader
A massive 30K of real time adventure

Wumpus Mansion
Incredible fun! Outwit the Wumpi
Uses 29K.

Hi-lo
Brilliant sound hi-res graphics.
Uses over 20K

Atom Hunt
95 per cent full-colour graphics!
Uses over 20K

Execution
Hours of fun guaranteed! A massive 30K

Wipeout
4 colour graphics.
Be warned — this is not easy.

Snail Pace
the excitement is almost unbearable.
A massive 26K

Air Assault
Uses sound and realistic movement of helicopter.

AMAZING VALUE
ONLY
£6.90

AVAILABLE SOON

For all of our customers who thought that **GAMES 1 TAPE** couldn't be equalled . .

WE HAVE DONE IT

GAMES TAPE 2 *SIMPLY AMAZING!!!!!*

WITHOUT QUESTION, THE BEST SELLING COMPENDIUM AVAILABLE. 1000's OF SATISFIED CUSTOMERS.

PROFESSIONALLY BUILT RS-232 INTERFACES

£49.95

Use your Dragon 32 to communicate with RS232 devices, I.E. Modems, Printers, Other Computers etc. Fully documented with test programs.

SUPER QUALITY COVERS

An investment at only

£3.50

NOT CHEAP stretchable loose fitting plastic, but a high quality tough, durable material in matching Dragon Colours. Looks very smart. GO ON, TREAT YOUR DRAGON

POST COUPON **NOW** TO:—

ACTIVE SOFTWARE
117 ICKNIELD STREET
BIRMINGHAM B18 6RZ.

PLEASE RUSH ME A COPY OF GAMES TAPE 1. ☐ RS232 ☐ COVER ☐
TFS ☐ CASS ☐ F.P. ☐

I enclose a cheque/P.O. for £ ________

NAME ____________________

ADDRESS ____________________

____

ADVERTISING RATES

If you would like to advertise in 'Dragon World', then please contact Stephen Davis at Dragon Data Limited on 0656-744700. Our rates are highly competitive:

