

# Windfall

Volume 2, No. 6 December 1982 £1



The Apple computer users' magazine

## The games Apples play

21 pages of ideas to cheer your computing Christmas

## Take Visicalc to the party

## Using Pascal to print graphics

## Wordpower with Screenwriter II

## Z80 cards reviewed

## Teaching 'O' level computer studies

## Taking the lid off assembler language



Make your Apple  
**REALLY** talk and sing!

Our special Christmas offer  
— a unique hi-fi adaptor  
— See page 49

# Windfall

The Apple magazine

The names Apple play

21 pages of ideas to do with your computer. December

Take Visicalc to the party

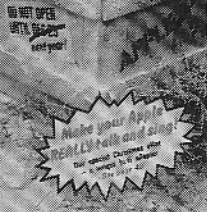
Using Pascal to print graphics

Workpower with Gorawriter II

280 words reviewed

Teaching 'D' level computer studies

Taking the lid off assembler language



Vol. 2 No. 6 December 1982

*Managing Editor*  
**Derek Meakin**

*Features Editor*  
**David Creasey**

*Art Editor*  
**Peter Glover**

*Technical Editors*  
**Peter Brameld**  
**Cliff McKnight**  
**Max Parrott**

*Advertisement Manager*  
**John Riding**

*Advertising Sales*  
**John Snowden**  
**Mike Hayes**

*Marketing Manager*  
**Bruce Everiss**

Tel: 061-456 8383 (Editorial)  
061-456 8500 (Advertising)  
Telex: 667664 SHARETG

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

**Subscription rates for 12 issues, post free:**

£12 - UK  
£13 - Eire  
£18 - Europe  
£15 - USA (surface)  
£25 - USA (airmail)  
£15 - Rest of world (surface)  
£30 - Rest of world (airmail)

Trade distribution in UK and Ireland by Cemas Ltd, New Road, St. Ives, Huntingdon, Cambridgeshire PE17 4DB. Tel: 0480 65886.

Writing for Windfall: Articles and programs relating to the Apple are welcome. Articles should preferably be typed or computer-printed, using double spacing. Unsolicited manuscripts, discs, etc, should be accompanied by a self addressed stamped envelope, otherwise their return cannot be guaranteed.

© 1982 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles or listings.

Apple and the Apple symbol are the registered trade marks of Apple Computer Inc. Windfall is an independent publication and Apple Computer is not responsible for any of the articles in this magazine, nor for any of the opinions expressed.

# LISTING

**17** **WHAT'S NEW**  
A quick look at the Apple world

**23** **THINK TANK**  
Our forum for programmers

**26** **GRAPHICS**  
Doing the impossible in Pascal

**29** **MACHINE CODE**  
Interactive editor-assembler

**32** **VISICALC**  
VC magic for Christmas parties

**36** **APPLETIPS**  
They make programming easier

Get into the Christmas spirit now

## GAMES!

Page after page of ideas start on 39

**60** **FEEDBACK**  
Programs that do . . . and don't

**63** **APPLECART**  
Aid for tyro computer teachers

**69** **SCREENWRITER II**  
Word processor package reviewed

**72** **PEEK and POKE**  
Max Parrott concludes his beginners' guide

**75** **COMPUCOPIA**  
The latest in software/hardware

**81** **CP/M**  
Z80 cards for the Apple II

# WHAT'S NEWS...

By David Creasey

## Sales figures confound the critics

THE current performance of Apple worldwide is confounding those critics who say that the company is on the way out and that it has left it too late in producing new products for it to be able to remain competitive.

The company's Cupertino headquarters has reported a 74 per cent increase in net sales and a 56 per cent increase in net income for the financial year ended last September, compared with the previous year.

World net sales increased to £343 million from £196.95 million a year ago. Net income increased to £36.06 million, or £0.63 per share, from £23.18 million, or £0.42 per share, in the comparable period last year.

And fourth quarter sales and net income were up 80 per cent and 71 per cent, respectively, compared with the previous year. Sales increased to £103.42 million from £57.48 million a year ago and net income rose to £11 million compared with £6.48 million in the fourth quarter last year.

Apple president A.C. Markkula said the sales increases resulted from special promotions still underway.

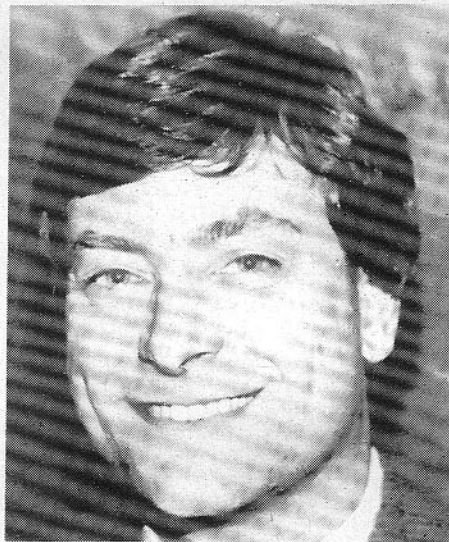
Apple has increased its investment in research and development to £22.36 million for the year, 81 per cent more than last year.

"Several major products resulting from this effort are scheduled for introduction in 1983, including enhancements to the present line," said Mr Markkula, "I'm confident that these products will further enhance Apple's already strong position as the unquestioned leader in the personal computer industry."

## Copycat is clawed

APPLE has brought its fight against copycat machines to Europe, and appears to have scored a first round victory. The Apple II lookalike, the Apolo II, manufactured by Sunrise Computers in Taiwan, went on sale in Europe in October and Dutch businessman Jaap Cornelisse, who was marketing it under the auspices of the CAB Trading Company, said his first consignment of 25 machines was quickly snapped up.

However Apple and its Benelux distributor took him to court and in



Keith Hall

October the president of the Amsterdam district court banned the Dutch company from offering, selling and delivering the Apolo II.

He also forbade the company from distributing a brochure using expressions like "CompAppleBle" or "Apple II compatible," and barred even the use of the word Apolo in every respect as, he said, it constituted a direct infringement of Apple, the registered trade mark of Apple Computer Inc.

The court ruled that the overall behaviour of CAB Trading was intolerable and represented unfair competition, as the Apolo II was a direct copy of the Apple II. The external and technical design, as well as the software contained in Eprom memory, were found to be copied.

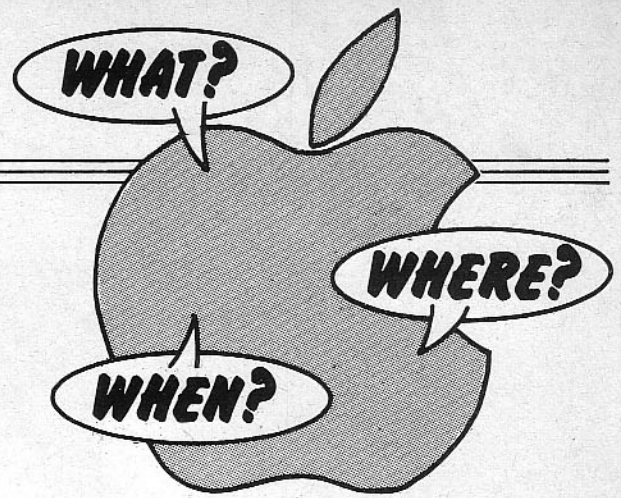
## Bargain offer is still open

APPLE UK's special cut price offer continues until Christmas Eve. It reduces the cost of an Apple II system with 48k memory, twin disc drives and software (including business graphics, Applewriter and a Fortran language card) to £1,506 — a saving of £300 on the old price.

Rather than reflecting a special festive mood, the offer is a response to market pressures (competitors such as Commo-

dore and Tandy also have cut price offers) and could also be the prelude to the long-awaited launch of a new machine such as Lisa, the Super II, Mackintosh, or the expected mini-Apple.

However that shouldn't discourage prospective purchasers. The Apple II has a huge existing user base and it is extremely unlikely that any new additions to the family will succeed in devaluing it.



## Keith joins the Apple board

SALES of Apples in Britain have been booming as never before since dynamic Keith Hall left Commodore six months ago to become sales chief for arch rivals Apple UK.

And while Keith would modestly disclaim personal responsibility for Apple's remarkable successes, he is getting his reward all the same.

Apple chief Peter Cobb has made him a full member of the board, and given him the title of Sales and Marketing Director.

Said Cobb: "The impact of Keith's presence upon the company has seen the development of a strengthened professional field sales team which has brought about an increase in all types of support for the dealer network, now totalling around 400 nationwide."

He added: "By combining the sales and marketing function we will strengthen the direction of our planned objectives for 1983, which will incorporate the introduction of several new products".

But as to what those mysterious new products will be Peter Cobb's lips are as firmly sealed as ever. Despite all *Windfall's* efforts to be more revealing on the cover of this month's issue, no hint of what the company has in store has been allowed to leak from their Hemel Hempstead fortress.

## Flying the flag again

ONE of the more remarkable aspects of Apple UK's official representation at micro shows in the past year has been its lack of it. More often than not individual dealers, software houses (and *Windfall*) have had to carry the Apple flag and handle consumer queries at exhibitions such as Apple '82 and the PCW Show.

However the situation seems to have changed. Apple UK turned up at the European Computer Trade Forum in Birmingham in September and last month was at Compec at London's Olympia, where the theme of its stand was networking and communications.

The company is currently installing a Zynar Cluster One networking system at its Hemel Hempstead headquarters, and the display featured both Zynar and Apple systems.

"The computer world is fast-changing," says Keith Hall, Apple's sales director, "and we had the stand designed with this in mind. We wanted to present Apple in the forefront of change, and with the addition of the Zynar range we feel we had a display which really represented our position as an innovator ready to meet the latest challenges in computer technology."

Another major feature of the display was a stand centred on the Apple III, which Apple sees as taking off from a sluggish first year to become the business machine of 1983.

## Nagging doubts

THE proud boast (or profound fear) of the anti-computer lobby is that you can't programme humans. In the same way, you can't programme horses, and humans so far have been unable to write a program on the horses that will generate fabulous and consistent winnings.

John Clement — we described him as an Appletipster in our July issue — spent 10 years "perfecting" an Apple-based betting system. Last year he formed a



Back in the family firm: David Collis, Michael and Stephen Brewer

## It's the old firm again

IT'S quite like old times down at micro dealer suppliers Data Efficiency. After a break of two years Michael Brewer, his brother Stephen and David Collis are back in the family firm.

And what a momentous two years it has been for all three. It was originally as an offshoot of Data Efficiency that they set up Microsense Computers, which won the sole UK distributorship for Apple products.

Their style of management quickly established Apple as Britain's top-selling business micro, overtaking main competitor Commodore and at the same time being instrumental in setting up around 200 new businesses supplying Apple-related hardware and software.

The parent Apple company bought out Microsense in October last year, but the three bosses were kept on — Michael Brewer as managing director, Stephen as director of sales and marketing, and David Collis as director of operations.

Now their stay with Apple is at an end, and their expertise in these three positions will be benefiting Data Efficiency, which is expecting a current year turnover of £7 million.

Says Michael Brewer: "The number of micros — and greatly increased computer awareness — will mean a healthy growth in most companies in the business or associated with it. Fortunately our company has an interest in most aspects of computing."

company, Frankoform, and began marketing his system, Courseplan 200.

This year he persuaded six people to invest £1,000 each with his company (the original target was 200 investors), the money to be invested, following the Courseplan 200 system, over a 10 week period on various horse races throughout the country. His system was based on jockey and horse form and eliminated from the reckoning certain races and racecourses that were statistically unsuitable.

Last year, claimed Mr Clements, the system gave a 403.58 per cent return on £1,000 over 10 weeks. This year, in the real world with real money at stake, it failed. "It was a total and utter disaster and we were completely wiped out," said Mr Clements.

"It started well, but then in two consecutive weeks we had 11 losers in a row — which finished us off. Towards the end of the period the system came back and we had a good result, but by then it was too late."

Mr Clement has returned half of the original £1,000 stake each to five of the six investors (although there was no

guarantee given at the start and they knew that they were merely indulging in a computerised flutter.

"The funny thing is," he said, "with only one exception the general reaction was one of being sorry for me rather than concern for having lost money."

"We now maintain daily contact with each other and have formed good friendships. Actually some of us are still working at it, trying, with hindsight, some re-programming... it looks like we have got the making of a consortium here."

## Catching 'em young

APPLE UK has caught the scent of competition in the air and has launched its own campaign. It is asking youngsters, who must be 19 or under at the end of this year, to write a software program of up to 500 words long which must be suitable for running on a standard 48k Apple



Liz Chapman (left) and Frances Pra-Lopez ... Ladybird book led to a booming business!

II that can be fitted with a language card if required.

Entries, either disc or cassette based, must be in by the end of next month. The winner not only gets a complete Apple II system, including colour monitor, but will have his program distributed by Sigma Technical Press.

Sigma will also distribute some of the other winning entries.

There are 300 runners-up prizes, including 50 packages of books and software. Entries should have "a strong relevance to education."

Bob Senior, of Apple, says young minds can only make a significant contribution to the development of the software programming industry if they have a practical sounding board for their talents. "We think this competition is such a sounding board and will stimulate general interest in the technology and applications of computers," he said.

## Playroom tycoons

BUSINESS is booming for two Hampstead women who set up shop in a child's playroom with an Apple, a printer and a photocopier. Frances Pra-Lopez and Liz Chapman (their company is called Frizco) offer an agency service handling administrative chores such as mailing and addressing for small manufacturers, businesses and local professional people.

But neither of them knew a thing about computers when they started out just over a year ago. They met at a local institute evening class, discovered they were neighbours and within a fortnight were discussing the possibility of setting up a business together.

"We had thought of going into freelance bookkeeping," said Liz, "but after we had completed our bookkeeping studies we decided that was the last thing we wanted to do."

A TV program on computers gave the first tenuous spur to the new venture. Fascinated with the prospect, they visited computer exhibitions and found out what computers could do, what were available and what they cost. They read a Ladybird book entitled "The computer - how it works" and moved from there to learn

about data-processing using ready made software.

Their first client was a group of astrological counsellors for whom they virtually ran the business, sending out information, arranging seminars, doing the books. That got them off the ground.

Robinson's Baby Foods is now their biggest customer. The company wanted to revive its club for health visitors in the UK and asked Frizco to take on the club's mailing of information. Frizco started with 3,000 members. They've now doubled that and expect the club membership to reach 10,000 a quarter fairly soon.

"We are managing to do that, plus our other jobs, using an Apple, the PFS software package and a large number of floppy discs," said Liz. "We don't know where it's going, but it is rather exciting."

## Apple that went to seed

AN Apple in Worcester has gone to seed and in doing so has solved a long-standing problem which threatened the profitability of Mill Farm, Harvington, near Evesham, one of the largest plant raisers in the country.

The difficulty lay in persuading customers to return the plastic seedling trays in which plants are delivered. As these cost £2 each - more than the value of the contents in most instances - the costs were unacceptably high. Keeping track of the non-returned trays and invoicing for them became a major priority.

Philip Boers, the grower, says: "The computer is going to pay for itself on this saving alone. We have 100,000 boxes in circulation, which is a sizable investment to protect."

Adroit Systems installed the system which uses a 5 mbyte Corvus hard disc,

and wrote a software package which is now the mainstay of administration.

One of its most vital tasks is scheduling the growing of the 45 million plants Mill Farm produces every year. As orders are received they are automatically compiled by the required sowing date to meet the delivery. Later orders can be inserted into the schedule if required, and a forward seed requirement report is also produced.

As seedlings become ready the Apple rationalises deliveries and box collections into areas to minimise transport costs. Invoices are produced automatically and a full sales ledger is maintained with a number of reports, VAT, aged accounts, statements and address labels.

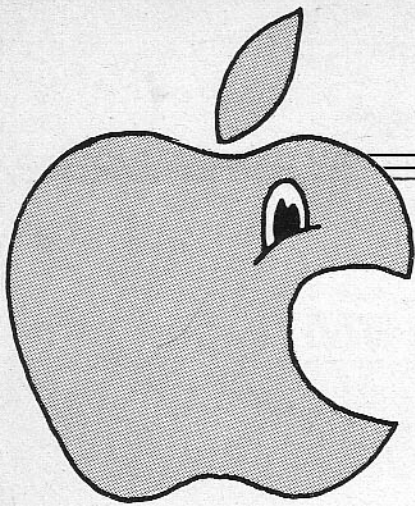
Phillip Boers added: "We have saved ourselves an immense amount of paperwork, improved efficiency all round and plugged a serious drain on our resources. The only major problem now is getting time on the system to run accounts and scheduling in parallel, so we've added Omninet and are linking a second Apple into the system."

## MBasic packs for Apple II

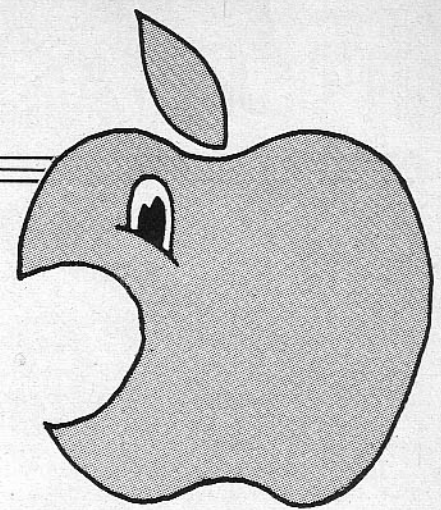
PEACHTREE Software's MBasic accounting packages, including sales ledger, inventory, invoicing, payroll and purchase ledger, are now available on the Apple II.

Originally the company had not planned to offer any software on the II in the UK, although it does so in the United States, but it has changed its mind after re-examining the market. Peachtree MD John Hale commented: "A lot of people are using the Apple II and we felt we should carve ourselves a niche in the marketplace."

The company is aiming its products mainly at large system users.



# THINK TANK



... the Windfall platform for those wishing to comment on or generally discuss specific articles in Windfall. Write to: Think Tank, Windfall, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

## Completing that Visicalc algebraic exercise

NICK Levy's Visicalc article, "First step for CBEs: Brush up your algebra", in the November *Windfall* urges Visicalc users to employ a working knowledge of algebra to make the best use of Visicalc, says **E.C.S. Eve**. Doubtless this is sound advice, but one is left wondering why it was not fully applied to the example shown in Figure 1 of this article.

In that example it was desired to calculate commission as a fraction of a net profit figure from which the commission had already to be subtracted. This is described as "a closed loop module, where two figures are mutually dependant on each other." In fact, however, with the application of a little algebra the module can be made open looped.

Let the required commission be  $x$ . Then let the net profit figure before deduction of commission be  $P$ , and the commission rate be  $R$ . Then, since the commission is to be a fraction  $R$  of the profit less commission we may write:

$$x = R (P - x)$$

which on re-arrangement becomes

$$x = \frac{RP}{(1 + R)}$$

This may be incorporated into the Visicalc formula by replacing the formula given for E5 (not E6 as in the text of the article) with the formula

$$((E1/100)*E4)/(1+(E1/100))$$

Provided also that the formula in E7 is corrected to E4-E5 the model will calculate the correct values first time round. The forward reference and hence the need for iteration will have been eliminated.

I imagine Nick was primarily interested in demonstrating a simple use of a recursive, "open loop", technique in Visicalc, and this would clearly be valuable in problems not so easily susceptible to algebraic manipulation; or where the result of such manipulation turned out to be a quadratic equation, or something even worse, which his Celebrated Business Executive might well be reluctant to tackle directly.

However, since the importance of a

working knowledge of algebra seems to be one of the main thrusts of Nick's article, it seems worth while pointing out that, at least in fairly simple cases, algebra may be able to convert a closed-loop problem into an open-looped one.

**Nick Levy replies:** *The object of the article was, as you say, to demonstrate the difference between the open and the closed loop modules. If, for example, you had to calculate the amount of VAT included in a gross sum, you wouldn't use that type of loop but would use mathematical formula as suggested. I acknowledge the typing gremlin that gave the E5 formula in E6!*

## Enigmatic LDPS

I'VE just been shown the May edition of *Windfall*. I might be a bit late, but in it Chris Clarke in his *Elements of the Apple* requests information on the mnemonic labelled LDPS.

This in fact means Load Parallel to Serial from the character generator (A5) into the shift register (A3). This in turn sends a bit-stream of data from pin 13 ending up at the video-selector/MPX (A9).

I hope this is of some assistance. — **T. Cullen**.

## Patching into DOS

SHAUN Hope, writing in the April *Windfall*, described how the &XREF facility provided with the DOS Toolkit would cause the system to hang if the program contained a DATA statement, or

block of DATA statements, not followed by a colon. This is because there is an incorrect assumption made after return from a subroutine.

To overcome the problem without having to add excess colons to your programs, add the following line to the LOADAPA program:

```
325 POKE 46989,32: POKE
46990,124: POKE 46991,145:
POKE 46992,201: POKE 46993,0:
POKE 46994,96: POKE
37229,141: POKE 37230,183
```

This patches the missing instruction back into the APA program, making use of an unused 6 byte area in the DOS (courtesy of Beneath Apple DOS).

Note that you should only make this change if you have DOS 3.3 and a 48k Apple. The area of DOS used for the patch does not conflict with the areas used by M.F. Sheppard's free disc space routine published in May, but you should check that you do not have any other customising routine in that area — **Neil Lomas**.

## Conceivable error

JOHN Pennel (*Feedback*, October issue) suggests that the transfer of a binary image between disc and text page 1 for instruction pages may lead to program failure.

Although I and other users of my Instruction File Editor (August issue) have not experienced this problem, he is quite correct to say that under certain special circumstances the error is conceivable, writes **Robert Beynon**.

Because the spare memory locations can be used by peripherals, BLOADing of

# THINK TANK

a binary image could alter the contents of these locations and cause program errors. I would be interested to know of other error-prone conditions.

The loss of current screen information can be circumvented readily by appropriate program design but I, too, have found it useful to move LOMEM above text page 2 and BLOAD the image into this RAM area, using a soft switch to display the page.

This generates problems with the display of a prompt line, but I have included subroutines to overcome these difficulties with the IFE (ver 1.8). Additionally such difficulties are discussed in the 20-page manual I supply with the software.

I do not consider myself to be a commercial software supplier but rather express a willingness to share my programs on a costs only basis (they are far too long to publish in *Windfall*).

In practice, I think that users of "non-linear curve fitting", or IFE 1.8, have found them useful and difficult to crash.

## More on ampersands

YOUR correspondent Kevin Cowtan is to be congratulated on his ingenious little HELLO program which substitutes an ampersand for the command POKE 33.33.

It does, however, depend on your having used that particular HELLO program to boot up before you started. Inevitably, there will be times when, for whatever reason, this will not be true.

May I therefore suggest that instead of a HELLO program, this be created as an EXEC file which can then be implemented at any time.

The listing below is for the program CREATE TXET which, when run, will create the EXEC file TXET. When EXECed at any time this will make the ampersand hook available. — **Stan Golding.**

```
1 REM PADDLE SCROLL - 2
2 FOR I = 730 TO 767 : READ N
3 POKE I,N
4 NEXT
5 DATA 169, 229, 133, 54, 169, 2, 133, 55, 32, 234, 3
6 DATA 72, 152, 72, 138, 72, 162, 0, 32, 30, 251, 200, 152,
170, 202, 208, 253, 136, 208, 249, 104, 170, 104,168,
104, 76, 240, 253
7 DEL 1,7
```

## Point taken, but . . .

THANKS for considering my Paddle Scroll routine worthy as an Appletip which appeared in the September issue, writes **Barry Hallam**. I note that in the November issue it was the subject of a revision by Mr Simon Edge. Well, I don't know whether to laugh, cry or simply blush! I swear that the relevant page in the DOS manual was NOT there when I was searching for an answer to my DOS register swap problem.

I did realise that my particular answer took for granted that DOS would be residing in a 48k system, but let's face it, 48k is the smallest Apple on the cart these days. However, point taken! Alas I must take issue with Simon. His revision (which doesn't do exactly as it should) infringes upon a vital proviso which I had taken special care to maintain. I purposely kept the routine one location above 768 (\$300) as this is a very popular start location for short routines, often 'sound'.

I had in fact hoped that someone would

take me to task, as indeed Mr Edge has done. He has pointed to the answer I really wanted, which was a simple re-connect CALL. Using the DOS routine (CALL 1002) from an assembly (ie. JSR \$3EA) provides the answer. I enclose my new listing (*above*) but wish to point out that only lines 2 and 5 are different. Once RUN, the utility is connected or re-connected by a CALL 730, the latter being necessary after RESET.

Certain Apples will respond with overflow error to HELLO progs like these:

```
10 REM HELLO PROG.
20 CALL - 1184
```

```
10 REM HELLO PROG.
20 CALL - 936
```

Apparently they cannot cope with a negative argument if it is the first evaluation requested. My fix is:

```
20 PRINT 2+2 : CALL - 1184
```

```
JPOKE 1657,80
```

```
JLIST
```

```
10 REM CREATE TXET
20 D$ = CHR$(4): REM CTRL-D
30 PRINT D$;"OPEN TXET"
40 PRINT D$;"WRITE TXET"
50 PRINT "POKE1013,76:POKE1014,128:POKE1015,3:REM SET & HOOK"
60 PRINT "POKE896,169:POKE897,33:POKE898,133:POKE899,33:REM POKE33,33"
70 PRINT "POKE900,76": REM RETURN TO BASIC"
80 PRINT D$;"CLOSE"
```

WHEN I bought my Epson MX80 F/T II, I was extremely pleased with the quality of print, the nice font (with descenders!), and the superb graphics facilities that it offered in Basic. The only disappointing thing was the notice near the back of the user manual which stated: "Some functions, including the copies of high resolution screen, cannot be performed in Apple Pascal".

Unfortunately the writer of that manual seemed to be right. I couldn't even get the 'bit-image' applications working completely, and they were far too slow to do a full HGR screen with anyway. Nevertheless, I refused to believe that HGR dumps were impossible, so I tried solving the problem through machine code.

My first approach was very simple. I wrote a piece of code (called by Pascal as an EXTERNAL procedure) that stored a \$91 (i.e. CTRL-Q) at the data input address of the printer, then did a JSR (jump saving return) to \$C100, which is the start address of the printer ROM. It worked, almost. There was only one nasty detail - as soon as the printer finished its dump, Pascal crashed.

The next step was to list the printer's driver program and disassemble it. It was quite easy to discover which bit of the code was used for the HGR dump, all I had to do was look for the line:

CMP £\$91 (compare with CTRL-Q) and examine what followed. The cause of the crashing soon became apparent. The routine was using the addresses \$279 to \$294 for bit-shuffling... perfectly safe in Basic because that area is comfortably inside the input buffer, but fatal in Pascal, since it wrecks some disc-handling information.

The routine was also using addresses

## Doing the impossible in Pascal

By J.P. LEWIS

\$0,\$1,\$24 and \$25 (the last two being Basic's horizontal and vertical cursor positions), but the contents of these were being swapped on and off the stack very neatly. And they aren't used (permanently) by Pascal anyway.

The key point in my second attempt was to save the contents of \$279-\$294 locally, jump to the printer routine, then put everything back in the right place afterwards. In a fit of caution, I also saved everything else (return address to Pascal, X,Y, and A registers), and set up a couple of values to ensure that the printer knew it was in slot 1.

This time everything went well. In fact, even when I shortened my code down,

and trusted the printer's driver not to mess the stack up, it worked. All the usual Epson functions behaved exactly as the manual described them for Basic. I could even do a pseudo-VTAB function by POKEing values into \$25 (decimal 37).

There is just one anomaly: since the procedure does not use the normal Pascal BIOS (Basic I/O subsystem), the printer does not 'know' that it has been called, even though it is printing; so, to ensure that you get the sort of dump you want, you should do the necessary POKES after initialising the printer through the usual process of:

```
rewrite(fileid, 'printer:');  
and end up with: close (fileid);
```

The routine that does the job is listed here with comments. There are two things you can do with it after you have assembled it. First, keep it as a separate code file, declare it as EXTERNAL in any program that uses it, and LINK it into your compiled Pascal program by hand. Secondly, you could make it into a UNIT (like Turtlegraphics or Applestuff).

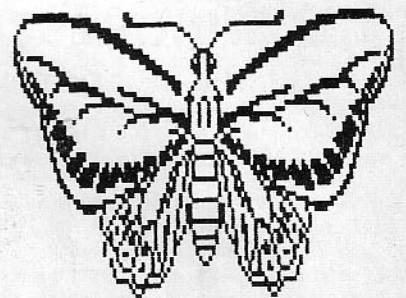
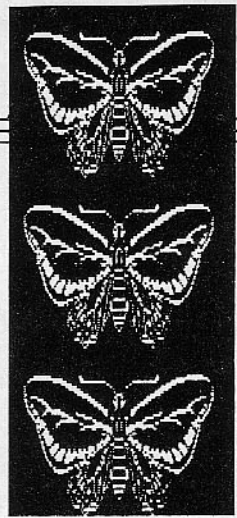
Since the Pascal manual doesn't actually tell you how to turn machine code routines into UNITS, I have written out the necessary Pascal code to produce a UNIT (no. 23) called EPSONDUMP. Copy and compile it, then use SYSTEM.LINKER to LINK the machine code into it (EPSONDUMP will be the HOST, and the machine code will be the LIBRARY file for it). Finally use LIBRARY.CODE to copy it into the SYSTEM.LIBRARY. After you have done this, you will be able to make use of procedure 'dump' simply by putting USES EPSONDUMP at the top of your program.

Although I wrote this code for the MX80, I would not be very surprised if it worked with the MX100. However, even if it doesn't, the method should be similar, so I hope this article has given enough hints to make it possible for someone else to do the job for other printers. For those of you with MX80s, though, have fun... it's very easy to make pretty pictures in Pascal.

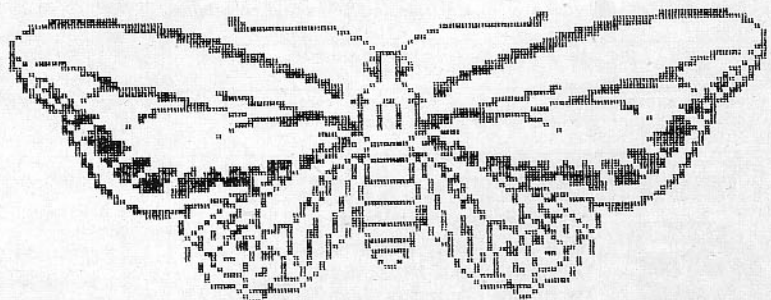
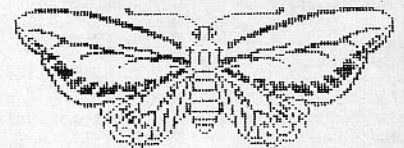
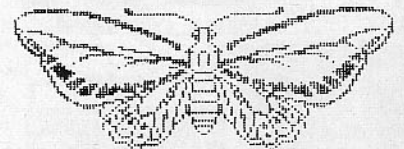
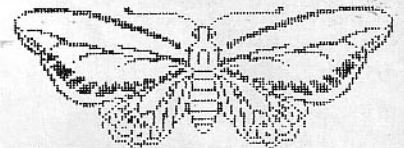
```
program epsondemo;  
uses peeklib, epsondump, turtlegraphics;  
  
var  
    f:file of char;  
  
begin  
    initturtle;  
    moveto(0,0);  
    pencolor(white);  
    moveto(0,180);  
    moveto(60,180);  
    moveto(0,0);  
    rewrite(f,'printer:');  
    poke(1913,1);           (*page 1, normal*)  
    poke(1145,76);        (*double density*)  
    dump;  
    close(f)
```

*This simple Pascal program shows how to use the procedures*





*Making pretty pictures in Pascal: Top – inverse, double density; above – double size, double density; below – normal (default); bottom – double size.*



```

:
:      Epson MX 80F/T
:
:      Interface to allow ctrl-q to act
:      as the HGR-dump instruction from Pascal
:
ctrlQ  .EQU 91
danger .EQU 279
print  .EQU 0C102      ;The address for a printer in slot 1

      .PROC dump
      LDY #1B
loop1 LDA danger,Y      ;The block from $279 to $294 is
      STA storage,Y    ;used by the HGR dump routine
      DEY              ;so save it locally.
      BPL loop1
      LDX #1           ;X = 1 for an Epson in slot 1
      LDA #ctrlQ      ;The Epson code to HGR dump
      JSR print
      LDY #1B
loop2 LDA storage,Y    ;Retrieve $279 to $294
      STA danger,Y
      DEY
      BPL loop2
      RTS

storage .BLOCK 1C      ;Temporary local storage

      .END

```

```

(**$S**)
unit epsondump;
intrinsic code 23;
interface
  procedure dump;

implementation
  procedure dump;
  external;

begin
  (*no initialization*)
end.

```

## Interactive editor - assembler

APPLE users should be familiar with Basic, in the sense that they will be able to enter and run Basic programs. They will also probably know that the program they write is not the one that actually activates the central processing unit of the Apple, the 6502.

Basic programs are translated, or "interpreted line by line" into machine code, by a systems program that is resident in the Apple's memory at all times. The Applesoft program acts as a systems intermediary, enabling the end-user to think in higher level terms, without having to know precisely what the computer is doing. The user is concerned with the results and not with the means. The systems software frees him from tedious considerations which are not related to his task.

The limitations of such hand-holding software may be negligible in many applications, where repetition, speed or complexity of calculation is relatively small. In other cases, the Applesoft program's laborious process of interpreting each line of Basic program into its machine language equivalent may make it wholly unsuitable for applications with frequent iterations, or where speed is important. In such circumstances the programmer will want to bypass the cumbersome translation and program directly in machine code.

However anyone who has written programs in raw machine code will know that it is an exercise fraught with pitfalls and excruciating detail - the very matters that the interpreter handles for the Basic programmer.

There are two ways of alleviating the problem. One is to purchase a compiler,

**This series by SEAN OVEREND takes the lid off assembly language and machine code programming, by describing the purpose and structure of a sophisticated assembler written in Basic.**

which translates the whole of the Basic program into machine code at one go. Using this technique, the user's Basic program is treated as an input data file by the compiler program, which when run generates an output data file containing the complete machine code equivalent. The compiler thus converts one set of data, the "source code", into another, the "object code", which can then be run as a machine code program without the Basic interpreter. Substantial time-saving can be achieved by this method.

Compilers can be expensive. They can also produce inefficient or unnecessary machine code, and they require complete recompilation runs if the slightest change in the original Basic program is made.

The second alternative is to identify which parts of the offending Basic program are responsible for the slow progress of execution and write them in machine code, using for the task an interim level language called assembly language. This language is closely related to machine code, and is converted to machine code by an assembler, which fulfills the same function as a compiler, namely translating source code (assembly language) into object code (machine code).

The advantage of the assembler over the basic compiler stems from the nature

of assembly language itself. Using assembly language, the programmer is able to dictate precisely what machine code will be generated. He is not dependent in any way upon choices built in to the Basic compiler. He has complete control over memory utilisation and is entirely responsible for the efficiency of the object code. The assembler's task is simply to relieve the programmer from as much of the tedium of detail and housekeeping as is consistent with maintaining overall control of the object code.

An assembler is thus a utility systems program that generates machine code output under total control of the programmer using assembly language input.

Before defining a machine code program let us look first at a Basic and at a Monitor program. A Basic program is initially entered into the Apple's memory through the keyboard. Once there, it may be executed using the RUN command. Subsequent reruns are only possible if the program is still in memory, either because it has not been erased, or because it has been reloaded. The RUN command is a keyword of the resident Applesoft program, which takes care of such minutiae as where precisely in memory the program is currently located.

The Monitor program is another built-in program, entered from Applesoft by use of the Applesoft command CALL -151. This program enables a user to look directly into the memory of the Apple and to alter its contents. Each unit of memory is called a "byte" and consists of eight bits of information in binary form. The contents of each byte of memory is displayed by the Monitor program as two hexadecimal characters. Each byte of memory is given a numeric address, starting at byte 0 and ascending (see Table 1).

A machine code program is information stored in a series of bytes of memory, the contents of which are fed in sequence to the 6502 central processor.

Two things follow from this - there must be some means of telling the 6502 where to start the memory reading process and secondly, the information stored in each byte of memory must be of such a nature that the 6502 "understands" it.

Using the Applesoft program, the 6502 can be directed to start machine code execution by the command CALL, followed by the decimal address of the start of the machine code information in memory. Using the Monitor program, the same can be achieved by entering the

DEC	HEX	
== ==	Hi-byte:Lo-byte Page:Offset	== : == ==
0 255	\$00:00 \$00:FF	} PAGE ZERO
256 511	\$01:00 \$01:FF	} PAGE 1
65280 65535	\$FF:00 \$FF:FF	} PAGE 255

Table 1

hexadecimal address of the start of the machine code information, followed by the Monitor command G. Thus CALL 768 has the same effect in Applesoft as 0300G from Monitor. Both cause the 6502 to commence execution of the machine code information which the 6502 expects to find starting at the beginning of page three of memory.

The information expected by the CPU must be either an opcode or an operand. An opcode is numeric information that is interpreted by the CPU as a command to perform one of its available operations, such as read in data from memory, add the contents of certain internal registers and so on. An operand is also numeric in form and is treated either as raw data by the CPU when carrying out the command specified by the preceding opcode, or is interpreted as a means of locating a memory address, the contents of which will be used or changed by the CPU when executing the opcode command.

Both the opcode and the operand consist of binary numeric information. To distinguish them the CPU simply assumes that the first byte of information is an opcode, then calculates how many operands are to be expected after that opcode, and so it is able to identify the next opcode. The 6502's internal architecture contains a built-in table of the number of operands that must be associated with each opcode (which will be a maximum of 2 and a minimum of 0).

A machine code program is thus information stored in a string of bytes of information, which the 6502 reads in sequence, all the time assuming that the rigorous but simple rules which differentiate opcode from operand have been complied with.

It would be wrong to assume that an assembler is all-singing, all-dancing. To become an assembly language programmer one has to be totally familiar with the various types of operations that the 6502 can perform and also with the various addressing modes. Control over the object code generated by the assembler is a meaningless attribute if the matter subject to control is not understood.

While a full discourse on machine language programming is beyond the scope of this series, certain aspects are relevant to a full understanding of the structure of an assembler. It is particularly important to understand the difference between a machine code opcode and an assembly language opcode.

The 6502 expects to be fed a series of opcodes and operands. Some opcodes are commands that do not require any further data to be read into the 6502, since its own internal registers already contain sufficient information for the operation to be performed. An example is the instruction "EA", which means "do nothing for one cycle". Another example is "6A" - "rotate one bit right."

Other operations do require further information to be fed into the 6502, such as "A2", meaning "load the contents of the next following byte into the X register", or "C5" - "compare the contents of the accumulator with the contents of the

10	0300	INFO	EQU	768	*0300
20	001A	ZP1	EQU	*1A	WITHIN ZERO PAGE
30	0301	ADR	EQU	769	*0301
40	001B	PRASE	EQU	*1B	WITHIN ZERO PAGE
50	001D	PTR	EQU	*1D	WITHIN ZERO PAGE
60	0302	VECTR	EQU	770	*0302
70	0304	START	EQU	772	*****
80	0304	EA		NOP	T=1 IMPLICIT
90	0305	6A		ROR	T=2 ACCUMULATOR
100	0306	90 15		BCC EXIT	T=3 RELATIVE
110	0308	AB 00 03		LDA INFO	T=4 ABSOLUTE
120	0308	C5 1A		ZCMP ZP1	T=5 ZERO-PAGE
130	030D	A2 FF		LDX #FFF	T=6 IMMEDIATE
140	030F	1D 01 03		ORA ADR,X	T=7 ABSOLUTE/INDEXED BY X REGISTER
150	0312	19 01 03		ORA ADR,Y	T=8 ABSOLUTE INDEXED BY Y REGISTER
160	0315	35 1A		ZAND ZP1,X	T=9 ZERO-PAGE/INDEXED BY X REGISTER
170	0317	96 1A		ZSTY ZP1,Y	T=10 ZERO-PAGE/INDEXED BY Y REGISTER
180	0319	61 1B		ZADC (PRASE,X)	T=11 INDIRECTION OF IMBEDDED ADDRESS USING X REGISTER
190	031B	91 1D		ZSTA (PTR),Y	T=12 INDEXED INDIRECTION USING Y REGISTER
200	031D	6C 02 03		EXIT JMP (VECTR)	T=13 ABSOLUTE INDIRECTION
0	0300				
1	0300	INFO	#0	0300	
2	001A	ZP1	20	001A	
3	0301	ADR	30	0301	
4	001B	PRASE	40	001B	
5	001D	PTR	50	001D	
6	0302	VECTR	60	0302	
7	0304	START	70	031D	
8	031B	EXIT	200	0320	
TOTAL BYTES 28					

Listing 1. The 13 6502 addressing modes are shown in lines 80-200

memory location whose address is contained in the next single byte of program".

It is plain from these examples that each single machine code opcode does two things. It specifies the operation that is to be carried out by the CPU, and also the way in which a given memory location (identifiable numerically from the succeeding operands, if any) is to be treated during the operation in question. The machine code opcode thus specifies both the operation and the "addressing mode" to be used during the operation. Each succeeding machine code operand is purely numerical, both in form and in information content, and does not specify what addressing mode is to be used.

The 6502 can perform 50 operation types and there are 13 different types of addressing mode, not all of which are available to each operation. The maximum number of combined alternatives amounts to 150, which is the number of machine code opcodes.

Assembly language, being directly related to machine code, also consists of operands and opcodes, which together provide the same information as their machine code counterparts. The split of information between the two is different, however, in that an assembly language opcode only contains information which specifies the 6502 operation, while the assembly language operand identifies the addressing mode to be associated with the 6502 operation. The program material, in the following machine code and assembly language lines, is identical:

	Opcode	Operands
M/Code	AD	00 03
Assy. Lang.	LDA	\$0300

The machine code opcode AD means "load the accumulator with the contents of the address specified by the next two bytes of program" (which will be in the

order lo-byte/hi-byte, also called offset/page). The machine code opcode thus specifies both the operation "load the accumulator" and the addressing mode "absolute" (which means "use the contents of the address which follows").

The assembly language opcode LDA merely specifies the operation "load the accumulator". It is the assembly language operand, or more precisely, the way in which it is written, that specifies the addressing mode "absolute". Examples of all 13 addressing modes are set out in Listing 1, which is described below.

Thus the use of an assembler does not obviate the need to understand the fundamental operations and addressing modes of the 6502.

Listing 1 is an example of assembled hardcopy output produced by the author's own assembler. It contains the following information, which will be developed as the series progresses:-

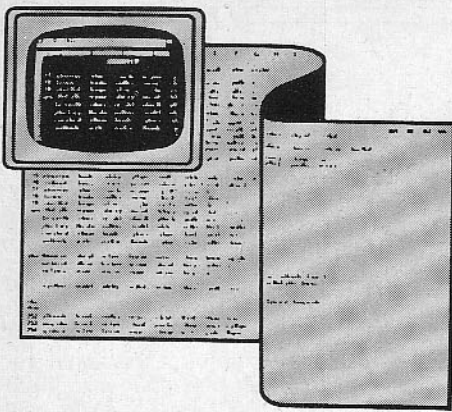
Column 1: Line number of the assembly language source code  
 Column 2: Initial memory address for M/C code in cols 3/4/5  
 Column 3: Machine code opcode  
 Cols 4/5: Machine code operands  
 Column 6: Assembly language label  
 Column 7: Assembly language opcode or pseudo-opcode (ignore initial "Z"s)  
 Column 8: Assembly language operand  
 Column 9: Comment, to assist assembly language programmer.

*Recommended reading: Apple Reference Manual pp 40-45 and (lightly) Appendix A.*

*Also Zaks, Programming the 6502, pp 38-58.*

*The author's interactive, 2 pass, disc-based editor-assembler is too long to publish in the series. It is available to readers by sending an initialised (48k DOS 3.3) disc, together with £15 handling charge to Sean Overend, 22 Highland Road, Amersham, Bucks, HP7 9AX.*

● Next month Sean Overend defines assembly language and explains some of its features.



# VC magic for your Christmas

CHRISTMAS brings the party-going season, which could create a few problems if you are the kind of person who finds that going to parties interferes with your Visicalc practice and the recreational enjoyment that you get out of it. In that case, have you considered taking your Visicalc with you to the party? (Not forgetting, of course, your personal computer.)

That was good lateral thinking! But what excuse would you possibly have for turning up with an uninvited Visicalc guest? The answer to this is that you are going to use it to find out how many people at the party have their birthday on the same day (ie, the same day as each other, not the same day as the party). If you think that this sounds daft and are worried that your friends will suspect you are becoming a VC paranoid, let me see if you would be prepared to change your mind after allowing me to test one of your

By NICK LEVY  
Principal,  
Interface Management

hunches and compare your guess with a Visicalc prediction.

Would you like to guess how many people must there be in a group before you can be *more than 99 per cent certain* that there are at least two people in that group with their birthday on the same day?

Fifty people? 100? 150? 200? 250? 300? Try and make a guess before reading on.

Now would you believe that if you have a gathering of only 57 people, then there is more than 99:100 chance that two of the people present could have their birthday on the same day! Furthermore, would you believe that if there are more than 22 people in a room then there is more than 50-50 chance that two of the people present have their birthday on the same day?

So if you think that this sort of useless information is just the thing for your party why not put it to the test and at the same time enhance your reputation as a kind of a VC magician (Very Clever or Very Crazy).

These odds were of course calculated by Visicalc, so how does it work out such probabilities and how do you go about presenting your findings? To begin with, you load the file shown in Exhibit I. Make sure that only the framed part of the model appears on your screen (so always save your model with either cell F7 or cell G7 in the top left hand corner on your screen).

The part of the model which your audience doesn't see is a table that Visicalc looks up when you make an entry in cell I16. The table itself is tucked away in columns B, C and D of your model.

This model, by the way, is not intended to be used just for entertainment. It could also be put to more serious business

applications in quality control or market research, or whenever it is necessary to calculate the probability that there are no repetitions in a sample, drawn with replacement, from a population of a given size.

In statistics the term 'population' does not necessarily always mean people. It could be almost anything, vegetable, mineral, animal or abstract.

Before going into a discourse on how the Visicalc @LOOKUP function works, let's see how the model was constructed.

Enter 1 in cell B12. In cell B13 enter 1+B12. Replicate the formula in cell B13 from cell B14 to cell B73. In answer to Visicalc's prompts 'No change' or 'Relative', respond with R. Next enter 1 in cell C12. Move to cell C13 and enter the formula: +C12\*(1-(B12/C5)). Then replicate that formula from cell C14 to C73.

In response to VC's prompts, key in R R and N. Next move to cell D12 and enter the formulae 1-C12 and replicate that formula from D13 to D73 (R).

The formula in cell I22 is a little bit more complicated. The main reason for this is because sooner or later some clever dick friend or colleague will test the model to see what will happen if there are, say, 23.5 people in the room. VC will respond with NA (not applicable). But it won't do that unless you instruct it to do so.

Alternatively, you could program your VC to reply with ERROR or FALSE, but you cannot instruct it to make any rude remarks to such silly questions. So the formulae in cell I22 is:

```
=IF(@AND(I16>1,I16=@INT(I16)),100-(@LOOKUP(I16,B12...B72)*100),@NA)
```

To make use of the model for other applications you only have to change the figure in cell C5 to suit the size of the population from which you are going to draw your sample. At the same time you can also shorten the table which Visicalc looks up or extend it by using the replicate commands described above.