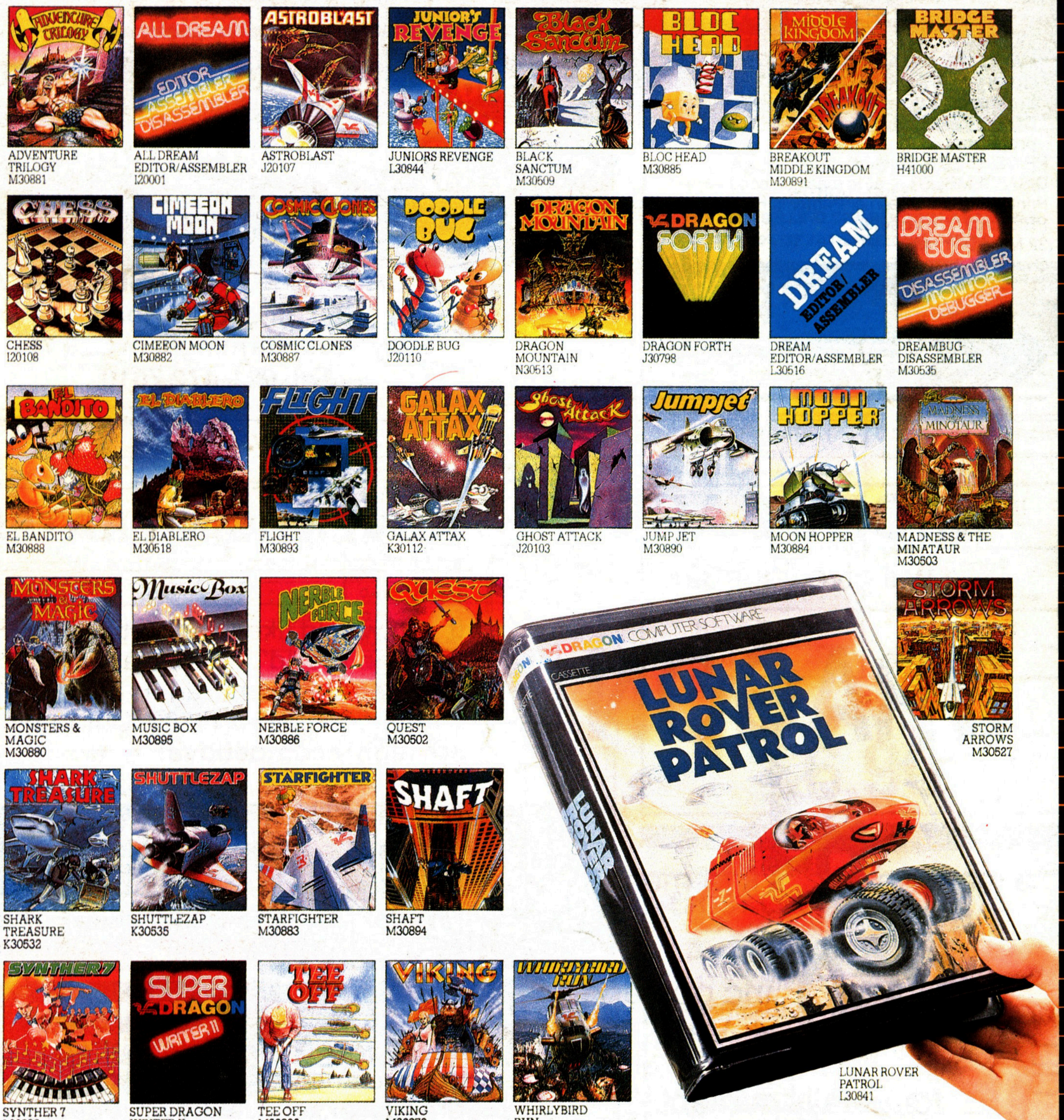
Full page colour	**£490**
Full page black and white	**£320**
Half page black and white	**£170**
Quarter page black and white	**£85**

(Spot colour 10% extra)

The publishers would like to notify readers that the appearance of an advertisement in 'Dragon World' does not necessarily mean that the product or service shown is approved by Dragon Data Limited.

FEED YOUR DRAGON

EXCLUSIVE SOFTWARE FOR DRAGON MICROCOMPUTERS



ADVENTURE TRILOGY M30881

ALL DREAM EDITOR/ASSEMBLER I20001

ASTROBLAST J20107

JUNIORS REVENGE L30844

BLACK SANCTUM M30509

BLOC HEAD M30885

BREAKOUT MIDDLE KINGDOM M30891

BRIDGE MASTER H41000

CHESS I20108

CIMEEON MOON M30882

COSMIC CLONES M30887

DOODLE BUG J20110

DRAGON MOUNTAIN N30513

DRAGON FORTH J30798

DREAM EDITOR/ASSEMBLER L30516

DREAMBUG DISASSEMBLER M30535

EL BANDITO M30888

EL DIABLERO M30518

FLIGHT M30893

GALAX ATTAX K30112

GHOST ATTACK J20103

JUMP JET M30890

MOON HOPPER M30884

MADNESS & THE MINATAUR M30503

MONSTERS & MAGIC M30880

MUSIC BOX M30895

NERBLE FORCE M30886

QUEST M30502

STORM ARROWS M30527

SHARK TREASURE K30532

SHUTTLEZAP K30535

STARFIGHTER M30883

SHAFT M30894

SYNTHER 7 L30838

SUPER DRAGON WRITER II F20310

TEE OFF M30896

VIKING M30879

WHIRLYBIRD RUN K30114

LUNAR ROVER PATROL L30841

DRAGON

Dragon Data Ltd., Kenfig Industrial Estate, Margam, Port Talbot, West Glamorgan, SA13 2PE.