If your lookup tables are more than 250

	A	B	C	D	E	F	G	H	I	J
1	PROBABILITIES OF REPETITION OR NO REPETITION IN A SAMPLE:									
2	DAYS PER YEAR: 365 (POPULATION)									
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										
54										
55										
56										
57										
58										
59										
60										
61										
62										
63										
64										
65										
66										
67										
68										
69										
70										
71										
72										
73										

Exhibit I

# party

FILENAME 1--AGING ACCOUNT ANALYSIS

C08130	F018SUM B8...E8)			
D08156	E01150 F018SUM B9...E9)			
E08125	F1018SUM B10...E10)			
F111210	F1118SUM B11...E11)			
G121140	F1218SUM B12...E12)			
H131180	E13115H013 F1318SUM B13...E13)			
I141120	F1418SUM B14...E14)			
J151290	F1518SUM B15...E15)			
K1618SUM B7...B16	C1718SUM C7...C16	B1718SUM D7...D16	E1718SUM E7...E16	F1718SUM F7...F16
B0814817/F17*100	B0814817/F17*100	B0814817/F17*100	B0814817/F17*100	B0814817/F17*100
B2514817	C251/F#2	D251/F#H0254825/100	E2514825-105	
B2614817	C261/F#5	D261/F#H0264826/100	E2614826-106	
B2714817	C271/F#10	D271/F#H0274827/100	E2714827-107	
B2814817	C281/F#15	D281/F#H0284828/100	E2814828-108	
B2918SUM B24...B29)	D3018SUM D24...D29)	E3018SUM E24...E29)		

Exhibit IV

rows long, there is also a technique to instruct Visicalc to look up a table which stretches over a number of columns. This technique, which consists of a number of commands nested together, is not explicitly mentioned in the manual.

Note that our lookup table (Exhibit I) will not give you precise answers for sample sizes of between 61-365, but who cares when in all these cases the probabilities are more than 99.50:100.00 - that is, only minutely less than certainty.

A final note about your Christmas festivities. If at a small party with about a couple of dozen participants there is a proportionally high number of people with

figures which are outside the screen you are looking at.

Here is how Visicalc does the trick. It looks at the figure in cell G5 (6) and compares it with the figures in column B. When it finds in column B the value which is either equal or the nearest smaller number to the value in cell G5 (in this example the figure 6 in cell B10) Visicalc will respond by entering the figure from the adjacent cell in column C (in this example the figure 21 which appears in cell C10). It's all in pages 2-73 of your manual.

In cell G11 we have a @CHOOSE command; an instruction to look at the figure in cell G5. Bearing in mind the figure in cell G5 (the figure 6), the operation moves to location D5. There it is instructed to report the content of the sixth cell down column D, starting from D5 (the content of cell D10). Note that had there been in cell D10 a word instead of a figure, then Visicalc would have responded with a zero.

You will find an entry of nine lines describing the CHOOSE command on page 3-77 of your manual.

There are only two formulae in the model Exhibit II (in cells G9 and G11). The rest of the model can be copied as shown. In order, however, to reject any entries keyed into cell G5 which are not whole numbers between 1 and 12 it becomes necessary to elaborate on these two formulae.

The formulae in cell G9 is:

```
@IF(@AND(G5>.999999999999,G5
<12.0000000001,G5=@INT(G5)
),@LOOKUP(G5,B5...B16),@E
RROR),
```

What the above formula stipulates is that the figure entered in cell G5 must be a whole figure (G5=the integer part of G5), or larger than .999999999999, or smaller than 12.0000000001.

If these three conditions are not fulfilled VC will respond with ERROR. If they are fulfilled then VC will carry out the @LOOKUP command. So if anyone tries to find out how many days are there in

month 4.15 or month 13 or month .99, VC will not be caught.

The formula in cell G11 will be executed only if there is no ERROR message in cell G9. If there is an ERROR message in G9 then the same message will also appear in G11. The formula in G11 reads:

```
@IF(@ISERROR(G9),@ERROR,
@CHOOSE(G5,(D5...D16)),
```

It is important that you understand the structure of the VC's @IF command. It must always consist of three parts with the first and second parts separated by a comma, and the second and third parts also separated by a comma. (See page 3-80 in your VC manual.)

Note also that @ISERROR (G9) means cell G9 shows an ERROR message. Remember what I said earlier - if all this proves to be too much for you there is nothing to stop you from looking up the tables yourself instead of asking Visicalc

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											

Exhibit II

the same birthday, then you must have been invited to a special Christmas party for Capricorns (people born between December 21 and January 19.)

To understand how Visicalc's @LOOKUP as well as @CHOOSE commands work, let us look at Exhibit II. All you have to do with this model is to enter a number between 1 and 12 in cell G5, and you will be informed in cells G9 and G11 of the number of working days and pay days corresponding to the month in cell G5.

You could of course be cynical about the whole thing and say: "Why ask Visicalc to look up a table which you can look up yourself?" And you will be perfectly right. So only use this technique when you have to search long columns of

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						

Exhibit III

# VISICALC

to do it for you.

In the meantime, as promised last month, here is another datagramme for your Visicalc. But first it is necessary to introduce the model on which our datagramme is going to work:

Exhibit III is an aging account analysis. A listing of the key figures and formulae is given in Exhibit IV. Copy that model and save it under the file name AGING A/C, or any other name that you choose.

The model should be self explanatory, but note the bottom part under the heading bad debt analysis. Column B between B25 to B30 is a copy of row 17. In column C you estimate the percentage of bad debts that you are likely to incur for each group of debts outstanding. In column D you convert and express these percentages in money terms, and finally in column E you calculate the amounts that you are likely to receive after allowing for the bad debts in each group.

As you can see from the example, a total amounts due of £3806 can easily turn out to be no more than £3646 (cell E30).

The datagramme for updating the aging account is shown as Exhibit V. (Note that the whole datagramme should be entered in a single column - not in two columns as shown in the exhibit). For the benefit of readers who have not seen the previous article on datagramming I shall briefly repeat the steps necessary to create that datagramme.

Make your VC columns 18 spaces wide (/GC18). Next right-justify 40 cells in any one VC column. You can do this by right-justifying a single cell (/FR) and then replicating the format down the column. Just in case it surprises you, you are indeed at this stage formatting and replicating empty cells.

Next copy Exhibit V into the right-justified cells you have just created. Remember that each entry is a label in the VC sense, so each entry must start with either a " or Q ESC.

When you have finished copying the datagramme you have to save it in print format. To do this put your cursor on the first entry (the cell with the entry: >E8#+D8#) and type /PF.

In response to VC's request for a file name, enter UPDATE AGING or any other file name that appeals to you and press

RETURN. Finally place your cursor on the last entry in the datagramme and again press RETURN.

It would be a good idea to save that datagramme also with the /SS command or with the DIF command (/S#S) using different file names. Such files will help you to correct any mistakes in your datagramme, which will only become apparent after it is saved in print format and you discover that it doesn't work.

If this happens go back to your datagramme saved under the /SS file or the DIF file, make the necessary correction and save the correct version in print format as before (replacing the previous PF datagramme file).

Remember that you cannot look at a PF file after it has been saved unless you use a special utility program that can perform that operation.

Let's now see the updating of AGING A/C in action. First /SL your AGING A/C (Exhibit III) and then /SL the file UPDATE AGING which you saved in print format. Your screen should now look like Exhibit VI. Enter a new month in cell B3 and copy into that model the figures from Exhibit VII. Again /SL UPDATE AGING, and your screen should look like Exhibit VI, but with the data updated.

Note that with Exhibit VI on your screen, instead of just copying Exhibit VII you can put your cursor on any of the figures in columns C,D and E, between row 7 and row 16, and perform the following operation.

Say you want to deduct 75 from 130 in cell D8, put the cursor on D8 and type #-75 RETURN. The 130 will become 55 (130-75). The # freezes figures in their VC cell (see page 2-68 in your manual). This is a very useful command which I could not find in Supercalc.

Finally, here's another VC mystery revealed - and you deserve it if you have managed to read this article to the end.

Have you tried the command /XV15 RETURN followed by /GC6 (followed by a few pressings of ;)?) Try it, clear the screen and now try /XH9 RETURN followed by /GC6. What can you make of it?

Save yourself time. Don't look up your VC manual for any of these /XV and /XH commands. 🍎

**DATAGRAMMING FOR AGING  
/GC18 BEFORE SAVING**

5	>E8#+D8#	21	>D15:0+C15#
6	>E9#+D9#	22	
7	>E10#+D10#	23	>C8:0+B8#
8	>E11#+D11#	24	>C9:0+B9#
9	>E12#+D12#	25	>C10:0+B10#
10	>E13#+D13#	26	>C11:0+B11#
11	>E14#+D14#	27	>C12:0+B12#
12	>E15#+D15#	28	>C13:0+B13#
13		29	>C14:0+B14#
14	>D8:0+C8#	30	>C15:0+B15#
15	>D9:0+C9#	31	
16	>D10:0+C10#	32	>B8:" *
17	>D11:0+C11#	33	>B9:/R:B9,B15:
18	>D12:0+C12#	34	/X:A1
19	>D13:0+C13#	35	>B3:" .....
20	>D14:0+C14#	36	/X!

Exhibit V

A	B	C	D	E	F
AGING ACCOUNTS ANALYSIS					
FOR THE MONTH					
ENDING -> .....					
CUSTOMER'S NAME	NOT YET DUE	PAST DUE 30-59	PAST DUE 60-89	OVER 90	TOTAL DUE
REGTON C C	0.00	130.00	0.00	0.00	130.00
EL ELECTRIC	0.00	0.00	306.00	0.00	306.00
EYE SEE EYE	0.00	125.00	0.00	0.00	125.00
BRAKE & CO.	0.00	0.00	210.00	0.00	210.00
EATON TOOLS	0.00	0.00	140.00	0.00	140.00
FLEET GARS	180.00	0.00	0.00	0.00	180.00
CLUBE TRAVEL	0.00	120.00	0.00	0.00	120.00
MILL'S PAPER	250.00	0.00	0.00	0.00	250.00
TOTAL	0.00	270.00	375.00	671.00	3016.00
%	0.00	72.52	9.95	17.63	100.00
BAD DEBTS ANALYSIS					
WHEN DUE (DAYS)	TOTAL AMOUNT	PERCENT BAD DEBTS	AMOUNT OF BAD DEBTS	AMOUNT DUE	
NOT YET	0.00	2.00	0.00	0.00	
30 - 59	270.00	5.00	130.00	2822.00	
60 - 89	375.00	10.00	37.50	337.50	
OVER 90	671.00	15.00	100.45	570.55	
TOTAL	3046.00		267.95	3278.05	

Exhibit VI

A	B	C	D	E	F
AGING ACCOUNTS ANALYSIS					
FOR THE MONTH FEB. '82					
ENDING -> .....					
CUSTOMER'S NAME	NOT YET DUE	PAST DUE 30-59	PAST DUE 60-89	OVER 90	TOTAL DUE
REGTON C C	250.00	0.00	65.00	0.00	315.00
EL ELECTRIC	0.00	0.00	206.00	0.00	206.00
EYE SEE EYE	0.00	125.00	0.00	0.00	125.00
BRAKE & CO.	300.00	0.00	0.00	0.00	300.00
EATON TOOLS	130.00	0.00	0.00	0.00	130.00
FLEET GARS	140.00	180.00	0.00	0.00	320.00
CLUBE TRAVEL	50.00	0.00	45.00	0.00	95.00
MILL'S PAPER	80.00	820.00	0.00	0.00	900.00
TOTAL	480.00	1000.00	255.00	486.00	2221.00
%	73.64	15.22	3.58	7.35	100.00
BAD DEBTS ANALYSIS					
WHEN DUE (DAYS)	TOTAL AMOUNT	PERCENT BAD DEBTS	AMOUNT OF BAD DEBTS	AMOUNT DUE	
NOT YET	480.00	2.00	96.00	4743.20	
30 - 59	1000.00	5.00	50.00	950.00	
60 - 89	255.00	10.00	25.50	211.50	
OVER 90	486.00	15.00	72.48	413.52	
TOTAL	2221.00		244.98	2476.98	

Exhibit VII

# Taking up DATA faster

**t** Programs which access a lot of DATA in the READ command, especially those which continually re-READ DATA depending on options, can be considerably speeded using an idea from **W.A.J. Folkett**, of Nottingham. Enter the DATA statements at the beginning of the program but type REM rather than DATA. Group together all that DATA for option A, that for B etc, and fill up lines as much as possible (see sample program, right).

Now add (after these lines and before the main program) these two lines:

a) FOR A = 2048 TO (NUMBER OF REM LINES \*255)+2048 : IF PEEK (A) = 178 THEN PRINT A,

b) NEXT : END

RUN the program and make a note of the values printed. These are the locations where the REM token (decimal 178) is stored. If more values are printed than there are REM lines note only those first ones up to the number of such lines. The extras are caused by REMs in the body of the program or by line markers corresponding to 178, 178 + 256, 178 + 2\*256, etc, or possibly by REMs etc, left over from a previously stored, long program.

Having noted all the relevant

```

5 REM A-100,A101..ETC (UP TO MAXIMUM LINE LENGTH) )
10 REM A-200,A201..ETC (UP TO MAXIMUM LINE LENGTH) )
15 REM A-300,A301..ETC (UP TO MAXIMUM LINE LENGTH) )
20 REM A-400,A401..ETC (UP TO MAXIMUM LINE LENGTH) )
25 REM A-500,A501..ETC (UP TO MAXIMUM LINE LENGTH) )
30 REM A-600,A601..ETC (UP TO MAXIMUM LINE LENGTH) )

35 REM B-100,B101..ETC (UP TO MAXIMUM LINE LENGTH) )
40 REM B-200,B201..ETC (UP TO MAXIMUM LINE LENGTH) )
45 REM B-300,B301..ETC (UP TO MAXIMUM LINE LENGTH) )
50 REM B-400,B401..ETC (UP TO MAXIMUM LINE LENGTH) )
55 REM B-500,B501..ETC (UP TO MAXIMUM LINE LENGTH) )
60 REM B-600,B601..ETC (UP TO MAXIMUM LINE LENGTH) )
  
```

STATEMENTS TO BE PROCESSED AS DATA ON SELECTION OF OPTION 'A'

STATEMENTS FOR OPTION 'B'

locations arrange via programming to re-POKE the correct locations to the data keyword (decimal token 131) on response to specific questioning.

If the first six DATA statements refer to an option A then on receiving this answer re-POKE only the first six locations. For option B POKE the second six locations only etc.

When DATA is read this will considerably speed up the process time and if there are many DATA statements this saving in time will be very pronounced. On finishing with the DATA and upon return to the main

menu or to a position in the program that may require the selection of different batches of DATA, POKE again the locations back to REMs and issue the RESTORE command.

**WARNING:** Be sure not to change or add any extra line or detail before the last DATA (REM) entry as this will require a change of POKE locations. It is for this reason that all DATA is stored at the beginning to allow easy changing of the main programme. Also do not use the Applesoft programmer's aid to either compress the programme or renumber it.

## An unexpected bonus

**t** Sometimes a large Applesoft program stops for about a minute because the Apple is running out of room and needs to do a "garbage collection" to clear out all unused variables. To minimise the delay have a line 'AS=FRE (0)' which will force a garbage collection. If this is done inside the main loop of your program the delay should be reduced to between 5 and 10 seconds. Another useful line that indicates when the program is doing a garbage collection is:

```
1xxx IF PEEK (112) - PEEK (110) < 2 THEN
PRINT "STAND BY":A = FRE (0)
```

**t** APPLE II users who have one of the new RGB monitors from Microvitec may be interested to learn of an unexpected bonus from this rather exciting system. The problem of text on the high resolution graphics screen on a colour monitor is well known. It is difficult to read because white text can be purple, green or white depending on its vertical location on the screen. This results in text which is very difficult to read.

Microvitec have solved this problem by providing a two-colour

high resolution mode which gives beautifully clear graphics and text in any one foreground colour on any one background colour. These two colours can be chosen at will by a simple POKE statement from a set of 16. In this two colour mode, Apple HCOLORs 1,2,5 and 6 appear as a "hatched" version of the foreground colour. This can effectively give a "third" colour.

In this two colour mode, however, one effectively loses the normal Apple configuration of four high resolution

Statement	Microvitec Purpose
POKE -16143,0	Colour select enable
POKE -16136,15	Set text foreground to white
POKE -16134,15	Set high res. foreground to white

Table 1

**i** Pascal or Fortran not booting up? When Pascal is booted up it looks at slot 3. So if there is a card in slot 3 it assumes that this is a terminal and therefore sends all output there and expects all input to come from slot 3, thus locking out the keyboard. The only cards that are normally placed in slot 3 are 80 column cards.

## Stuck with 40 columns?

**i** If you are limited to letters printed with only 40 characters across the page you are using a parallel printer and the interface card defaults to 40 characters. The solution is to reset the print width by having 'CONTROL I 80N' as the first thing printed. In Applewriter 1.1 this has to be inserted into the Hello program (ie Line 3 = 'PRINT CHR\$(4);"PR#1":PRINT "CTRL-I 80N":PRINT CHR\$(4) ;"PR#0"); in Applewriter 2 the solution is to insert it in your 'TL' option on the print menu.

## from RGB monitors

colours 0-3 or 4-7.

This may be an important factor in certain circumstances, e.g. games, where the full four colours are required. In this case one must elect to run the system in the normal Apple configuration which results in the "multicoloured text" mentioned above. The set of POKE statements in Table 1 provide a "halfway stage", giving much more readable text while maintaining four colour graphics.

The command to enable to two-colour mode must not be given, i.e. POKE-16141,0. Provided that no "reset" or "power off" occurs this configuration will be unaffected. To re-enable the normal Apple configuration requires only POKE -16144,0.

If you require to run a different disc simply put it in the drive and RUN the new program. If you have a menu-

## Program writes itself into a text file

**i** This program shows how a program may write itself to a text file so that the text file may be used by word processors; some useful call statements which, in this example, create a top of screen logo for the display of messages; and, finally, a subroutine to centre strings.

Lines 100-140 write this program to a text file for the word processor

```
100 D$ = CHR$(4)
110 PRINT D$;"OPEN PROGRAM TEXTFILE"
120 PRINT D$;"WRITE PROGRAM TEXTFILE"
130 LIST
140 PRINT D$;"CLOSE PROGRAM TEXTFILE"
```

and lines 150-240 produce two bars at the top of the screen with a blank line in between ready for messages.

The call statements are as follows:

- 1184 Home + Apple logo
- 922 Line feed
- 384 Inverse
- 380 Normal

```
150 TEXT : HOME
160 CALL - 1184
170 CALL - 922
180 CALL - 384
190 PRINT"      40 blank spaces  "
200 CALL - 380
210 CALL - 922
220 CALL - 384
230 PRINT"      40 blank spaces  "
240 CALL - 380
```

Lines 250-260 tabs cursor to screen line 5 and locks the screen display. The CALL -868 clears to end of line.

```
250 VTAB (5): CALL - 868
260 POKE 34,5
```

Lines 290-310 are messages (N\$) to be displayed at the top of screen at the tab position defined by X, and

```
270 HOME
280 X = 3
290 N$ = "HELLO": GOSUB 350
300 N$ = "WELCOME TO": GOSUB 350
310 N$ = "THIS PROGRAM": GOSUB 350
```

Lines 320-340 reset cursor position and end program. Your program lines would normally be between 330 and 340.

```
320 X = 6
330 VTAB (X)
340 END
```

Line 370 is a subroutine to centre the N\$ messages at VTAB(X). The call on line 360 clears the VTAB line for the next N\$. Line 380 is a delay loop so that the N\$ message is displayed for two seconds

```
350 VTAB (X)
360 CALL - 868
370 PRINT TAB((41-LEN(N$))/2);N$
380 FOR N = 1 TO 2000: NEXT
390 RETURN
```

Sean Pendleton

Allan Dubost



# The games Apples play

Compiled by  
CLIFF McKNIGHT

Reviews and  
listings to  
brighten your  
festive season

WHEN Shakespeare said "the game's afoot", he obviously hadn't anticipated the coming of the microprocessor. If he had, he might have been tempted to say "the game's afinger" because all over the world fingers are avidly consigning space invaders to their doom, flicking pinball flippers, steering helicopters and generally wreaking seven kinds of havoc.

The last year has seen an enormous increase in the number of games available for the Apple. It has seen the introduction of the 'super-game' - Time Zone - which occupies six double-sided discs and is reviewed in this issue. It has also seen the introduction of a game in which the aim is to *save* extraterrestrials - Suicide. This game stands out not only as a fun game but also as one which reverses the tide of death, destruction and general xenophobia.

In addition to the games requiring digital dexterity, some games have taxed not only the ingenuity of the player but have drawn on his knowledge of the Apple's capabilities. The Prisoner is a case in point. Some even allow you to cheat if you know enough about the system, but I won't spoil your fun by saying which!

The ability of games to hold children's interests has been exploited by a variety of educational packages. These range from the Monster Maze type for practising maths to the more complex group games like Expedition to Saqqara. As micros become more common in the schools we can expect to see more of these types of applications.

What, then, does the coming year hold? No doubt the alien hordes will continue to threaten our existence and writers will dream up new adventure scenarios. Watch out for Watership Down, and remember, you read it here first!

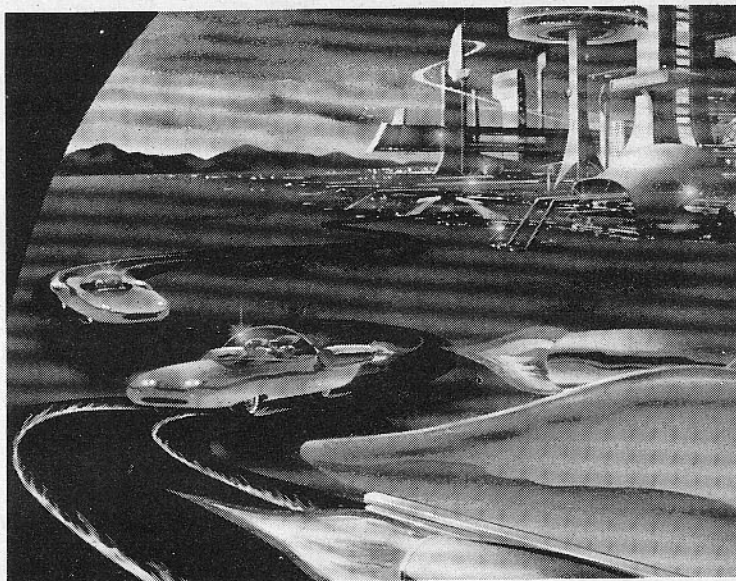
Personally, I'll be happy to see general proliferation. I enjoy playing games on the Apple and I like having a wide choice. I hope you do too, and I hope this issue gives you some ideas.

Merry Christmas!



# From space battles to deadly mazes ...

# those keys take a pounding



I THINK I'm going to bill *Windfall* for a new set of springs for my Apple keyboard – or at least certain selected keys. The reason for this is the amount of battering they have taken recently as a result of playing *Mars Cars*, *Crazy Maze*, *Space Kadet* and *Star Maze*, all in the line of duty. Mind you, if all my duties were as enjoyable as this batch of software I'd be in clover.

I'll deal with *Space Kadet* first because it doesn't conform to the same format as the other three. The program is described as a *Space Academy Battle Simulator*, which means that you, the Kadet, are the pilot of a spaceship which you fly around a matrix of corridors. As you do so various baddies enter the system and proceed to attempt your destruction either by shooting you or just simply crashing into you. Your movement is controlled by four keys (Q, A, S, D) and the Z key is your firing button. According to the instructions, CTRL-C should enable you to change the keys if this arrangement doesn't suit you, but this didn't work on the review copy.

Points are scored for the baddies you destroy, but you have a limited supply of torpedoes which replenish at a fairly slow

rate. Also, as time goes on, more baddies enter the arena so there is no point in just avoiding them – it is kill or be killed.

You start with 11 ships and the instructions claim that you earn an extra ship for each 1,000 points you score, up to a maximum of 16 ships at any one time. I find this incredible because the three avid gamers in our family have yet to achieve a score greater than 149. CTRL-C is also supposed to change the starting level, but since it didn't work on the review copy it is possible that we have been playing on one of the harder levels.

Even at this level the game is quite enjoyable, and can also be played by two Kadets simultaneously. In monochrome it is difficult to distinguish the pair, but in colour they are a bit easier to tell apart. The colour on the title screen is suitably gaudy, and the sounds (which can be suppressed) are of the usual blam-splat variety.

The other three games are all based on the concept of a series of mazes containing treasure and populated by hostile aliens. If you succeed in capturing all the treasure on a level, play moves to the next level until you run out of lives.

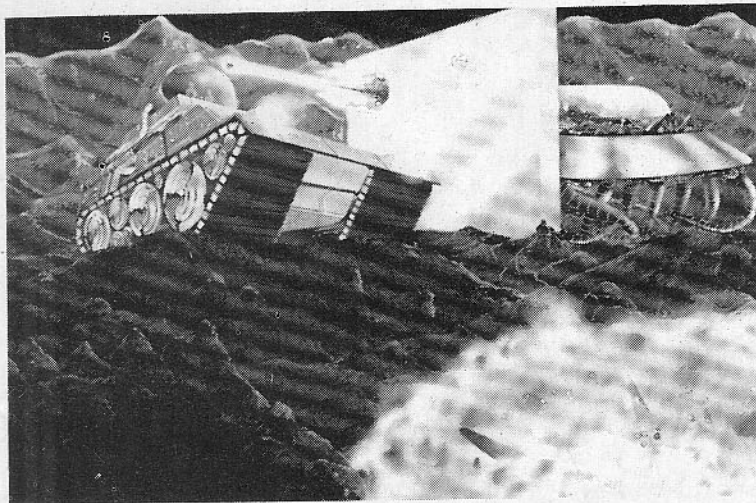
In *Crazy Maze* the mazes are based on square blocks with various passages blocked off. Level 1 has a single killer car to chase you as you collect the six bundles of loot and make it back to the transporter beam. Level 2 has two cars, and so on to the nineteenth level. If you can make the killer cars crash into each other they wipe themselves out and you gain points. They can also sometimes be trapped in some of the dead ends.

Once you have devised a strategy for a particular level it will almost invariably be successful. However, two options in the program stop this spoiling the fun. The first is the one to set the speed from a slow 7 to a rapid 1. On the slow speed I can get to level 9 most times, but on the fast speed I have difficulty getting through level 1. My four-year-old, on the other hand, can regularly get to level 4 on the fast speed. Maybe that is what the manual means when they say it will separate the men from the boys (even if she is a girl!).

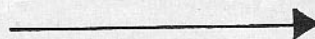
The other useful option allows you to set the starting level, up to level 9. This means that once you have definitely mastered the lower levels you don't have to work your way through them each time. Control keys can also be changed and the sound can be suppressed.

*Mars Cars*, although based on the same format, is a completely different game. On each level there are four bundles of loot, one in each corner of the screen. There are four varieties of baddy, each level being populated by a particular variety. Their movement is restricted by the walls of the maze, but you have to drive through the walls in order to get to the loot. You get points for each section of the wall you demolish, so you can either play to get a high score or play to get as high up the levels as possible. There are 16 levels, and I won't tell you what happens when you get through level 16, but it took me a long time to do so.

The game can be played using keyboard or joystick, and the sounds can be



*The tank that turns into a Renault*



turned off. However, turning off the sound has a subtle effect on the game. Relieved of the task of interspersing the action with noises, the program becomes much smoother and the game becomes faster (and therefore harder). Unlike Crazy Mazey, the baddies in Mars Cars have enough of a random element to prevent precise strategies being worked out. This means that I occasionally have games where I can't get past level 3.

Incidentally, the manual (a single folded A4 sheet) deserves a Hugo award for creative science fiction. For example, you have apparently "heard the tales whispered about wherever spacemen meet and you've been captivated by their descriptions of the priceless treasures of Mars". Where do spacemen meet? Also, the cover illustration shows a tank-like vehicle, but the car in the game looks disconcertingly like my little Renault!

Although Star Maze shares the same format, it is done on a much grander scale. The mazes are much bigger than a screenful and scroll in all directions so that you are effectively moving over the surface of a sphere. You not only have to control direction of movement but also speed, and you have to aim your bullets at the baddies. Some of these will ignore you unless you fire at them, but others will fire at you.

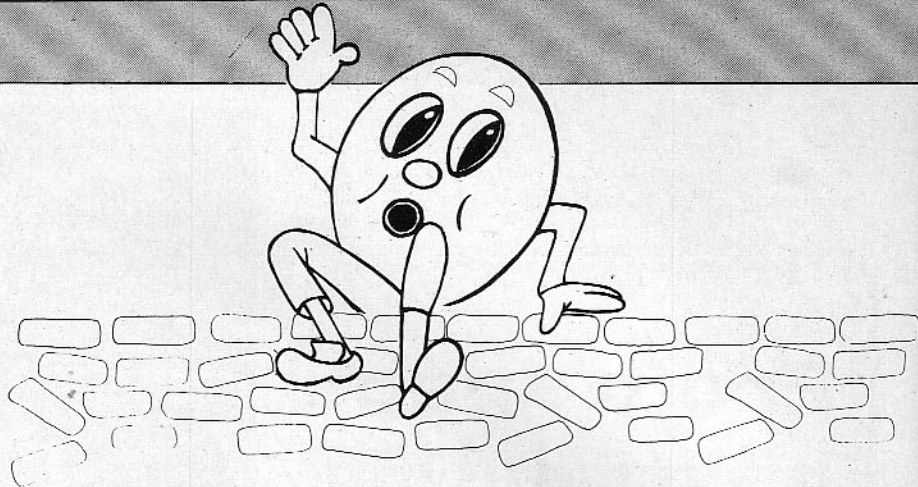
On each level there are nine power jewels to be captured and returned to the mothership before your fuel runs out. If that all sounds easy, it is further complicated by the fact that you have to have a speed of less than 200 to capture a jewel, and less than 100 in order to refuel at the mothership.

Two 'panic buttons' are available. A limited supply of antimatter bombs will clear the screen of all enemy objects, and a hyperspace button will transport you to somewhere else in the maze. However, hyperspace is tricky because you don't know where you will end up, and you might also find yourself going at uncontrollable speeds when you rematerialise.

Of all these games, Star Maze makes the best use of colour and sound. Playing in colour makes it easier to find your way round the maze because the various segments are coloured differently. Star Maze also requires a greater time commitment - hours rather than minutes - and so a 'save game' facility would have been useful. All four use the ESC key to freeze the game, which is very handy if the phone rings or you want to answer a different call.

I enjoyed playing all these games, but found myself playing Mars Cars most often when I wanted a short game or Star Maze if I had an hour to spare. I dread to think how long it would take me to get to level 16 of Star Maze because it took an hour to get most-of-the way through level 3 before being wiped out. Right... beam me aboard, Mr Spock.

**Cliff McKnight**



THIS game is played on the hi-res screen. It features Humpty sitting on top of a crumbling wall. To stop the wall's collapse and to prevent Humpty taking a tumble you have to position your "man" (using the games buttons) underneath the falling bricks and "blow" them back into place. The longer you keep Humpty up, the higher your score.

As there are a number of DATA lines to be keyed in, the following short program should be added to the end of the main program to facilitate checking.

```
1000 RESTORE! S=0 : ONERR GOTO 2000
1010 READ A: S=S+A
1020 GOTO 1010
2000 PRINT"SUM IS "S
```

Then when the command RUN 1000 is issued the checksum 13728 will be returned. If it is okay delete lines 1000,2000.

**Max Parrott**

```
10 GOTO 310
20 J = INT (E * RND (PEEK (BO
+ D)):Y1 = B(J):B(J) = B(J)
- E: IF B(J) < YL AND (J >
D OR J < SE) THEN C = C + O:
GOTO 20
30 IF C > F THEN GOTO 160
40 IF B(J) < YL THEN 20
50 X1 = FO + J * E1
60 DRAW D AT X,Y: HCOLOR= Z
70 IF PEEK (BO) > LI THEN DRAW
D AT X,Y: X = X - D: IF X < L
L THEN X = LL
80 IF PEEK (B1) > LI THEN DRAW
D AT X,Y: X = X + D: IF X > U
L THEN X = UL
90 SCALE= D: DRAW T AT X1,Y1
100 Y1 = Y1 + D
110 IF Y1 > Y THEN HCOLOR= D: SCALE=
O: S = S - O: GOSUB 150: GOTO
20
120 IF ABS (X - X1) < D AND Y1 -
Y < D THEN HCOLOR= D: SCALE=
D: B(J) = B(J) + E: DRAW T AT
X1,B(J): SCALE= D: S = S + O:
GOSUB 150: GOTO 20
130 HCOLOR= D: DRAW T AT X1,Y1: SCALE= D
140 GOTO 60
150 FOR K = T TO F: POKE A1,TN: POKE
A2,K: CALL A3: NEXT : RETURN
160 X = 140
170 FOR I = 28 TO 130 STEP F
180 TN = I: GOSUB 150
190 HCOLOR= Z: DRAW D AT X,I - F
200 HCOLOR= D: DRAW D AT X,I
210 NEXT
220 FOR J = O TO F
230 TN = 70 - 10 * J: GOSUB 150
240 SCALE= J: DRAW D AT X,I
250 NEXT
260 TEXT : HOME : VTAB 8: HTAB 1
2: PRINT "YOUR SCORE IS "S
VTAB 12: HTAB 7: PRINT "DO Y
OU WANT TO PLAY AGAIN ":
280 INPUT A#: IF A# = "NO" THEN
HOME : END
290 IF A# = "YES" THEN RUN 430
300 GOTO 270
310 HOME : HTAB 13: INVERSE : PRINT
"HUMPTY-DUMPTY": NORMAL
VTAB 10: PRINT "HUMPTY-DUMPT
Y IS STOOD ON A FALLING"
330 PRINT "WALL. USE THE GAMES
BUTTONS TO MOVE"
340 PRINT "YOUR MAN."
350 PRINT : PRINT "IF HE IS UNDE
RNEATH A BRICK HE CAN"
360 PRINT "BLOW IT BACK AGAIN."
```

```
370 PRINT : PRINT "HOWEVER. IF T
HE TOP COURSE FALLS...."
380 VTAB 23: HTAB 8: FLASH : PRINT
"PRESS SPACE BAR TO START":
NORMAL
390 FOR I = 24576 TO 24576 + 213
: READ J: POKE I,J: NEXT
400 FOR I = 770 TO 790: READ J: POKE
I,J: NEXT
410 POKE 232,0: POKE 233,6 * 16
420 GET A#: IF A# < > " " GOTO
420
430 BO = - 16287: B1 = - 16286: D
= 3: UL = 230: LL = 32: Z = 0:
LI = 127: T = 2: TN = 60: A1 =
768: A2 = 769: A3 = 770
440 Y1 = 124: YL = 69: S = 55: C = Z
: E = 11: O = 1: F = 6: SE = 7: E
I = 18: FO = 40
450 FOR I = 0 TO 10: B(I) = Y1: NEXT
460 HGR2 : HCOLOR= 3
470 ROT= 0: SCALE= 3
480 FOR I = 40 TO 220 STEP 18
490 FOR J = 80 TO 128 STEP 11
500 DRAW 2 AT I,J
510 NEXT : NEXT
520 SCALE= 1: DRAW 3 AT 140,22
530 X = 140: Y = 175
540 GOTO 20
550 DATA 3,3,8,0,55,0,64,0,45,44
.37,45,44,37,60,60,44,44,54,
54,54,54,54,55,62,54,55,62,4
7,109,73,41,61,39,60,36,39,6
0,36,36,36,36,36,53,53,55,55
.46,53,45,46
560 DATA 53,45,5,0,45,45,37,36,6
3,63,55,54,0,63,62,55,63,62,
62,62,54,55,62,54,55,126,73,
33,37,45,46,53,62,62,39,231,
219,51,62,54,55,54,54,54,54,
53,53,45,44,46
570 DATA 46,37,39,39,231,211,27,
54,53,53,46,46,54,53,46,54,5
4,55,62,54,55,55,55,46,46,
45,44,37,45,44,37,37,37,36,
37,44,53,46,54,46,46,46,53,4
5,46,53,45,37,37
580 DATA 60,60,60,60,36,39,60,36
,36,37,44,36,37,37,44,44,36,
223,216,58,62,62,46,37,37,53
,45,44,44,36,36,36,60,36,
39,252,219,58,62,39,39,44,37
,45,46,110,73,35,60
590 DATA 36,39,60,36,39,39,63
,60,39,63,5,0
600 DATA 173,48,192,136,208,5,20
6,1,3,240,9,202,208,245,174,
0,3,76,2,3,96
```

# Stop Humpty taking a tumble

# DEADLINE



YOU are standing at the entrance to the estate of industrialist and philanthropist Marshall Robner, who was found dead of a drug overdose in his locked library. You have a file of evidence under your arm and 12 hours to discover whether the dead man was murdered and, if so, by whom.

This is the theme of the latest game from the authors of the "Zork" adventure series. It is, to my knowledge, the first attempt to produce a computerised "whodunit".

How well does it succeed? First, the presentation of the game is excellent. Included with the disc (DOS 3.3, 32k minimum) is a document wallet sealed with a label reading "Documentary Evidence", and containing records of interviews with the prime suspects, a forensic report, a lab analysis report and other items. Also included is an instruction manual disguised as an Inspector's Casebook.

The game is auto-starting on booting the disc. There is no use of graphics; like the original Adventure games, the proceedings are described using text only.

At each halt in the action the computer describes your surroundings, who is present in the room and whether anything has been moved or altered since you were last here. If a printer is attached, you may have a printout of the dialogue on request – useful for keeping track of events.

You can move around the house and grounds, picking up and examining anything of interest. There is a well-equipped fingerprint and analysis laboratory available for more thorough investigation of objects, but this does take time.

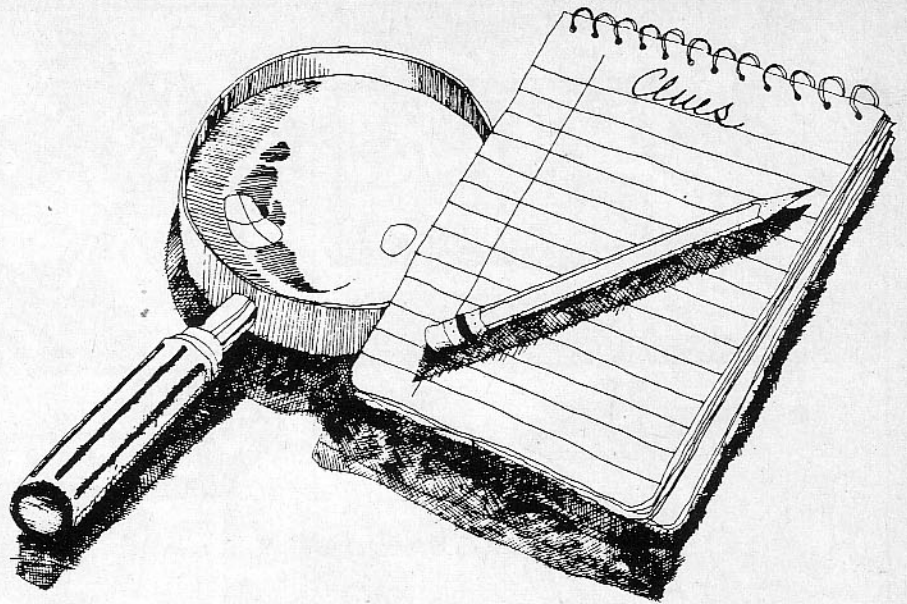
The game is played in "real time" but not actual time. The game clock advances by one or two minutes for each action you perform and you are limited to 12 hours of game time to complete the task. The makers state that, on average, 20 hours of actual time are needed.

As well as examining objects you may question the people concerned, and this brings us to the main strength – and weakness – of the game.

The program is designed to understand simple English, so input to the computer consists of statements like "Walk into the library", "Pick up the revolver" and "Look behind the curtain".

This really does work, and it is uncanny to sit and "talk" to a computer in this way. The authors claim a vocabulary of 600 words, and this appears sufficient to understand most commands concerning actions. However, the problems come when trying to talk to people using similar phrases.

According to the instructions, most questions relevant to the investigation have been answered in the interview reports, but you soon discover that this is not so. I found that simply finding words which the program would accept, to ask a simple question of a character, was the most time-consuming and frustrating part of playing the game.



## Enter Holmes, computing clues

The other main fault appears to be a lack of debugging. Quite often I received wrong responses to simple questions. For instance, on entering a room where the dead man's widow had just put down the telephone receiver, and saying "Tell me about the telephone call", I received the response "You cannot see a telephone here".

This sort of thing can be quite amusing at times (if you really want to get into a maze, try opening the bottle of LOBLO in Ms Dunbar's bathroom cabinet, then try to do anything else with it), but most often it is simply annoying.

When you have assembled sufficient evidence and decided on a suspect, you arrest the person and send him or her for trial. The result is given as a trial verdict. If the evidence supplied is insufficient the person will not be convicted.

This does add an aspect of realism to the game since it is not sufficient simply

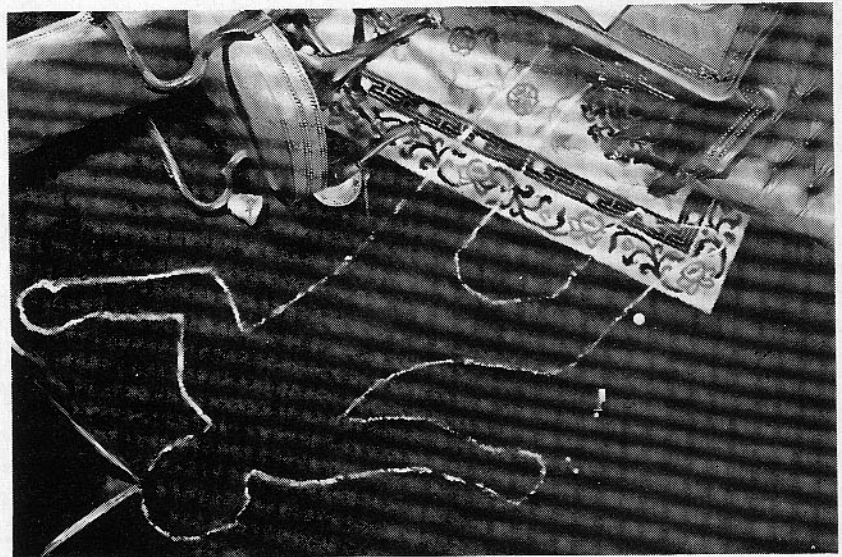
to find the murderer without obtaining evidence.

There is a facility for saving and retrieving incomplete game positions but unfortunately this appeared unreliable. I tried the game on four Apple systems and had reliable results on only one. On one system I could not save or retrieve positions at all.

However, despite these reservations, I was hooked by Deadline. It is intellectually very challenging, and the droll humour of some responses to impossible requests is quite funny.

If you fancy yourself as Sherlock Holmes rather than a dragon-slayer or a spaceship captain this game should keep you amused for some time. The program, however, really needs more debugging and rather more guidance to the allowed vocabulary for questioning witnesses.

**Keith Williamson**



# Musicomp

# Electric Duet

AS we have seen recently in Windfall, it isn't very difficult to get the Apple to play music. However, if you don't want to spend time entering the various routines, then maybe Musicomp or Electric Duet would suit you. Musicomp comes from Apple Special Delivery Software and costs about £23; Electric Duet comes from Insoft and costs about £20.

Any similarities between these two packages almost certainly arise out of the fact that both were written by Paul Lutus. A quote from the Electric Duet manual will explain the relationship between the two and the author's motivation: "The Electric Duet is the result of a three year algorithm development task, begun in 1978 with Musicomp, a program presently being sold by Apple Inc. Musicomp plays with one voice.

"Since that time a number of music synthesisers, each requiring extra circuit cards and costing several hundred dollars, have appeared in the marketplace. It has been the author's view that a computer owner shouldn't have to buy expensive add-ons in order to use a program. Therefore it was decided that a two-voice synthesiser should use only Apple II features."

Both programs have two basic sections, one to play the various musical selections included on the disc and one to allow entry of new music. The selections on Electric Duet benefit from the extra voice, sounding much better to my ears than those on Musicomp.

On the other hand, the Musicomp selections have the added feature that as the notes are played they are displayed visually on a staff. It's nice to see the tune unfolding in front of you, rippling up and down as the music does.

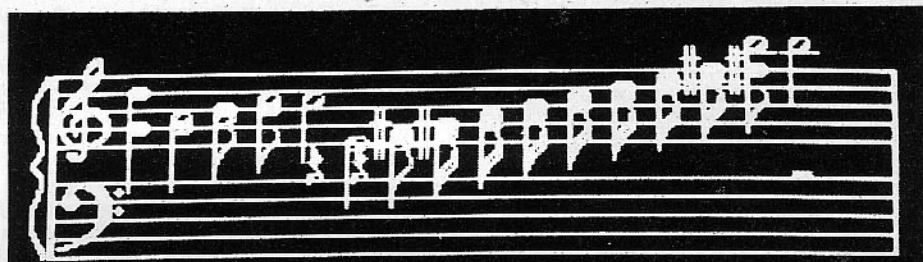
The Musicomp selections can only be chosen one at a time, whereas the Electric Duet has a "play all" facility which means you can just leave it playing in the background. Both packages contain mainly classical selections, presumably because of royalty problems with modern pieces.

The Electric Duet manual makes the point that the quality of the sound being produced is better than the Apple speaker can reproduce. You can therefore use the tape sockets to feed the signal to your hifi. I didn't try this because I have the two systems on opposite sides of the room to reduce interference and I didn't have a long enough cable!

Apparently when the signal is amplified it is sometimes possible to hear the 14080 Hz carrier tone used to simulate voltage increments. However, with the treble turned down or the scratch filter on this shouldn't be too troublesome. If you are using a direct line to a better speaker and the carrier is noticeable, a 1 to 5 microfarad capacitor across the speaker terminals might help to filter it out.

Entering music tends to be a time-consuming affair since notes are entered

## Make beautiful music - and just watch those minims mince along



Press the 'esc' key for the menu.

SELECTION: SWAN

one at a time (or in Electric Duet one pair at a time). However, the Electric Duet editor is far more powerful and I also found it easier to learn to use. I tried entering the first few bars of Paul Simon's "America" and it was a lot easier using Electric Duet.

Because the length of a musical note has no absolute value, entering music note-for-note from the sheet music results in tunes where the relative values are preserved but the absolute values sound wrong - typically, the machine whole note is longer than the whole note on the record.

The Electric Duet editor allows you to transpose the whole (or part) of your tune, not only to a different pitch but also by, for example, reducing all the note durations by the same proportion. Hence, the relative durations are preserved but the tune is effectively speeded up, or slowed down, if required.

Reading the two manuals, I got a bit confused over the use of the word "voice", since it appears to be used in two senses. In the sense that Electric Duet plays two notes simultaneously, it is a two-voice synthesizer. However both packages use the term voice to refer to the sound quality of the note played. In this sense both offer six voices. Neither offers a volume control.

Electric Duet also has two more bonus points. Firstly, one of the options allows the Apple keyboard to be used as a divided manual piano keyboard. You

couldn't rattle off a virtuoso performance on this, but it is a useful alternative method of entering music. Only the notes are entered this way, and you have to adjust the durations later. It is also a fun toy in this mode.

The other advantage is that Electric Duet can be used to put tunes or noises into your own programs with relative ease. This means that you can add the dimension of sound to all your own games and don't even have to pay a royalty to Insoft - all they ask (not unreasonably, I feel) is that your program acknowledges that Electric Duet was used to produce the sounds.

Electric Duet comes on a double-sided disc where the reverse side is a back-up. I don't find this a very sensible system, although it is not uncommon. I'm sure a corrupting magnetic field would affect both sides, as would most accidental mistreatments. I prefer the Apple Special Delivery system where a separate back-up disc is provided. Both packages require DOS 3.3.

Overall, I found Electric Duet easier to use and better sounding, although I really like the visual display used in Musicomp. Despite the fact that Musicomp is an interesting package, it is clearly the "mark 1" version of Electric Duet, and I wonder why Paul Lutus moved to Insoft.

I'm off to try Bach's Italian Concerto now . . . watch out, George Malcolm!

**Cliff McKnight**

## QUICK SPINS

**Transylvania.** Your task: to rescue a captive princess. Is she in the forest, the castle, or somewhere else? Watch out for the vampire and the werewolf. Can you put all the clues together in time? (Penguin Software.)

**Warp Destroyer.** It is the year 3526 and your enemies the Zalbians have broken a long-standing neutrality pact and seized 12 of your colonised planets. You must destroy the enemy bases to recapture the planets. (Piccadilly Software.)

**Cyclotron.** The Cyclotron provides life-sustaining energy to the surrounding empire. The evil Mangan forces launch terror weapons to try to destroy it. Your task is to manipulate repair pods and to destroy the Mangan devices before they damage the Cyclotron irreparably. How long can you hold out? (Sensible Software.)

**The Blade of Blackpoole.** Wisdom, cunning and bravery are the requisites for the successful search for the magical sword Myraglym which has been stolen from an altar and must be found and returned. The sword is said to lie in a secret chamber protected by evil serpents and horrid plants. Playing the game involves solving many puzzles to reach a successful conclusion. (Sirius Software.)

**Spitfire Simulator.** You'll have to learn to fly, and once the lessons are over you'll be thrust back to 1941 and into the midst of the Battle of Britain. You're the pilot of a Mark I Spitfire, your mission is to rid the British skies of German aircraft (there are eight different types of target aircraft) and you are limited by your own skills and experience and your fuel capacity. (Mind Systems.)

**Oil Rig.** As an oil tycoon your strategy is to gain enough money through the oil products markets to be able to afford the costs of prospecting for oil and developing oil fields. You also need enough cash to carry you through should you not hit oil. Market conditions fluctuate continually and you are threatened by periodic disasters. (CPU Software.)

**Pot o' Gold Plus.** Forty-six games with lo-res graphics and text that are simple, entertaining, and good for learning programming techniques. They include hangman, fox and hounds and pinball as well as arcade and action games such as saucer war, depth charge, shooting stars, shooting gallery and bowling. (Rainbow Computing.)

**Succession.** In this maze game you, the Masher, have to capture small numbered creatures in the correct sequence while avoiding the Chaser, a large creature with a big mouth out to swallow you. (Piccadilly Software.)

● Fuller reviews of some of these games will appear in future issues of Windfall.

# Play Time Zone and have fun reinventing critical path

"IT takes a year to play", the adverts tell us. Well we had Time Zone for about eight weeks before writing this review and have succeeded in saving the world. However we've played for a few hours practically every night and it has been a hard fight, powered by much gin. If you played it at a reasonable pace it probably would take you a year. Now we've done it there seems to be an empty space in the evening. Still, that's how it is when you save the world from destruction.

Briefly, the story is that in the year 4082 the inhabitants of the planet Neburon, under the evil ruler Ramadu, are about to blast Earth out of existence. In 1982 you are wandering along when you discover a time machine. Your task is to use it to save the Earth. However you can't just go to Neburon and punch Ramadu on the nose. That would be too easy.

There are two prehistoric time zones and then five historical ones: 50 BC, 1000 AD, 1400 AD, 1700 AD, and 2082 AD. For each of the five zones there is a choice of seven locations: Europe, Africa, Australia, Antarctica, Asia, North America and South America. Thus there are 37 locations, each of which is potentially a mini-adventure game to be solved before you can brave the perils of Neburon.

Because there is no random element it is difficult to write a review without giving clues. We'll try, though. Certainly the manual gives very little away. It contains a useful discussion of adventure games and gives some examples of commands, though pointing out that the vocabulary may change from place to place. It also recommends that you play some other adventure games before Time Zone. This is good advice, and we found it useful to have completed On-Line's "Wizard and Princess" shortly before, because they seem to have a "house style".

So off you go, exploring time and space and meeting all sorts of characters like Julius Caesar, Robin Hood, Ben Franklin, Peter the Great, Columbus, et al. There is no obvious order to go through the locations or times. The only constraint is that you can't take things back in time which would be anachronistic. But you can return with knowledge. As for playing the

game, a grasp of critical path analysis is useful, but don't worry if you haven't studied it recently - by the end of Time Zone you will have reinvented it.

Each frame consists of a hi-res picture and four lines of text. Where the message is longer than four lines, a beep sounds as a prompt to press return for the rest of the message. Not all the required information is necessarily given out automatically, so sometimes you need to interrogate the program. Time Zone includes two types of mazes, the hallway which is fairly straightforward as mazes go, and the trickier identical frame variety.

Not all the things you need to collect are immediately visible, so keep your eyes peeled.

One cunning facet of the game is that some places may only be accessible, and indeed visible, if you have solved an earlier puzzle. This means that you can think you have exhausted a zone when in fact it still has something to offer. Also success sometimes depends on not having certain things in your possession.

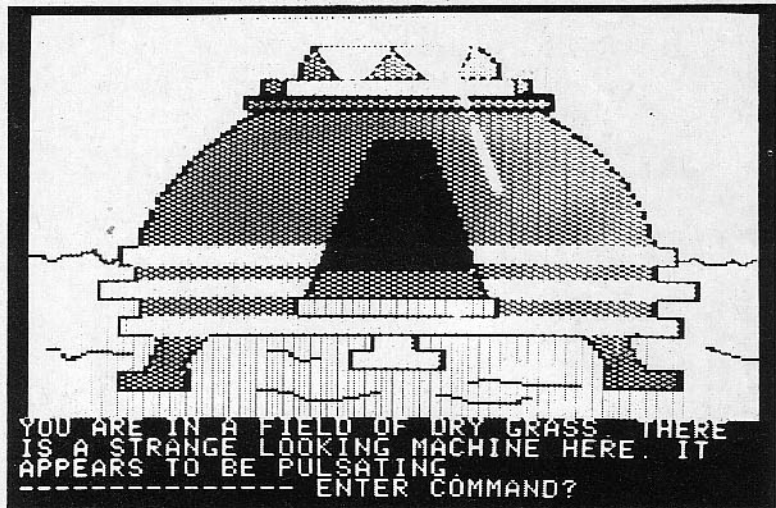
The view of the future offered is interesting. For example, gourmets may be pleased to note that Japanese food is incredibly cheap, especially in comparison to the price of rice in 50 BC. However environmentalists may be dismayed by the pollution in the Thames in 2082 after all their recent efforts. Cars in the future no longer pump lead into the atmosphere because they are solar-powered. One paradox is that street crime is still quite high, yet you can leave your possessions around and nobody seems to touch them. Everybody on Neburon speaks English and they don't seem at all alarmed by someone carrying an amazing assortment of paraphernalia.

The graphics are really good, fine in monochrome and excellent in colour, but very little use is made of sound. Despite the fact that there are a million and one ways to die in Time Zone (and we tried them all), there is quite a lot of wit and humour, and we frequently found ourselves laughing at the comments made out of the actions required.

There is so much to do that we developed a division of labour whereby one person manipulated the keyboard and

one

# analysis



The time machine that takes you through time and space

the other acted as navigator and map-maker. It also helped to have someone to bounce ideas off because some of the adventures require a fair bit of ingenuity, and it is often very easy to miss a clue. Be warned, too, that in at least one place you need to speak American as opposed to English. That, and the fact that you will need to use almost everything you acquire, are the only clues we'll give.

Having been positive about Time Zone, let's now look at reservations. Firstly, the price is quite high. Time Zone retails for around £50 and although it comes on six double-sided discs (not eight, as some adverts claim) On-Line recommend that you make back-ups of 11 of the sides and use these, and you also need a save game disc. If, like us, you use single-sided discs, this means that you have another £18 or so to pay.

Incidentally, the master disc provides a back-up facility. However, on three occasions the back-up failed to operate properly although we'd had a "verifica-

tion good" message. Twice it was possible to re-copy successfully, but the last time we had to resort to using Locksmith because the apparently well copied back-up refused to operate no matter what.

At times we got fed up with the need to type the same phrases repeatedly and got to the point where, for example, we would "pull lever" because the three consecutive l's were quicker to type than "push lever". Also at times the required commands seemed unnecessarily pedantic.

Given the size and complexity of Time Zone, it is inevitable that some errors creep in. Neglecting the various spelling errors, a couple of the frames hadn't had all the colours filled in, and at one point the wrong frame appeared. The picture was totally out of context and was obviously from another part of that time zone, although the text was appropriate.

With 11 sides of disc (side 1A is a master disc), a lot of time is spent putting in and taking out discs. This is exacer-

bated by the fact that movement in and out of the time machine usually involves a disc change, as does saving the game. This latter becomes a habit after you have been killed a few hundred times! Have a ream of paper at the ready, too, because you'll need it for all the maps you need to make.

On-Line suggest that if all else fails and frustration exceeds threshold level, you can telephone them for a hint. If you get to this point it might be cheaper to ring *Windfall* and get our number. We would be happy to give a hint, preferably between 10.30am and 10.30pm. It should be cheaper than phoning California – and no time difference is involved.

Although on the face of it Time Zone is expensive, it is only about three times the price of the average adventure game and it is certainly much bigger and more involved than three such games.

**Cliff and Denise McKnight**

## Now make your Apple really sing!

WHEN it comes to games and music the Apple's built-in speaker leaves a lot to be desired. And connecting the audio output directly to a hi-fi amplifier can have disastrous results.

Now *Windfall* has the answer. Designed to our own strict specification, the *Windfall Hi-Fi Adapter* fits neatly inside the Apple with a toggle switch just protruding through a cooling fin on the left hand side.

With the switch in the up position the Apple's own speaker is activated. Switch down and the sounds of missiles, collisions, explosions – and their musical accompaniment – come through your hi-fi with a startling realism you would not have believed possible.

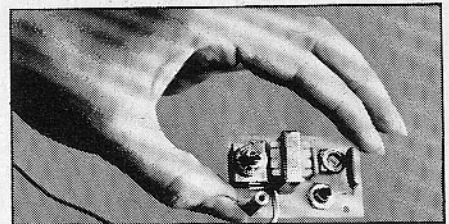
Fitting is extremely simple and no technical knowledge is required.

The *Windfall Hi-Fi Adapter* will go on sale in 1983 for £25, but as a special pre-Christmas introductory offer *Windfall* readers can obtain one

for £19.95 (including VAT) plus £1 post and packing.

This special offer expires on December 31, 1982.

You can now add a completely new dimension to the capabilities of your Apple. But to make sure you get one in time for Christmas please order without delay.



Please supply ..... *Windfall Hi-Fi Adapter* at the introductory price of £19.95, plus £1 post & packing.  I enclose cheque  Paid by credit card

Name ..... Credit card name

Address ..... Number

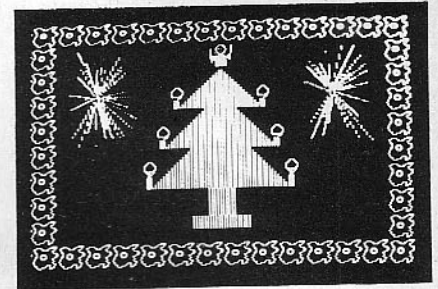
..... Expiry date

..... Signed .....

Price for orders received after 31/12/82 £25, plus £1 p&p.

Send to: *Windfall*, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY. No stamp needed if posted in UK.

# Bit Stik competition comes up with a real Christmas cracker



THE staff of *Windfall* have long regretted that Britain has seemed unable to produce home-grown games software writers, and that Apple users have always had to rely on American imports to keep them entertained.

Yet the ability is certainly here. It only needs a little coaxing to bring it out.

This has been shown by the interest aroused by the *Windfall* contest in which we asked readers to design a Christmas card making the fullest use of the Apple's graphics, colour and sound capabilities.

The winner is Mr J. Taylor, of 14 Woodway Close, Teignmouth, Devon. His prize is the very latest version of the

revolutionary Bit Stik, the creative design system that has been receiving rave reviews in the computing press.

Although it arrived after the closing date for entries, we have been most impressed with a disc submitted by 12-year-old Clifford M. Davies, of Doncaster. Startech are sending him a copy of the DIY games creator, The Arcade Machine, and we hope we will see more of his talent in the future.

Below is the listing of Mr Taylor's winning entry.

```

10 REM      SUBMITTED BY
      J.TAYLOR
      14 WOODWAY CLOSE,
      TEIGNMOUTH,DEVON.

20 HGR : TEXT : HOME
30 VTAB 10: HTAB 16: PRINT "BON
      NOEL!"
40 HTAB 15: PRINT "*****"
50 DATA ,32,74,255,160,128,152,3
      2,21,81,73,255,32,21,81,136,
      208,244,32,63,255,96,170,202
      ,208,253,44,48,192,96
60 A = 20736: FOR J = 1 TO 29: READ
      S: POKE A,S:A = A + 1: NEXT

70 DATA 169,0,132,0,169,32,133,
      1,177,0,73,255,145,0,200,208
      ,247,230,1,165,1,201,64,208,
      239,96
80 A = 20480: FOR I = 1 TO 26: READ
      J: POKE A,J:A = A + 1: NEXT

90 DATA 4,0,10,0,109,0,130,0,158
      ,0,60,63,44,45,40,100,228,60
      ,60,63,23,30,54,118,150,54,5
      4,54,54,39,36,36,4,32,63,63,
      36,36,36,77,73,13,77,73,49,5
      4,30,54,54,62,32
100 DATA 36,30,63,54,54,54,62,63
      ,39,36,36,36,45,54,54,54,39,
      36,36,13,54,54,182,255,77,9,
      36,36,36,40,4,88,88,88,88,88
      ,64,9,36,223,219,219,146
      ,146,146,50,63,39,45
110 DATA 22,55,54,0,0,162,2,171,
      36,36,36,36,63,55,54,54,54,4
      6,37,39,37,39,37,39,37,39,44
      ,6,0,24,56,60,60,60,36,36,37
      ,37,37,45,46,46,46,54,62,62,
      62,62,46,44
120 DATA 44,63,63,60,45,45,0,37,
      60,60,55,55,46,46,173,146,24
      2,63,23,63,7,32,63,32,4,32,1
      2,44,36,60,36,56,102,1,40,45
      ,21,53,45,37,45,40,37,45,53,
      55,30,55,14,45
130 DATA 54,62,23,55,22,14,53,30
      ,62,31,36,63,149,10,37,88,88
      ,64,1,32,64,9,32,223,219,219
      ,219,219,219,147,146,50,0,0,
      2,50
140 A = 24576: FOR I = 1 TO 237: READ
      J: POKE A,J:A = A + 1: NEXT

150 GOSUB 930
160 AHI = INT (24576 / 256):ALD =
      24576 - 256 * AHI
170 POKE 232,ALD: POKE 233,AHI
180 SCALE= 1: ROT= 0
190 HCOLOR= 1
200 GOSUB 810
210 FOR I = 15 TO 150 STEP 20
220 XDRAW 2 AT 90,130: XDRAW 2 AT
      192,130
230 XDRAW 2 AT 110,70: XDRAW 2 AT
      172,70
240 XDRAW 2 AT 100,100: XDRAW 2 AT
      182,100
250 XDRAW 3 AT 90,122: XDRAW 3 AT
      192,122
260 XDRAW 3 AT 100,96: XDRAW 3 AT
      182,96
270 XDRAW 3 AT 110,66: XDRAW 3 AT
      172,66
280 X = 90:X1 = 190:Y = 130:Y1 =
      100
290 FOR DRW = Y TO (Y1 - 20) STEP
      - 1
300 HPLLOT X,DRW TO X1,DRW
310 X = X + 1:X1 = X1 - 1
320 NEXT
330 F = F + 1
340 IF F = 2 THEN 380
350 IF F = 3 THEN 400
360 X = 100:X1 = 180:Y = 100:Y1 =
      80
370 GOTO 290
380 X = 110:X1 = 170:Y = 70:Y1 =
      60
390 GOTO 290
400 REM
410 HCOLOR= 5
420 FOR Y = 131 TO 150
430 HPLLOT 130,Y TO 150,Y: NEXT
440 HCOLOR= 2
450 FOR Y = 151 TO 160
460 HPLLOT 120,Y TO 160,Y: NEXT
470 POKE - 16304,0: POKE - 163
      02,0: POKE - 16297,0
480 POKE 20746,10
490 D = .01745:R = 20
500 FOR A = 0 TO 340 STEP 60
510 X = SIN (D * A) * R
520 Y = COS (D * A) * R
530 XDRAW 1 AT 50 + X,60 - Y
540 XDRAW 1 AT 230 + X,60 - Y
550 XDRAW 3 AT 90,122: XDRAW 3 AT
      182,96
560 XDRAW 3 AT 100,96: XDRAW 3 AT
      172,66
570 XDRAW 3 AT 110,66: XDRAW 3 AT
      192,122
580 XDRAW 3 AT 110,66: XDRAW 3 AT
      192,122
590 XDRAW 3 AT 100,96: XDRAW 3 AT
      182,96
600 XDRAW 3 AT 90,122: XDRAW 3 AT
      172,66
610 CALL 20736
620 NEXT A
630 IF FL = 2 THEN 660
640 FOR I = 1 TO 1000: NEXT
650 FL = FL + 1: IF FL = 1 THEN F
      L = 2: GOTO 500
660 HCOLOR= 3
670 XDRAW 1 AT 142,30
680 POKE 20746,200: CALL 20736: POKE
      20746,90: CALL 20736: CD = CD
      + 1: IF CD < 3 THEN 680

690 FOR I = 1 TO 500: NEXT
700 POKE 20746,255
710 O = 50:P = 60:S = 0
720 CALL 20736
730 FOR A = 1 TO 50
740 HCOLOR= RND (1) * 7 + 1
750 HPLLOT O,P TO 50 * RND (1) +
      (O - 20),78 * RND (1) + (P -
      38)
760 NEXT
770 CALL 20736
780 S = S + 1
790 IF S = 2 THEN 890
800 O = 220: GOTO 730
810 REM HOLLY
820 FOR I = 10 TO 278 STEP 20
830 XDRAW 4 AT I,9
840 XDRAW 4 AT I,180: NEXT
850 FOR I = 28 TO 170 STEP 19
860 XDRAW 4 AT 9,I: XDRAW 4 AT 2
      70,I
870 NEXT
880 RETURN
890 REM INVERT SCREEN AND TUNE
      HERE
900 FOR I = 1 TO 2000: NEXT : CALL
      20480
910 FOR I = 1 TO 2000: NEXT : CALL
      20480
920 GOTO 960
930 FOR L = 776 TO 798: READ B: POKE
      L,B: NEXT
940 DATA 255,255,173,48,192,136,
      208,5,206,9,3,240,9,202,208,
      245,174,8,3,76,10,3,96
950 RETURN
960 GOTO 1060
970 READ P
980 IF P = 999 THEN RETURN
990 READ D
1000 POKE 776,P: POKE 777,D: CALL
      778
1010 GOTO 970
1020 DATA 86,108,64,144,64,72,6
      4,108,51,108,57,162,64,72,57
      ,108,999
1030 DATA 51,108,64,162,64,10
      8,51,162,43,72,38,255,999
1040 DATA 38,108,43,162,51,72,5
      1,72,64,108,57,162,64,108,57
      ,162,999
1050 DATA 51,108,64,216,76,72,7
      6,108,86,108,64,108,999
1060 GOSUB 970
1070 FOR I = 1 TO 50: NEXT : REM
      PAUSE
1080 GOSUB 970
1090 FOR I = 1 TO 50: NEXT
1100 GOSUB 970
1110 FOR I = 1 TO 50: NEXT
1120 GOSUB 970
1130 FOR I = 1 TO 1000: NEXT I
1140 HOME : TEXT : VTAB 10: HTAB
      12: PRINT "HAPPY NEW YEAR!":
      END

```



# Review reprise

## Wizardry

Highly recommended fantasy adventure. If you don't get it you are missing a lot. It is not a simple game, but it is one that gives a player a personalised approach to his characters that has been so sadly lacking in other adventure efforts.— *January*

## Olympic Decathlon.

Up to six players can compete in 10 Olympic events. Excellent graphics, highly recommended.— *February*

## Dragon's Eye

Fantasy adventure where the "confrontations" offer hi-res graphics fights, rather than the usual verbal exchanges.— *February*

## Crush, Crumble and Chomp

An adventure game where you play the monster (and can design your own monster to play with). Good presentation graphics and sound effects.— *March*

## Castle Wolfenstein

Well thought-out war adventure where you have to escape from Second World War SS prison. Good graphics, variable scenario.— *April*

## President Elect

Simulation of an American presidential election campaign. Fairly heavy going, but fun if you have time and are politically inclined.— *April*

## Threshold

A spaceship defence arcade game which has a nasty habit of keeping you out of bed. Gives you a sporting chance to at least get started if you were at the end of the queue when lightning reactions were handed out.— *April*

## Flight Simulator

Must be regarded as something much more than "just a game". You are the pilot of an aircraft and the experiences you are put through are rewarding, highly demanding and stimulating.— *May*

## Looking back at the games Windfall reviewed in '82

## Night Mission Pinball

Nice but noisy arcade game. Allows you to "tamper" with the machine to alter the challenge and capabilities.— *June*

## The Prisoner

Based on the TV series of the same name, this is a radical departure from the usual adventure game genre or shoot 'em all space odyssey. Complex, riveting and makes you think.— *June*

## Pursuit of the Graf Spee

Second World War battle, which is a boon to armchair strategists. Enjoyable and generates tension.— *July*

## Flywars

Arcade adventure for the fleet of finger, which will stop you in your tracks. As a spider you have to spin a web to trap and devour flyfighters and the occasional caterpillar. The struggle is by no means one sided.— *August*

## Bandits

An arcade game that is indebted to Space Invaders for the original concept. Great fun and with 28 levels of play should satisfy everyone.— *August*

## Suicide

Absorbing arcade game where you have to try and stop silly creatures taking the plunge off tall buildings. It looks comically simple but your "safety net" moves so fast that it is difficult to position it under the would-be suicides.— *August*

## Swashbuckler

Hi-res graphics swordfighting game. Simple scenario, credible duels, excellent challenge. At first the pirates come at you one at a time. But that is too good to last,

so take two, with the odd killer rate, scorpion or snake thrown in for good measure.— *August*

## Choplifter

An SAS-style rescue game. It is tactical rather than high speed, has spectacular graphics and is addictive.— *September*

## Odyssey

A very good fantasy adventure game with a liberal sprinkling of randomness.— *September*

## Wordscore

Wordgame with listing, written by Geoff Buckle. Ten characters are chosen at random and the player must assemble words of three characters or more. Gives plenty of practice in playing around with strings.— *September*

## Knight of Diamonds

An extensive fantasy adventure which follows on from, but is not as good as its predecessor, Wizardry.— *October*

## Pig Pen

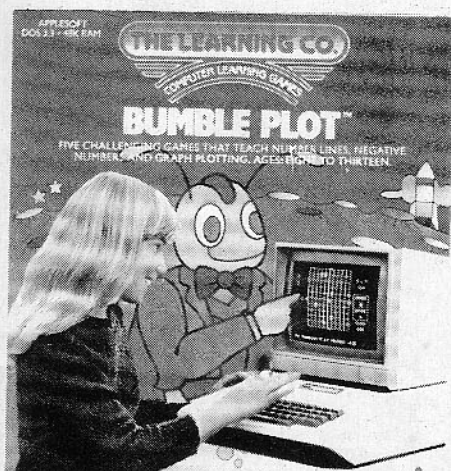
Reminiscent of the famous Pacman but with greatly improved graphics. Not particularly impressive but will appeal to those who like maze chases.— *October*

## Galactic Wars

A cross between battleships and space invaders and a refreshing mixture of brain-taxing strategy and wrist wrenching battle.— *November*

## Raster Blaster/David's Midnight Magic/Night Mission Pinball

Three pinball games all enjoyable and all providing good simulation and control.— *November*



## Games with a message

A COMPLETELY new approach to the presentation of computer games for younger children has been developed by an American educational software house, The Learning Co.

Their first discs, which they say are suitable for children from the age of three upwards, have arrived in Britain in time for Christmas and are marketed by Pete & Pam.

In a breakaway from the conventional software package, each disc comes in a 6in square hardback book with a colourful cover containing a profusely illustrated manual.

The Learning Co. say until now no one has offered playful computer software that takes a serious approach to interac-

tive learning for children.

And they add: "Our games let children learn to enjoy the learning process, enlivening their imaginations, expanding their creativity and sparking their thrill of discovery."

There are spelling and word unscrambling games, creative games that teach shape and colour relationships, and a series that tempts them to explore the concepts of left and right, up and down.

One disc, called Logic Gators, is ambitiously described as a fast-moving game that "lets children use modern computer circuit logic to construct increasingly complex machines".

A review of the series will appear in a future issue of Windfall.



# SCRAM

It takes skill to make  
the melody linger on..

```

10 GOTO 1000
20 VTAB 21: PRINT "SCORE = "SC +
  XN * N
30 FOR I = 1 TO N
40 KEY = PEEK ( - 16384): POKE -
  16384,0
50 IF KEY = 201 THEN YN = YN - 1
  : GOTO 90
60 IF KEY = 202 THEN XN = XN - 1
  : IF XN < 0 THEN XN = 0: GOTO
  90
70 IF KEY = 203 THEN XN = XN + 1
  : GOTO 90
80 IF KEY = 205 THEN YN = YN + 1
  : GOTO 90
90 IF SCRN( XN,YN) = 12 THEN PRINT
  "": XN = X0:YN = Y0: GOTO 120

100 IF SCRN( XN,YN) = 9 THEN 50
  00
110 IF XN = 39 THEN 6000
120 COLOR= 0: PLOT X0,Y0:X0 = XN
  :Y0 = YN
130 COLOR= 15: PLOT XN,YN
140 IF XN > X(I) THEN NEXT : GOTO
  20
150 TEST = SCRN( X(I) - 1,Y(I))
160 IF TEST = 15 OR TEST = 0 THEN
  X(I) = X(I) - 1: GOTO 210
170 Y(I) = Y(I) + YC(I)
180 IF Y(I) < 1 THEN Y(I) = 1:YC
  (I) = - YC(I): GOTO 200
190 IF Y(I) > 38 THEN Y(I) = 38:
  YC(I) = - YC(I): GOTO 200
200 IF SCRN( X(I),Y(I)) = 15 THEN
  TEST = 15
210 COLOR= 0: PLOT X0(I),Y0(I):X
  0(I) = X(I):Y0(I) = Y(I): COLOR=
  9: PLOT X(I),Y(I)
220 POKE 6,MC(I): POKE 7,MT: CALL
  768
230 IF X(I) = 0 THEN X0(I) = 0:X
  (I) = 39
240 IF TEST = 15 THEN 5000
250 NEXT I
260 VTAB 21: PRINT "SCORE = "SC +
  XN * N
270 GOTO 20
970 :
980 REM *** INITIALISATION ***
990 :
1000 TEXT : HOME :D$ = CHR$( 4)
  :SC$ = "SCRAM HIGH SCORER"
1005 GOSUB 40090
1010 INPUT "YOUR NAME PLEASE ";N
  AME$(0)
1020 PRINT D$:"OPEN"SC$
1030 PRINT D$:"READ"SC$
1040 INPUT NAME$(1),HS
1050 PRINT D$:"CLOSE"SC$
1060 VTAB 10: HTAB 5
1070 INPUT "DO YOU WANT INSTRUCT
  IONS ?";AN$
1080 IF LEFT$( AN$,1) = "Y" THEN
  GOSUB 2000
1085 NU = 10:TIME = 3
1090 POKE 34,20
1140 DIM X(44),Y(44),X0(44),Y0(4
  4),YC(44),MC(44)
1170 :
1180 REM *** DRAW WALLS ***
1190 :
1200 FOR I = 1 TO 30
1210 X(I) = INT ( RND ( 1) * 38) +

```

HERE'S a game which measures achievement by the speed of a song, with the ultimate goal being to have that song played at the correct tempo.

Scram, written by **Michael Hambly**, uses low resolution graphics and sound. Vertical walls are drawn on the screen, each with two holes or passages in it, and the aim is to manipulate "your man" to the right hand side of the screen, avoiding destroyer "blobs."

First time across and you are rewarded with a few notes of Greensleeves (played at a very fast speed). There are 10 levels, and at the end of each you are rewarded with a few more bars of Greensleeves, played nearer the correct speed.

The game incorporates a facility for recording "highest score", but when running the game for the first time the line 1015 GOTO 1070 should be inserted.

```

2: X0(I) = X(I):Y0(I) = 16 + INT
  ( RND ( 1) * 6):Y0(I) = Y(I):
  YC(I) = - 1 ^ ( INT ( RND (
  1) * 2)) * ( INT ( RND ( 1) *
  3 + 1))
1220 MC(I) = INT ( RND ( 1) * 100
  + 50)
1230 NEXT
1240 BR :N = 8:MT = 15:XN = 0:YN
  = 19
1250 COLOR= 12
1260 FOR I = 0 TO 38 STEP 2
1270 VLIN 0,39 AT I
1280 NEXT I
1290 HLIN 0,39 AT 0: HLIN 0,39 AT
  39
1300 COLOR= 0
1310 VLIN 16,23 AT 0
1320 VLIN 16,23 AT 38
1330 FOR I = 2 TO 36 STEP 2
1340 PLOT I, INT ( RND ( 1) * 38)
  + 1: PLOT I, INT ( RND ( 1) *
  38) + 1
1350 NEXT
1360 COLOR= 15
1370 GOTO 20
1970 :
1980 REM *** INSTRUCTIONS *** ;
1990 :
2000 HOME
2010 PRINT "YOU WILL BE PLACED O
  N THE LEFT OF A NUMBER O
  F WALLS, AND WILL HAVE TO MA
  KE IT TO THE OTHER SIDE."
2020 PRINT "EACH WALL WILL HAVE
  TWO HOLES IN IT BUT FEROCIOU
  S FIENDS WILL BE AFTER YOU !
  !"

```

```

2030 PRINT "TO MAKE YOUR MOVE PR
  ESS THE FOLLOWING:- "
2040 PRINT " I - .UP"
2050 PRINT "- LEFT - J K - RIGH
  T"
2060 PRINT " M - DOWN
  "
2070 PRINT
2080 PRINT "ONLY ONE PROBLEM ...
  ....."
2090 PRINT
2100 PRINT "WHEN YOU REACH THE O
  THER SIDE YOU WILL BE PUT B
  ACK TO THE START AGAIN WITH
  NEW WALLS AND MORE FIENDISH
  ,FEROCIOUS BLOBS"
2110 PRINT
2120 PRINT "IF YOU GET ACROSS 10
  TIMES YOU ARE FREE"
2130 NU = 73:TIME = 21: GOSUB 400
  00
2140 PRINT : PRINT
2150 INPUT "PRESS RETURN TO BEGI
  N ";AN$
2160 HOME
2170 RETURN
4970 :
4980 REM *** YOU LOSE ***
4990 :
5000 FLASH : PRINT "GOT YU !!": NORMAL
5010 FOR I = 90 TO - 90 STEP -
  2: POKE 6,30 + ABS ( I): POKE
  7,3: CALL 768: NEXT
5020 SC = SC + XN * N
5030 IF SC > HS THEN HS = SC:NAM
  E$(1) = NAME$(0)
5040 HOME : POKE 33,40: HTAB 26
5050 PRINT "HIGH SCORE ";HS
5060 HTAB 26: PRINT "BY ";NAME$(
  1)
5070 POKE 33,22
5080 PRINT "LATEST SCORE "SC
5090 SC = 0:PAUSE = 4000: HOME
  5095 NU = 10:TIME = 3
5100 GOSUB 7000
5110 GOTO 1200
5970 :
5980 REM *** REACHED RIGHT OF
  SCREEN - RETURN TO LEFT ***
5990 :
6000 SC = SC + XN * N
6010 N = N + 4: IF MT > 4 THEN MT
  = MT - 3
6020 XN = 0:WIN = WIN + 1
6030 IF WIN < 10 THEN NU = NU +
  9:TIME = TIME + 2: GOSUB 400
  00: GOTO 1250
6070 :
6080 REM *** ACROSS 10 TIMES, T
  HEREFOR YOU WIN ***
6090 :
6100 NU = 73: GOSUB 40000: REM
  ** VICTORY TUNE **
6110 TEXT : HOME : SPEED= 100
6120 VTAB 10: PRINT "CONGRATULAT
  IONS YOU MADE IT ACROSS 10 T
  IMES - NOW YOU CAN RETIRE !"
6130 SPEED= 255: GOSUB 40000
6140 GR : VTAB 21
6150 GOSUB 7000
6970 :
6980 REM *** NEW GAME OR SAVE
  HIGHEST SCORE ***

```

and, finally . . .

## If you don't play you can't win..

ALTHOUGH it is produced by a major hardware house, Living, by Evolutionary Hardware Ltd, is actually a mixture of hardware, software and firmware. All the necessary hardware is provided and some of the "wired in" routines operate to update the software as the game proceeds.

The Time Zone boast that "it takes a year to play" pales in comparison with this game. It is generally considered that several years of apprenticeship to one or two experienced players are necessary before a player can start in his own right.

One of the major problems stems from the fact that the manufacturer refuses to publish a manual. Consequently several unofficial ones have been circulated, each claiming to be the official one. That so many manuals can survive is a testimony to the game's flexibility, since each manual offers a different scenario, a different command set and in many cases a different definition of a successful game. Some players claim to manage without a manual and others try out several, but most people find one they like and stick with it.

When the system is first booted it is not at all clear what is happening. It is at this time that an experienced player is vital to guide and prompt the new player, who gradually takes control as Living develops into a more-or-less standard adventure game. Indeed, it bears many similarities to Odyssey and The Prisoner. Like The Prisoner, most of the situations in which a player finds himself are ambiguous and have a second level of meaning which is not always easy to access. As they say in The Prisoner, the game is rigged, but if you don't play you can't win!

Gradually the player learns a command set and begins to explore, gaining new experience and meeting new challenges, although most of this phase is played out in a single scenario. Briefly, this involves about 30 players being transported to an inquisition room where it is necessary to provide the resident inquisitor with answers to his questions. Hints are given so the new player is not completely lost, and as the number of correct answers increases so the difficulty level is incremented.

After several iterations in this scenario, many players drop out. However, the resolute continue since success means that more movement is possible in the second scenario. Success at this stage seems to be defined largely in terms of proportion of right answers given, although this is not the only possible exit route. As in Wizardry, it is important to

build a strong character before movement to the second scenario, and number of correct answers is just one (some say the easiest) way to achieve this.

Movement into the second scenario usually signals the end of the apprenticeship and at this stage many people join with another player in readiness to train their own apprentice. This makes the game stable over several plays, since players tend to coach their apprentices in the same way in which they were coached. However, Evolutionary are to be congratulated because their self-replicating hardware/software package has such a powerful random element built in that trainer performance is not a very good predictor of apprentice performance.

It is success in this scenario which is difficult to define. Some players insist on adopting an Odyssey-type strategy, aiming for a high score (by amassing as many possessions as possible). Others view it more in terms of The Prisoner, constantly trying to escape and being suspicious of everyone. Ultimately, it is up to each player to define winning, although the unofficial manuals all offer their own thoughts on the topic.

So, how does Living rate as an adventure game? Some hardware problems are occasionally reported in the literature, but I will confine my remarks to the normal package because, despite the problems, nobody has yet instigated successful litigation against the manufacturer. No warranty is given, and no correspondence is replied to, so complaint is useless.

It has to be said that the graphics are superb, and the self-replication routine has been known to occupy many players to the point of obsession. However, overall the game lacks many of the standard adventure game facilities. For example, there is no facility for suspending play and returning at a later time. Movement between scenarios is erratic, and lack of proper documentation is disgraceful in this day and age.

On the positive side, the discovery of the various magic potions can considerably enhance play in a manner similar to the Electric Duet editor; some of them speed play up, some slow it down and some transpose it to a different plane.

If you haven't played this game yet, don't wait too long because there is a rumour in the industry that the government intend to price it out of the market!

**Geoff O'Dee**

```
6990 :
7000 POKE - 16368,0
7010 INPUT "ANOTHER GAME (Y/N)?";AN#
* 7020 IF LEFT$(AN#,1) = "N" THEN
7070
7030 INPUT "SAME PERSON (Y/N) ";
AN#
7040 IF LEFT$(AN#,1) = "N" THEN
INPUT "NAME PLEASE ";NAME*(
0)
7050 HOME
7060 RETURN
7070 PRINT D$;"OPEN"SC$
7080 PRINT D$;"DELETE"SC$
7090 PRINT D$;"OPEN"SC$
7100 PRINT D$;"WRITE"SC$
7110 PRINT NAME$(1)
7120 PRINT HS
7130 PRINT D$;"CLOSE"SC$
7140 TEXT
7150 END
39970 :
39980 REM *** NOTES FOR GREENSL
EEVES ***
39990 :
40000 IF N1 = 0 THEN GOSUB 4007
0
40010 FOR I = 1 TO NU
40020 POKE 6,N0(I):POKE 7,NL(I)
* TIME: CALL 768
40030 NEXT
40040 RETURN
40050 :
40060 :
40070 REM **** POKE CODES FOR M
USIC ****
40080 :
40090 FOR I = 768 TO 786
40100 READ NO:POKE I,NO
40110 NEXT
40120 DATA 173,48,192,136,20
8,4,198,7,240,8,202,208,246,
166,6,76,0,3,96
40130 :
40140 REM **** NOTES FOR MUSIC
****
40150 :
40160 READ MUSIC#,TIME,NU
40170 DIM NO(NU),NL(NU)
40180 FOR I = 1 TO NU
40190 READ NO(I),NL(I)
40200 NEXT
40210 N1 = 1
40220 RETURN
40230 DATA GREENSLEEVES,17,73
40240 DATA 140,4,116,8,104,4,92,
6,86,2,92,4,104,8,124,4,156,
6,140,2,124,4,116,8,140,4,14
0,6,148,2,140,4,124,8,148,4,
184,8,140,4,116,8,104,4,92,6
,86,2,92,4,104,8,124,4,156,6
,140,2,124,4,116,6,124,2,140
,4,148,6,164,2,156,4,140,8,1
40,4,140,12
40250 DATA 78,12,78,6,82,2,92,4,
104,8,124,4,156,6,140,2,124,
4,116,8,140,4,140,6,148,2,14
0,4,124,8,148,4,184,12,78,12
,78,6,82,2,92,4,104,8,124,4,
156,6,140,2,124,4,116,6,124,
2,140,4,148,6,164,2,156,4,14
0,12,140,12
```

# Wanted: A program to print out music

*CAN any Apple users help me find a program to write music? There have been many reviews in Windfall of Apple programs enabling music to be composed and played back through amplification systems, but nothing which enables these sounds to be written onto paper.*

*With my limited knowledge of computers it seems feasible on the face of it to press a key on the computer or keyboard, commit it to memory and have it played back as a sound. It should be possible for the instruction to apply to a means of printing music symbols as in the case of a typewriter/word processor.*

*I am seeking a method of composing a full score, transcribing the instrumental parts and have the ability to transpose. Any advice on what is available would be most welcome. — J.A. Cleary, Ashwellthorpe, Norfolk.*

● See the review of Musicomp and Electric Duet on Page 47. Next month Windfall will have news of a revolutionary system which produces the score of music performed on a keyboard synthesiser.

## Patching for Palsoft

*I RUN an ITT 2020 48k with Palsoft in ROM. Does anyone know how to use the Tasc Applesoft Compiler with this machine? I have tried to use it, but because of differences in the interpreters of Applesoft and Palsoft it will not run. I would be grateful if anyone knows how to patch it or has already patched it and is using it with Palsoft. — A. Lightowler, Pontefract.*

● The Tasc Compiler makes extensive use of the Applesoft interpreter which is located in ROM. The appropriate routines (ie multiplication etc) are accessed using an absolute address. These addresses are not necessarily the same on the ITT machines. **Peter Brameld.**

## Split program modification

*I AM interested in modifying Applesoft to support an IDS440G Paper Tiger. I noticed Mr R.D. Boylan's letter in July Windfall and your comments relating to it.*

*I wonder if you could supply me with a listing of Appleplot modified to support the IDS440G. I appreciate that Appleplot*

*is split about the Appleplot logo so I would also like to know either how to "unsplit" Appleplot or if the spacer program in February's Windfall will do the job. — John Taylor, Sheffield City Polytechnic.*

● Modifying a split program is too complex a topic to be dealt with here. The only simple way to effect such modification is via the monitor, finding the appropriate point within the program by identifying the hex token codes, e.g. the hex code for the PRINT command is BA(186 dec).

An intermediate solution is to save the Appleplot graphs as a picture and to use the commercially available Integral Data Systems hi-res dump. Make sure you get the one appropriate to your particular printer model. Some dealers give this disc free on the purchase of a printer. **Peter Brameld.**

## Awaiting promotion?

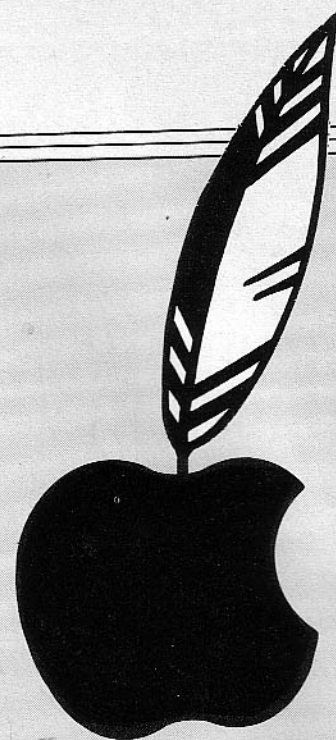
*I NOTE that you have abbreviated part of my address for my Windfall subscription to read: "Computer Stud Dep".*

*It's bad enough being a computer student ... but a DEPUTY one? — Mr Dobbs, Newtown School, Waterford, Eire.*

## CP/M solution sought

*I HAVE a problem with a CP/M operating system on an Apple II. I am using APLV80 under CP/M which requires a special APL terminal. Standard input/output on CP/M is via slot 3. The terminal interface card normally resides in slot 1.*

*To avoid swapping circuit boards each time APL is used I wish to reassign the console input/output to slot 1. This will involve patching a machine code program into the UC1: (user defined console) sec-*



*tion of the CP/M BIOS.*

*I need to know the address of UC1:, the address of slot 1, and the form of the input/output program (possibly cribbed from TTY: if I knew its address), and would be grateful if anyone can offer me a solution. — Howard Carter, High Wycombe.*

## Tiger topic

*I WOULD be very interested to hear from anyone who has managed to use a Paper Tiger 445 or similar direct from Apple Plot.*

*I have tried to add a routine, but the peculiar split in the program will not allow modification, at least not with my limited knowledge. — J. Gooding, Establishment Manager, Philips Business Systems, Communication and Control Division.*

## Practical Micro Programming

*IN the September issue of Windfall there is a very interesting article on med-res graphics by Ed Peach. It contains a reference to a book "Practical Micro-computer Programming — The 6502" by Weller. Could you please let me know the book's publisher, price, and what it contains.*

*Using this opportunity, I should like to see more articles and programs on graphics animation in your magazine and then especially in machine language. — Finn Ravndal, Blystadlia, Norway.*

● The book (author's initials W.J.) is published in the United States by Northern Technology Books and costs \$32.95 (about £20.) Basically it is a teaching primer on the assembly language for the 6502.

## Yes, you can buy that puzzle

I LIKED the novel jigsaw puzzle with the picture of the transparent Apple which made such an effective cover for the November issue of Windfall. Can you tell me where I can buy one from? – **Martin Dukes, Maidenhead.**

● Yes, from any Apple dealer. It costs £6.30. But be warned – it's not as easy as it looks!

## Pascal program outline error

I HAVE just received my copy of the November Windfall, and have an apology and a complaint about my Appletip for producing a Pascal program outline. The apology is for an error about half way through:

```
The line WRITELN(F); should
read WRITELN(G);
```

The effects of this error are not totally disastrous, but correcting the input file after running this program is fiddly and annoying.

The complaint is about the clipping of the two END statements. Although an experienced Pascal programmer will realise immediately that they are missing, a less competent programmer will probably have to suffer a couple of UNEXPECTED END OF INPUT errors in compiling before he sorts out the problem. – **Jonathan Lewis, Abingdon.**

## Disabling RESET key

I HAVE read the article by John S. Littler on Interrupting your Apple in the July and August issues of Windfall, and decided to try and completely disable the RESET key. I have tried the following code and various others but to no avail:

```
*3F2:00 03 (RETURN) CALL-1169
(RETURN)
• 300:40 (RETURN)
```

This should set up an RTI at \$300 and set the address where the RESET key will force a JMP to. The CALL-1169 is to set the power-up byte at \$3F4. Have you any

suggestions as to how this could be achieved as this does not work? – **Alec Myers, aged 11.**

● Thank you for your query about the RESET key. The difficulty you have found is that although you have forced the reset to go to a RTI instruction at 300, the RTI tries to pull off the stack the programme counter, and process status, but they aren't there! Unfortunately, unlike NMI, IRQ, and BRK, RESET does not save the current situation on the stack, and indeed probably resets the value of the stack pointer register inside the processor to FF. So there is no way of recovery once the reset has occurred!

The only way of disabling the key is to physically disconnect it, possibly fixing an alternative switch somewhere else, or to gate the signal with another one such as an annunciator output. In either case the warranty on your machine would be void. If you just wish to avoid it being pressed accidentally, the later models of Apple have a switch under the keyboard which couples it to the CONTROL key, so that both must be pressed together to cause a reset. It should also be possible to make an NMI do a RESET by resetting the stack and vectoring to the appropriate code, and this could easily be enabled or disabled by software. Why not try it? – **John S. Littler.**

## Shape tables to disc

COULD you send me a method of loading shape tables on to discs if there is one?

I have looked for one unsuccessfully in all my magazines and the manuals. This information would help me with a lot of my Apple computer programmes – **G.A. Boyd, Palace Hotel, TE Aroha, North Island, New Zealand.**

● A shape table, once created (p.92 of the Applesoft manual), is merely a series of bytes forming plotting vectors and an index to these vectors. The address at which the table starts (let's suppose it is 16384) is known as is its length (suppose it is 210) and so the table is easily saved to disc by the command BSAVE SHAPE TABLE, A 16384, L210 (p.92 of the DOS manual).

Now to be used by a Basic program, the shape table has merely to be BLOADED back into memory at any convenient location safe from Basic. Or the bytes may be converted to decimal equivalents and POKEd in from DATA

statements (see Windfall Vol. 1, 5, p33 for a program to do this, and also Vol. 1, 9, p23).

Basic has to be told where the table starts by calculating the high and low order bytes of the address and POKeing these into 233 and 232 respectively. More information on this and other ways of dealing with shape tables are given in Windfall Vol. 2, 5, p26 and 6, p72). We are planning a further article that deals specifically with shape tables. There are a number of programs, both commercial (eg Shape Manager reviewed Vol. 1, 6, p23) and contributed, which create shape tables and save them to disc. – **Max Parrott.**

## Non-running programs

REFERENCE Mr A.M. Oldacre's letter (November Windfall), I can sympathise with his frustration – particularly as med-res graphics will NEVER BRUN and do anything immediately apparent. In fact it sets up an ampersand vector which is used later in the series. Hopefully Mr Oldacre has now managed to get his colour demonstration by entering monitor (CALL-151) and typing B89G.

His letter does raise a number of points worth consideration. Other magazines often include a short paragraph in each issue showing how to enter the code.

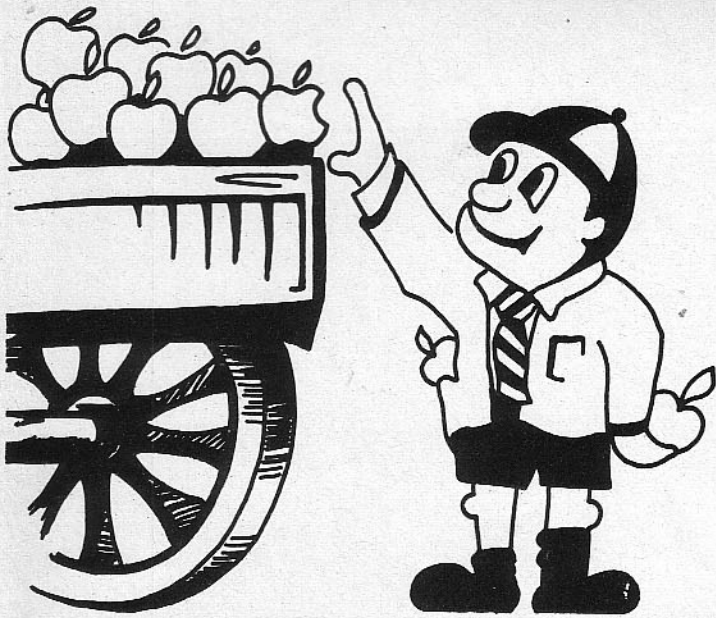
Windfall mainly features articles assembled using the DOS 3.3 Toolkit. It may be worthwhile giving the Synertek and Toolkit equivalents where these differ.

I would guess that most of Mr Oldacre's problems stemmed from the poor printing at OBC7 : 90 DF which was my fault and not Windfall's. However, it would have helped him if the code could have been BRUN. I would suggest this may be a rule contributors could adopt in future.

Finally, I hope Mr Oldacre persists in typing in machine code programs as he will gain many useful utilities to use from Applesoft. – **Ed Peach.**

## Teacher's plea

I WOULD like to contact other readers who are engaged in the teaching of 'O' level computing studies. I find that I am working in not-so-splendid isolation! – **Miss D. Sweeney, Manor House School, Raheny, Dublin 5. Tel: 316782.**



Monthly review of  
Apple in education

## Helpful for teachers new to computers

By CLIVE  
TALLON

Head of Computer Studies,  
Ribblesdale School,  
Clitheroe.

AS the demand for computer education continues to grow within all age groups, so must the demand for good resources to support the many different kinds of course which are currently evolving. "An introduction to microcomputers" must be the most popular course title offered by a variety of institutions, but the bodies catering for the largest number of students must still be our schools with such courses as CSE and O level. This and the computer's capability to "teach" about itself must have prompted Pedagog O level aids from Pedagog Computer Services.

This package, which comprises five discs with a user's guide selling for about £60, is aimed at the person faced with the task of teaching computer studies in a secondary school situation. Good computer studies teachers are not readily available in quantity within secondary schools for a variety of reasons, the prime one being that the subject is relatively new.

An optimum method of teaching has not yet been established and there is insufficient collective knowledge and experience to achieve broad agreement.

With these points in mind the users' guide not only covers the Pedagog aids but also gives excellent support to teachers who are new to the task offering a great wealth of teaching ideas. Ten sections correspond to the 10 programs on the discs. They are:

Computers and what they are not; some thoughts; Notes on history; The central processor; Input devices and their media; Output devices; Magnetic tape; Magnetic disc; Computer applications; People in computing; and Computers and life.

Key topics in each section are dealt with concisely and clearly, and there are valuable lists of classwork suggestions which could easily be converted into lesson plans. Incidentally, page references are made throughout the manual to the

excellent book "Computer Studies, A First Course," by Shelley and Hunt.

My first real problems, and criticisms, started with the software discs. Of the five supplied — each supposed to have two programs — one did not appear to have anything on it, and two did not boot correctly, generating the error message  
OVERFLOW IN LINE 12.

This was overcome by displaying the catalog of each disc and running all the files until the correct one was found. Still, two out of five is not a very good score.

The next problem wasn't long in surfacing. Each program has its own menu of three or four units comprising several screens full of text and incorporating an autopause facility. I found it infuriating when an average of 110 words disappeared every 10 seconds or so. No command to cancel this was immediately apparent and although the user's guide referred to "hold the frame" and "slowly undulating the space bar," the process refused to work on all but the last frame of the unit.

However after several hours I found that pressing the space bar once within the 10 seconds allowed by the autopause froze the frame. Restarting the page scrolling was achieved by pressing the space bar again, but sometimes up to four presses were required to get the frame moving — a serious fault in a package which will be used by complete beginners.

Control and a definite response to commands are two of the most appealing aspects of machines, and in a classroom situation a delay in the Apple's response to input could be disastrous. While on the subject of control, there was no facility to go back to the last frame as suggested by the guide. I also found it possible to crash the program by giving the wrong responses to menu prompts.

A program like this ought to be "child-safe" and

# Programming the Apple — for kids of all ages

a possible enhancement would be a more user-friendly control menu incorporating commands such as A for Autopage, M for manual with space bar to continue, etc.

Each frame of the text conveys information pertinent to the unit with very few graphic representations. The text is written in an entertaining and informative style suited for use with young students and is as comprehensive as it is accurate. It is worth mentioning that the presentation format would suit any other subject just as well as it suits computer studies.

Some of my colleagues disagree, suggesting that against this format a good text book is far superior, while Pedagog themselves claim that "the young are conditioned to accept their wisdom from a cathode ray screen." While I cannot see the novelty of a screen lasting very long, controlling a machine must be more interesting than holding a book.

Where the package does differ from a book is in the provision of a quiz at the end of each unit, where the computer allocates questions to random numbers (within the range of the number in the group). At the end of the quiz the Apple displays a class result. While this is a long way from CAL or CAI, and not even close to programmed learning, it is encouraging to see an English software production of this nature.

Most of the work done on CAL seems to have been done in America, presumably as an offshoot from huge projects such as PLATO. Different methods, language and culture limit the usefulness of American educational material, but the prospect of British programmers and teachers taking advantage of the American experience to produce British-orientated packages is exciting.

I have highlighted some of the faults of the Pedagog package. However any school with a 48k Apple and one disc drive would find it a useful investment.

It would be of greatest use to a teacher new to the subject who is looking for lesson ideas. The user's guide claims "the suite is very effective for class use provided adequate monitor screens are available," but this is a dubious claim even if the modifications I have suggested were incorporated.

I agree that it would be useful for small groups, although its greatest use would be as an individual resource where a student could work through the programs at his or her own pace. Outside lesson time the package would be of use as a "library" resource to support other computer studies courses as well as being a useful source of information about computers in general. 🍏

IT IS easy for experienced programmers to forget just how confusing and daunting can be a user's first confrontation with "the computer."

Many people have still to break away from the idea that "if it is to do with computers it must be difficult and the domain of experts." Often, it is only with hindsight that you realise that breaking down that basic fear of the unknown is a simple, step by step process.

When I first showed "Kids and the Apple" by Edward H. Carlson (published by Datamost, £10) to a Windfall technical editor he flicked through it and commented: "Yes, it looks fine, but I wonder if it is like so many others of its kind — teaching programming commands but not teaching you how to program."

When I showed the book to computer illiterates and to first-time users who were struggling to comprehend both the Apple and the manuals, they flicked through it and responded: "Tremendous — can I borrow it after you, please?"

The latter reaction sums it up. If nothing teaches you to program other than experience, then this book will point you in the right direction to start accumulating that experience.

One of the people who had a look at the book has just turned 60. The other is 12. The book would suit both of them, as well as anyone in between, including teachers and proud parents wanting to keep one step ahead of their technological tinies.

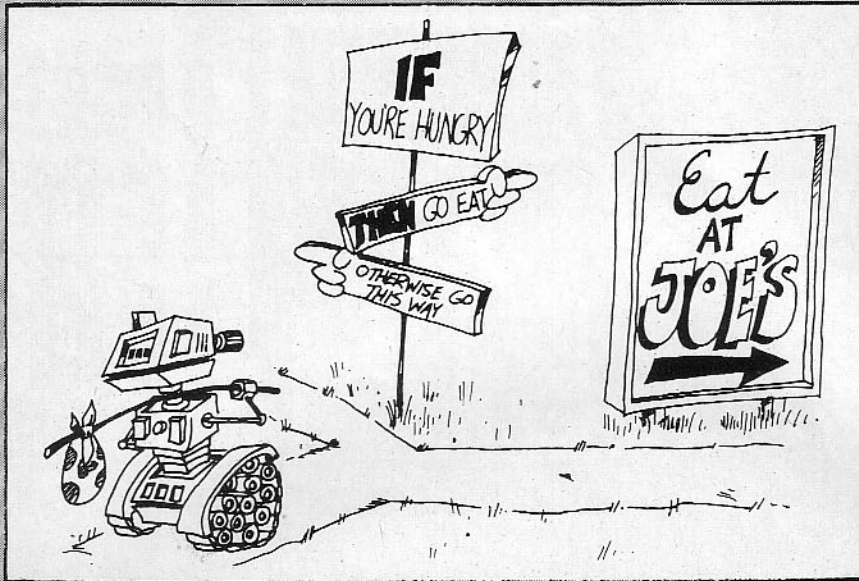
However, it is definitely a doing book and its pages have to be worked through on an Apple. Reading it on its own may do little for the beginner other than reinforce a sense of bewilderment.

The book has a strong ring binder and is arranged in 33 lessons in three main sections — introduction; graphics, games and all that; and advanced programming.

Appendices cover disc usage, saving to tape, Applesoft reserved words, a glossary, an index of commands and functions explained in the book keys and error messages and answers to the various lesson assignments (usually in the form of listings.)

It starts by telling you HOW TO TURN ON THE COMPUTER and dives straight in by getting students to type ANYTHING, clear the screen and develop control over words without getting bogged down in a detailed description of the keyboard.

Remember, a complete beginner has nothing



---

---

By DAVID CREASEY

---

---

assimilated, no building blocks available to fall back on and to relate to when he approaches a micro. The idea of being taught about a keyboard or taught about programming isn't exciting. But here, suddenly and within a few pages, the child is getting the computer to flash, make a noise, print his own name, change speed . . .

And after teaching the bare bones of programming mechanics the lessons move quickly to the point where interesting programs can be written, dealing with editing, debugging and user friendliness.

The book was written to teach serious computing to children of middle school age and to help parents help their children learn programming. The format consists of a well illustrated lesson (simple, easy to follow and entertaining) preceded by instructor notes. These notes give more detail, sum up the lesson and suggest review questions.

For a 60-year-old "kid" it is like being taken back to the comfortable world of "no previous knowledge assumed" type primary school lessons - but at a school where you can be both pupil and teacher at the same time.

For the average youngster, the author says: "You will learn how to make your own action games, board games and word games.

"You may help your younger sisters and brothers by writing drill programs for arithmetic facts or spelling. Even your own school work may be made easier by programs you write."

In a note to the parents, he says: "Learning to program is not easy because it requires handling some sophisticated concepts. It also requires accuracy and attention to detail which are not childhood traits. Children will be well rewarded if they can stick with the book long enough to reach the fun projects that are possible once a repertoire of commands is built up."

He sums up his aim, and what I believe is his achievement, by saying: "I view this book as teaching programming in the broadest sense, using

the Basic language rather than teaching Basic."

One indication of whether he succeeds comes from 12-year-old John Stevens of Stockport in Cheshire who wrote:

"Sometimes, once you have learnt Applesoft and go on the computer, you don't know what to write your programs about but this book gives you many useful ideas. It also gives you a lot of fun while learning something that can be quite serious.

"In certain parts of the book there are cartoons which, if you have been working on one thing for hours just because you don't understand it (which is unlikely), they give you a break."

The author leaves such mundane topics as saving to disc until chapter 14 because, he says, the process of programming is being emphasised, not the end result of useful programs.

And towards the end, when you discover the EDIT and the RUN modes, he comments that the material "is placed rather late in the book despite its fundamental nature because it is abstract and because we did not wish to slow down the race to mastery of the core commands in Basic."

Once all the elements are in place for a student to write a substantial program, he is taken on a tour of some programming short cuts and their potential pitfalls, graphics routines and games programming. He is shown how to write a long program (a hangman game) but always with gentle guidance rather than rigid rulings and with an excellent use of REM statements to explain what a particular program, subroutine or line is doing.

Finally the book considers the concept of user friendly programs and gives clear suggestions about layout and design of a program. It is an outline that tells you to "put the main loop at the beginning because it will run faster there, to put the starting stuff at the end because it may be the biggest bit of the program and you may keep adding to it as you write, and to put sub-routines into three places depending on whether they are to run fast, are for starting stuff or are for the rest of the sub-routines."

It explains why and how to set traps for errors and concludes: "Don't forget you are a user of your own programs too. Be kind to yourself."

Many beginners can't actually conceive of themselves writing programs at all, never mind user friendly ones, but with the use of this book it becomes a realistic goal.



# Applecart

## Educational, with an American bias

By CLIFF  
McKNIGHT

HANDS up if you remember Jeremy Hawk. Well, that dates you, doesn't it? If you don't remember him, then you probably don't remember Criss Cross Quiz, a program on ITV in the late '50s. Chances are, though, that you have played noughts and crosses at some time, or Tic Tac Toe if you are American.

Tic Tac Show is an Apple version of the television quiz show. It comes from Computer Advanced Ideas (you thought CAI stood for computer assisted instruction, didn't you?) and requires 48k and DOS 3.3.

It plays the game in quiz show format, complete with presenter Carol whose mouth moves as her speech "bubble" fills up. You can either play against another player or against the computer. Although the computer claims to be "thinking" when asked a question, its thought processes are infinitely less fallible than yours or mine. It hardly

ever gets a question wrong but then it *is* holding the answers too!

The disc comes with a variety of subjects, from world capitals to sports trivia, and several other subject discs are available. The review disc came with Subject Disc 4 on the reverse side and this is titled "History". It contains close on 600 questions of a historical type although they are subdivided into categories like "women", "myths", "elementary science" and so forth. As you might imagine, there is a distinct American bias in most of the subjects.

In addition to the subjects already available there is also an option to create subject areas of your own choosing. This is remarkably easy to do, and the manual is a model of clarity as far as the instructions are concerned. The manual even shows you how to carry out a hardware mod in order to get lower case, and the program checks whether lower case is available when you start to create a new subject area. Normally, of course, all text you enter is in the standard Apple upper case.

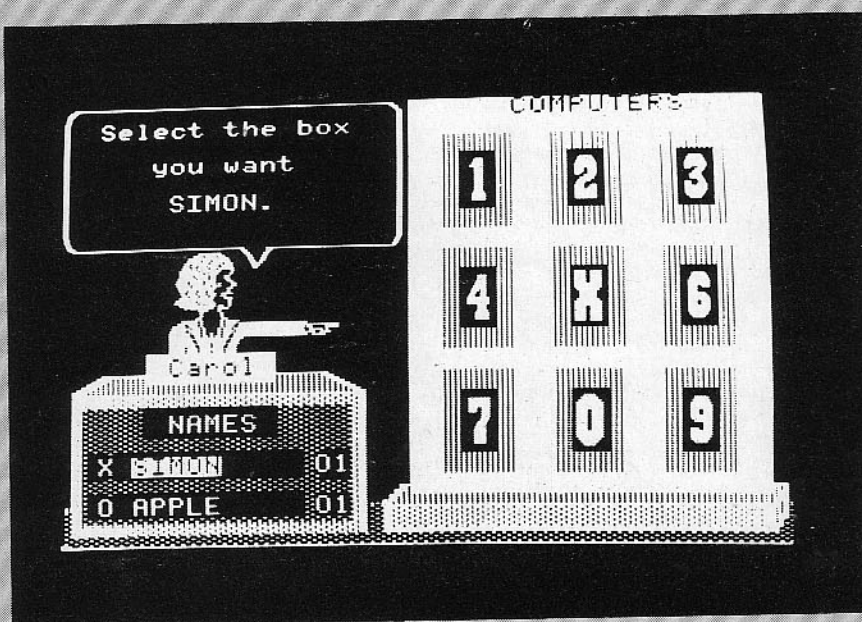
I encountered two problems with Tic Tac Show. The first was that the disc frequently failed to boot or failed to access the subject disc successfully. I'm not sure why this should have happened intermittently but it seemed to be a disc fault rather than a program error. However, it happened often enough to be quite irritating.

The second problem concerned a spelling mistake. I know I always find at least one but in this case it made me lose the game! I was asked "Who was the god of poetry? He was also associated with the sun". Easy, I thought, and entered "Apollo", whereupon Carol said "Nope, next time try Appollo" and promptly awarded the game to the computer.

I know the Americans sometimes spell words differently but I didn't know they had their own versions of Greek proper nouns. Next time I see some footage of the space shots, I'll look closely to see if they got it right on the side of the rockets.

Apart from these two problems the game is quite enjoyable. The manual describes it as an "educational tool" and it would be easy to use it for testing knowledge of a particular area, although no reporting facility is available. The game makes some use of both colour and sound, although the sounds can be turned off if desired.

I think Tic Tac Show could be a useful addition to a school's software library or used at home on rainy days as a change from arcade games. ☺



# Developer

The DEVELOPER is a powerful program development aid for programmers writing in APPLESOFT.

DEVELOPER speeds up the program development cycle by providing facilities to make rapid local and global changes to program text. DEVELOPER consists of 4 modules, each of which is accessed by typing the ampersand '&' followed by a single command letter and a line number or string of characters. DEVELOPER is designed to operate either on its own or in conjunction with the RENUMBER utility present on the DOS 3.3 SYSTEM MASTER or the APPLESOFT PROGRAMMER'S ASSISTANT supplied with the DOS 3.3 TOOL KIT. Once installed DEVELOPER is completely transparent and does not interfere with DOS or APPLESOFT.

The 4 modules that comprise DEVELOPER are:

- \* A Line Editor offering rapid cursor moves, insert and delete.
- \* A Global Editor allowing all lines containing a specified string to be listed and an optional string substituted for it.
- \* A Lister with forward and reverse list, single step and suspend list.
- \* A System Variable Dump displaying much useful information on memory status.

SYSTEM REQUIREMENTS DEVELOPER requires a 48K APPLE II+ with APPLESOFT in ROM or language card and at least one DISK II running under DOS 3.3.

£19.95 + VAT

# Merlin™

By Glen Bredon

MERLIN is more than just an assembler. It is an extremely powerful macro assembler, with a sophisticated editor, combined with numerous other files and programming utilities into a truly remarkable package. MERLIN ASSEMBLER. Besides having the common features you would expect, allows such enhancements optional writing of object files directly to disk, linking files to assemble source listings otherwise too large to fit in memory at once. The source listing can also use macro routines.

MERLIN will read and write text files as well as binary source files, and is often capable of using files generated on other assemblers with little or no adjustments. The global search/replace function of the editor also makes it easy to change pseudo-ops that may have been peculiar to the other assembler.

Additionally, MERLIN supports SWEET 16 op-codes as well, and the manual includes a short tutorial on this subject by Steve Wozniak, co-founder of Apple Computer Inc. In addition the MERLIN assembler, package also includes:

SOURCEROR: This generates pseudo source code from raw binary data.

MACRO LIBRARY: A library of commonly used macro definitions and fundamental operations.

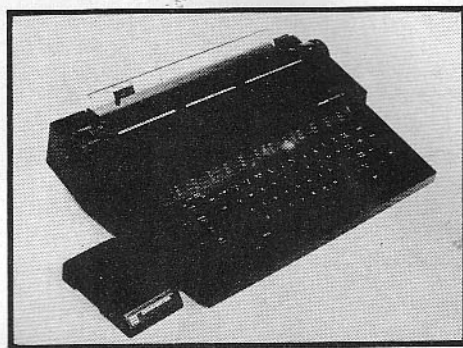
SWEET 16 SOURCE: A source code for a transportable SWEET 16 interpreter, usable even without the Integer non-Auto Boot ROM.

APPLESOFT SOURCE: If you have Applesoft in ROM or LANGUAGE CARD, you can use utilities included in the MERLIN package to create a fully labelled and commented listing of Applesoft BASIC. MERLIN requires a 48K Apple, with language or RAM card. MERLIN is compatible with most 80 column cards and supports upper/lower case entry, including the one-wire shift key mod and commercial lower case devices.

£49.95 + V.A.T. (inc. P & P)

**LEICESTER**  
computer centre limited  
67 Regent Road, Leicester LE1 6YF.  
Tel: 0533 556268

# Buy our £475\* Daisy Wheel Printer for your computer and you have an Electronic Typewriter absolutely FREE



The T/Printer 35 is the lightest weight and lowest cost daisy wheel printer you can buy for your computer. So it will fit within your budget and you can carry it wherever you take your micro. Yet it is tough enough to give years of reliable service. Interchangeable typefaces (standard Olivetti 100 character daisy wheels), variable pitch, multiple copies—all the features you would expect of more expensive word processing printers.

Yet the T/Printer 35 costs only £475 with parallel interface. Operating speed under computer control is approximately 120 words per minute of letter perfect output. What typist can equal that?

Then when you're finished using it as a computer printer, the T/Printer 35 is ready to go right on working as an electronic typewriter.

That's the dual-purpose T/Printer 35—the versatile computer printer that fits your budget.

*\*The T/Printer 35 costs £475 with Centronics compatible parallel interface. With RS-232C interface it costs £535. Prices listed are exclusive of VAT.*

# DATARITE

Datarite Terminals Ltd  
Caldare House  
144-146 High Road  
Chadwell Heath, Essex RM6 6NT  
Tel: 01-590 1155

# SCREENWRITER II

---

---

**If the Greeks  
have a word  
for it,  
then we have  
the technology**

SERIOUS users of word processors have often found the Apple II limited, partly because of its upper case only, 40 column screen format, and partly because the processing programs have been indifferently different. However a new package, Screen Writer II (which replaces SuperScribe II), from On-Line Systems in California, has enough significant new features to make it well worth consideration by everyone.

Screen Writer II is an impressive word processing program, especially since no extra hardware is necessary beyond a 48k Apple II Plus and at least one disc drive. It displays software-generated upper and lower case characters – you could use a Greek alphabet if you wished – without splitting words at the end of a line in either 40 or 70 column format, handles documents larger than memory and allows complete string search and replace.

It has full cursor control by character, word, line, or page as well as jump to text beginning or end. Deletions may be made by character, word, line, or text block – and a deletion may be retrieved if it is not larger than 256 characters. Any key can be defined as any character or string (Macros) and because the program works on DOS 3.3 text files, Basic programs or data can be edited. It will count words and characters.

For printing the program features customised page headers, footnotes, numbers and dates, as well as full hyphenation, justification, line look-ahead, underlining and bold type (depending on printer model), indexing, mail list merging for form letters, etc. Finally, it provides for printer spooling – the ability to print while typing – which is a powerful tool for those with fast fingers and slow printers.

This is the first word processing program for the Apple II that offers all these features in one package at a price less than an 80-column board.

Now for a more detailed look. Although Screen Writer II uses ordinary DOS, only the discs supplied (two) will boot properly and thus run. We guess that if the dog chews up the master disc or someone spills hot coffee on it, you send the disc back to the company for a new one, meanwhile using the spare. After booting, a simple menu gives you the choice of Editor or Runoff with or without RAM card or quitting. Unlike the old SuperScribe II, one can use this program with a 16k RAM card so that the Editor and Runoff are co-resident in memory. This is an excellent improvement, especially since the cost of

---

---

By TED LODER and  
ANDREW BANGHAM

---

---

RAM cards has come down recently.

The Editor uses the bottom four lines of the screen to display editing information, namely the current command, screen tab stops, in and out file names, editing status flags and the amount of remaining space. Initially the cursor flashes next to the COMMAND prompt. A large variety of commands are available, including DOS commands.

One of the commands is C which allows the user to move to screen work space, which is the top 20 lines on the screen where text may be changed or inserted. In either of these modes one can issue commands without having to return to the command level. This is done using control characters (*press CTRL and the appropriate other key*) which include by default such commands as move forward or backwards one character, word, line, or page, etc.

From the command level any key or control character can be defined to represent any character string (including commands and control characters) called a Macro. Macro tables can be saved onto disc and, if required, automatically booted into memory along with the Editor. Useful

if you wish to tailor the package to your own printer control sequences, etc. You could even define a control key to represent "the hero said . . ." for that novel you are writing.

It is also easy to insert control characters in the text. A CTRL-x allows the next CTRL character to be input literally into the text, which is often necessary for printer control. CTRL characters are given their own special display characters, so they are easy to identify on the screen.

The ESC key controls upper case. However the shift keys can be made to work like a normal typewriter by connecting them to button 3 (pin 4) of the paddle port inside the Apple. Complete instructions for this modification are in the manual, and it is really worth doing. If done neatly it doesn't affect the use of the paddles.

Screen Writer II solves the problem of having no lower case on the Apple and exploits the solution to give more than a normal shift key. For example, it offers not only A,a, CTRL a, but also CTRL A (shift/CTRL A) as a new code. It effectively adds an extra bit to the Ascii coding. This is elegantly exploited by, for example, CTRL/z, which by default means move on one word and shift/CTRL/z (CTRL Z) which means move backwards one word. All the cursor move commands, including FIND, can go backwards in the same fashion, and all these codes give plenty of scope for defining macros. Finally, the program allows one to case change characters or even whole blocks of text.

# SCREENWRITER II

---

---

Typing is a pleasure since Screen Writer II uses a keyboard buffer which cannot be overrun. Under some circumstances, such as inserting text at the top of the screen in the 70 column mode, fast typists may have to wait for the display to catch up with their typing.

Both upper and lower case letters appear on the screen, as well as a c/r for carriage return and — most usefully — underlined small upper case letters for the control characters. All characters are software-generated.

One of the more interesting features of the package is that it comes with a 70 column character set (also software generated) selected by typing 70c on the command line. We found the 70 character set just a bit small on a nine-inch monitor, but it should be fine on the 12 inch variety. We do use it for previewing the text, since it nearly doubles the amount of text on the screen. Of course, an 80 column card offers slightly better resolution, but the extra 10 columns and clearer display have to be considered against the cost.

Both FIND and FIND AND REPLACE commands are available. These can be global if required. Furthermore, a FIND or REPLACE operation can be repeated without respecifying the string to be found, both forwards and backwards merely by issuing the command followed by a return, and wildcards can be used in the FIND string. Once found, the string is placed in the middle left of the screen for review before any changes are made.

Moving paragraphs or larger text blocks around is often necessary for most writers and is easily done here by marking the beginning and end with control characters and then writing it out wherever needed, or even saving to disc — provided there are two drives. The reverse, merging disc files into the text, may be done as well. A working file on Screen Writer II may be up to 64k characters in length, longer than nearly every other word processor for the Apple II. It handles files this large because while you are editing parts of the text are moved to and from a disc buffer file automatically set up by the program.

Incidentally, the program keeps the beginning and end of the text in memory all the time no matter how much text has been entered, and so it is possible to move very quickly to the start or finish of the text.

Up to this point we have discussed the Editor program. The other option on the main menu is Runoff, used to print the results. This has a main menu and separate menus to control all aspects of the printing process. Basic formatting of the output can be done from the menu at the time of printing, or from format commands embedded in the text which override menu formatting and provide more options. A very nice feature here is the

ability to preview (Test print) formatted output on the screen using the 70 column mode.

The commands, which take up nearly two pages of the manual, include yet more ways of globally replacing strings. The .RE command for example allows the replacement of one string by another after formatting, which is useful for controlling the printer, since printer commands can be given control codes — ignored by the formatter — which are replaced by the proper characters at this time. This allows justification to continue in spite of embedded printer commands.

The usual margin settings, justification formats, tabs, indents, etc. are available, including proportional spacing on some printers. One can insert page headings which include date, page numbers (decimal or Roman), and up to a four line title, all of which can be placed on a designated line in the centre, or to the left or right sides. These can be turned off or on at will.

The handling of form letters is important to many business users of word processors. Screen Writer II does this in the output file section of Runoff which simply links the address (data) file on disc to the letter so that numerous files may be linked together for unattended final printing. It also allows one to specify which fields within any particular data file record are to be used.

Specific information may be inserted in each letter during printing at a predetermined location. Runoff can also branch to print different texts depending on the results of a logical comparison based on numeric or alphanumeric data stored in the address record. If one specifies acceptable hyphenation points in long words by inserting a specified character, then Runoff will, if necessary, hyphenate long words during printing.

Another nice feature for the book writer is Screen Writer II's ability to produce an index of key words at the time of printing, including the page numbers. This is a powerful tool if the author is thorough, but remember that an index is only as good as the person setting it up, since the typist must mark the appropriate words.

Bold type and underlining may also be controlled during printing if the printer is so capable and Runoff will also stop the printing process at a predesignated place and display a message on the screen, for example: "Change printerheads" for different print styles.

Any of the Runoff specifications may be saved on disc and automatically loaded every time it is run. For specific format situations, i.e. business letters, the manual suggests that a text file be stored with all formatting commands set up on a library disc. Examples are included in the instruction manual.

The manual is a pleasure to use,

especially in comparison with the old SuperScribe II manual. It is presented in a reduced size three-ring notebook and, at 187 pages, appears slightly daunting at first. However, to get started all you need to read is Section I — "The Mini-Writer", a well written introductory summary of the programs which even comes with its own reference card.

Section II, "The Creative Writer", gets down to the details of what the programs can do in the areas of advanced editing, form letters, formatting and producing reports and manuscripts, all with many examples. This is a section to be read in little chunks when needed.

Section III contains summaries of all commands and short descriptions of their use. This is followed by an Appendix with sections on disc care, shift key modification, custom print drivers, converting SuperText II and AppleWriter files (programs included), error messages, etc.

Finally, to our great joy there is a comprehensive seven-page index. (On-Line Systems obviously responded to the pleas of SuperScribe II users!). Its use showed that SuperScribe II had many features which we had not identified during a year's use — demonstrating the importance of a good manual.

Well, after all this praise, what of the problems? There are a few things such as file deletion which could be made easier. Also, if a 16k RAM card is not being used the Editor and Runoff are loaded separately and the text resides on a different disc. This results in a certain amount of disc shuffling, especially on a one drive system. Two drives are easier, though not essential.

It is possible however to get a quick unformatted listing of text from within the Editor, which will, if necessary, flag all embedded control characters and format commands. Another minor limitation is that error messages such as "file not found" are sometimes only displayed for a subliminal moment too fast to read — and thus useless.

These criticisms are trivial, however, and in our view this package is first rate and even outclasses the much vaunted Wordstar, particularly when its overall cost is included on the balance.

Up to this point we have reviewed the "normal" Screen Writer II package which costs £78 and is obtainable from Spider Software. The "Professional" package incorporates On-Line Systems' spelling correction program called "The Dictionary", a word book disc containing about 28,000 words. Another 2,500 words can be added for customising, and words may be added or deleted from the word book at any time. According to the manual the word book disc is supposed to be copiable by COPYA or any normal disc copy utility. However, since ours would not copy, we have not reviewed it here. 🍎

## Hiding your bytes from

MANY programmers wish to speed up a portion of their Basic program and so they write a small machine code routine to effect their goal or they construct a shape table to effect a picture. Now they have to put these some place where Basic won't overwrite the bytes. Generally speaking there are three favourite spots. If it is short enough there are a number of bytes available and safe from Basic beginning at \$300, hence you will often see the command CALL 768. Another favourite place is at the top of available memory, immediately under the area used by DOS, normally \$9600 upwards. The highest location available to basic plus one is stored in \$73 - \$74 (115,116) in the usual reverse order (*usually these will contain \$0 and \$96*) and you may find them being PEEKed before being manipulated either by POKing or by issuing the Applesoft command HIMEM.

The third place to tuck a machine code program or shape table is immediately after the program text and before the beginning of the variable space pointed to by \$69 - \$6A. There is a pointer to the end of the program text at \$AF - \$B0 (175,176) and manipulating this has the advantage that its value dictates how much memory is actually saved to disc or to tape when the SAVE command is issued.

Hence if the code is put after the Basic text (*and this is easy to find by looking at \$AF - \$B0*) and then \$AF - \$B0 is adjusted to point to after the code the lot will be SAVED. More advantageously, once in position the Basic may be edited in the normal way and the code will be moved up and down in memory, always being protected by \$AF - \$B0. Moreover, when the program is subsequently LOADED then the pointers to the various variables (such as \$69 - \$6A) will be set up automatically by the Basic.

The problem remains how to find the beginning of your code after it has been moved around, but this is not so difficult since you know the length of it and you know the end of it from \$AF - \$B0. Let us pretend we have a shape table \$D1 bytes long (209 decimal) and the end of program pointer holds the values \$AD and \$1E, ie. \$1EAD is the address. Move the shape table to this address and calculate the new end of program pointer. This will be \$1E AD + \$D1 + \$1 or \$1F7F, so change \$AF - \$B0 to \$7F and \$1F respectively, SAVE the program and then reload it. (*The object of this is to let Basic set up the other pointers for you*). Now the difference between the old value held by \$AF - \$B0 and the new is \$D2 or 210 decimal, so inserting a line such as

```
ADD = PEEK(175) + 256*PEEK(176)
- 210
```

will tell you the start of the shape table. Basic has to be told where to find the shape table, the appropriate address is held in zero page locations \$E8 - \$E9 (232,233 decimal) so it is necessary to calculate the values needed. Remember we need to find the high order byte and

the low order byte of the address. A line of Basic such as

```
HI = INT (ADD/256):LO = ADD
-256*HI
```

accomplishes this, and so finally a line

```
POKE 232, LO: POKE 233, HI
```

tells Basic where the shape table is even if the program is edited. (*NB. This technique will not work for any machine code program which is location dependent, for obvious reasons*).

Now let's find another use for PEEKing and POKing using the information we

**MAX PARROTT  
concludes  
his beginners' guide  
to PEEKs and POKEs**

already possess. It is very easy to type NEW before SAVEing a lengthy program. However, you haven't actually lost the program text and it can be found again. What happens is that each line of Basic is constructed such that the first two bytes point to the start of the next line in the usual reverse order and the last line points to the location immediately after itself and here are stored two zero value bytes signifying the end. When you type NEW the Basic merely overwrites the links of the first line with two zeros so it looks like no program is there. Therefore to put these two bytes back as they should be (*assuming that nothing else has been entered*) a small program can be entered and executed in the immediate mode. It depends for its functioning on the fact that each line of Basic is sandwiched between two zero bytes. I will leave you to figure out how it works. It assumes at least a 48k system.

```
LOMEM: 38200:X = PEEK (103) +
PEEK (104) * 256:Z = X + 5:
FOR A = Z TO Z + 275: IF PEEK
(A) < > 0 THEN NEXT
```

```
A = A + 1:HI = INT (A / 256):
LO = A - HI * 256
POKE X,LO: POKE X + 1,HI
```

Short machine code routines are usually carried within the text of a Basic program in some way and then the code is POKed into the appropriate memory loca-

tions. There are three main methods. The first has the locations and values in decimal and a series of POKes puts them in. Let us use a short shape table to draw circles (*incidentally, this is the quickest way I know to draw them*). We will POKe the table into memory at \$300:

```
10 TEXT : HOME
20 POKE 768,1: POKE 769,0: POKE
770,4: POKE 771,0: POKE 772,
4: POKE 773,0
30 POKE 232,0: POKE 233,3: REM
BASIC POINTER TO SHAPE TABLE

40 HGR2 : HCOLOR= 3
50 FOR B = 1 TO 9: SCALE= B
60 XDRAW 1 AT 20,105
70 FOR A = 0 TO 63
80 ROT= A: DRAW 1
90 NEXT : NEXT
```

The second method carries the required values in DATA statements: These are READ and POKed in inside a FOR/NEXT loop. For example:

```
100 FOR J = 770 TO 790
105 READ K: POKE J,K: NEXT
110 DATA 173,48,192,136,208,5
120 DATA 206,1,3,240,9,202,208
130 DATA 245,174,0,3,76,2,3,96
```

This routine will create a note on the Apple's speaker which has a tone related to the value held in location \$300 (768) and a duration related to the value in \$301 (769). Thus these values may be POKed in and the routine CALLED from Basic. Try adding the lines

```
150 FOR DUR = 10 TO 2
STEP - 1
160 FOR TNE = 10 TO 100 STEP 2
170 POKE 768,TNE: POKE 769,DUR
180 CALL 770: NEXT : NEXT
```

Try experimenting with different values yourself. Of course, to know which values to POKe in to set up the machine code program requires a knowledge of such programming which you will have to find in books and articles elsewhere (*see the bibliography at the end of this article*).

I mentioned that the zero page locations may hold data as well as pointers to other addresses. One which is often used is \$21 (33) which holds the width of the text window. This is commonly used in the immediate mode when editing a Basic program with the classic command POKe 33,33. The effect of this is to reduce the width of the available text window from the normal 40 to 33. This has the desirable effect of suppressing the spaces

# big bad Basic

which are inserted when Applesoft lists a program and hence copying a line back into memory with the right arrow key doesn't introduce unwanted spaces.

Similar commands can be issued from within a Basic program, usually together with POKES to 32 (*sets left margin*), or to 34 (*sets top of window*) and 35 (*sets bottom of window*). These are explained on page 129 of the Applesoft manual. Related PEEKs or POKES which you may see used are 36 (the current horizontal cursor position) and 37 (the current vertical position). These are explained, together with some useful CALLs, on page 130 of the manual. Following pages describe POKES and PEEKs which will enable you to get more from the graphics, the games paddles and buttons.

Most of the graphics POKES have a different function from those already described. Rather than POKeing in new data or addresses these POKES act as switches, flipping between text, graphics, pages 1 and 2, and mixed graphics and text. Although well described in the manual they take some getting used to because the order in which these switches are flipped can be important. I would advise you to experiment with a simple Basic program and a straight line plotted from top to bottom on each of the high resolution graphics pages and one on the first low resolution page.

When you have a good understanding of these switches one more POKE, \$E6 (230), will prove useful. This one will dictate on which high resolution page drawing will occur, regardless of which page you are looking at. You POKE in the high order byte of the starting address of the page on which you want to draw. Hence \$20 (32) for page 1, \$40 (64) for page 2, and if you want, \$60 (96) for a mythical page 3. You will not be able to see this page, but you can draw on it and then quickly move the results to either page 1 or 2 to effect animation.

Some locations, often used, may well puzzle you at first. Errors may be discerned and subsequently dealt with by PEEKing 222 (see page 136), and after "using" the ONERR command a POKE 216,0 is necessary to cancel the effect. PEEKing -16384 enables the program to see if a key has been pressed. This may not seem useful at first, but in fact it is often the best way to "break out" of a piece of coding without halting its action unnecessarily. If a key has been pressed then the value held by this location changes from one less than 128 (\$80) to one greater than 127 (\$FF). Thus in the circle drawing

program above add the line

```
85 IF PEEK ( - 16384 ) > 127
    THEN 100
```

Now when this location has been "read" it should be cleared, and this is easily done by another POKE, so add the line

```
100 POKE - 16368,0: TEXT
```

The loop part of the program can now easily be exited, but its action is not stopped. This PEEK (-16384) is even more useful in that the value which it contains is related to the Ascii code of the key pressed. Now I said that the whole numbers which bytes can contain can be made to express characters etc, if certain rules are obeyed. One set of rules is known as the Ascii code, where certain numbers signify certain letters, etc. For example "0" is \$30, "1" is \$31, etc, up to "9" (\$39). In the same way "A" is \$41, "B" is \$42 ... "Z" is \$5A. When the keystroke has been detected with a PEEK(-16384) and subsequently cleared with a POKE -16368,0 location -16384 will hold the value belonging to the key. Also before the POKE -16368,0 is issued to clear the location the value held by it is again related to the key pressed by the Ascii code but each value has \$80 added to it (i.e. 128 decimal). The full set of Ascii codes are given on page 15 of the reference manual and the higher values from the keyboard (*technically known as having the high bit set*) are listed on pages 7 and 8.

Now that we know something of the Ascii code we can use it to convert decimal numbers to hexadecimal ones. The following program will do this one-way conversion: It is not the quickest way to do it but it is understandable. I leave you to code the conversion the other way.

```
10 HOME : VTAB 7: PRINT "ENTER Y
    OUR DECIMAL NUMBER"
20 INPUT "ENTER 0 TO FINISH ";N
30 N = INT (N)
40 IF N > 65535 OR N < - 65536
    THEN VTAB 15: PRINT "SORRY!
    CAN'T HANDLE THIS": GOTO 140
45 IF N < 0 THEN N = N + 65536
50 IF N = 0 THEN HOME : END
60 PRINT : PRINT : PRINT N; " = $
    ";
70 HI = INT (N / 256):LD = N - H
    I * 256
80 H1 = INT (HI / 16):H2 = HI -
    H1 * 16
90 L1 = INT (LD / 16):L2 = LD -
    L1 * 16
```

```
100 X = H1: GOSUB 170
110 X = H2: GOSUB 170
120 X = L1: GOSUB 170
130 X = L2: GOSUB 170
140 PRINT : PRINT : PRINT : PRINT
    "PRESS SPACE BAR TO CONTINUE
    "
150 GET T$: IF T$ = "0" THEN
    HOME : END
160 GOTO 10
170 IF X < 10 THEN PRINT X;:
    RETURN
180 X = X + 55: PRINT CHR$ (X);:
    RETURN
```

Others POKES which you are likely to come across frequently are \$3F5, \$3F6 and \$3F7 (1013 - 1014) which will be used in conjunction with the & command, which is not well documented in the manual. What happens is that the & causes the Basic interpreter to look at \$3F5-\$3F7 for a machine language command to jump to another location where the programmer has put his machine code. It is another way to start a machine code routine from within Basic. There is a third way, using the USR( ) command. When Basic finds this command it looks at locations \$A, \$B and \$C (10 - 12) for an address to which it will jump. The advantage of this command is that a value can be transmitted easily to the code routine and an answer returned to the Basic via the expression within the brackets and the value of USR returned.

## Bibliography

Besides the excellent Applesoft Manual, Apple Reference Manual, and DOS Manual, there are a number of books, presently available, from which much information may be gleaned.

Information from the manuals has been gathered together in "The Apple II User's Guide", by Lon Poole with Martin McNiff and Steven Cook (Osborne/McGraw-Hill).

DOS is extremely well documented in "Beneath Apple DOS", by Don Worth and Pieter Lechner (Quality Software).

An atlas and gazetteer to the Apple's memory locations is provided by "What's Where in the Apple?", by W.F. Luebbert (Micro Ink).

Introductions to machine code programming and assembly language, specifically for the Apple, are provided by "Apple Machine Language", by D. Inman and K. Inman (Prentice-Hall), "Using 6502 Assembly Language", by R. Hyde (Datamost), and by "Apple II Assembly Language," by M.L. de Jong (Howard W. Sams and Co.)

# Z80 cards for the Apple II

SINCE the advent of microcomputers, arguments have abounded on the suitability of available microprocessors for driving them.

Until recently, 8 bit machines have been dominated by two microprocessors – the Z80 and the 6502. The Apple II uses the 6502 which offers fast, high speed addressing. However, if you want to use CP/M which offers a flexible, wide-ranging operating system, you need the Z80. CP/M cannot be converted to run on the 6502.

An obvious answer to the dilemma over which microprocessor to go for is to provide a microcomputer with a choice. This was first accomplished on the Apple II by CCS who made a card containing a Z80 processor which plugs into one of the I/O slots, effectively turning the Apple into a dual processor system.

The 6502 remains the dominant processor, the Z80 CPU being accessed either by a PR# command or by booting a CP/M disc.

CP/M was developed by Digital Research Inc, and Microsoft provided two Basics, M-Basic and G-Basic, so that the user could have a complete Z80 system running on the Apple. Since then, at least two other manufacturers have produced Z80 cards, but the CP/M operating system comes only with the original Microsoft Softcard (£199 including Basic) and the Advanced Logic Systems card (£99 but without Basic).

The original card and that sold by U-Microcomputers (£89 but without software or Basic) both provide a full Z80 working environment but they don't handle interrupts. These are passed to the native 6502. The card sold by ALS handles interrupts itself and this is said to give an increase in speed.

So why bother with the Z80 card at all? The answer lies in the Z80's abilities to operate CP/M, a truly flexible and powerful disc operating system. It is more powerful than the DOS, with which we are familiar.

The easiest way to give some feel for its speed and flexibility is to describe

some of the available commands. The disc drives linked to the system are known as A:,B:,C:, . . . F:, and a command involving one of these letters logs that drive to the system.

DIR will list the directory of the disc, ERA will remove a file or a whole disc, programs are RUN by merely referencing the file name, and there are commands such as SAVE and LOAD; these are not the familiar Basic program commands but are to do with the memory itself.

On the disc there are software

copies whole discs, supporting one or more drives. APDOS converts Apple DOS to CP/M. It is also possible to read in programs from other computers and to send programs to other computers, but this does require knowledge of machine language.

Because of CP/M's extreme flexibility it is run on a wide range of micros, minis and mainframes and as a result there is a quantity of high quality software available.

In order to make CP/M readily portable between computers, the format of disc files has been standardised to a far greater extent than most other operating systems. This means that as well as simplifying communication between computers of different types, files created by one software package can be accessed by and manipulated by other programs. For example, you can do calculations using Supercalc which can later be incorporated in a letter produced by Wordstar.

CP/M is also an ideal system for mass data storage. The most expensive item in operating a database is the staff time spent in entering the data. CP/M data files are more flexible than those created under most other operating systems and, of course, your data files may be readily transferred for use on another machine even if it is not running the same data management program which created the files originally.

Having enthused over the versatility of the CP/M operating system, the time has come to sound a note of caution.

While CP/M itself is standardised, this consistency does not extend to the physical format of the discs, so that it is not usually possible to take a CP/M disc created on one type of machine and access it on another. The data has to be transferred from the memory of the source machine to the memory of the object machine and then saved to disc.

This can be done in a number of ways, the simplest of which is via a serial port. The CP/M system contains utilities to enable the user to effect transfers of this type (Download and Upload) but an in-depth knowledge of the workings of both machines is required.

The first of the two Basic interpreter discs supplied with the original card is

---

By PETER BRAMELD  
and MAX PARROTT

---

packages which are invoked by typing their names, which therefore become transient commands. For example, STAT will give a lot of statistical information on a file or a disc or on the system. PIP allows the copying of files from disc to disc, the combination of files to form new files, the listing of files or a facility to copy a file up to a certain string. ASM invokes the CP/M 8080 assembler (not a true Z80 assembler) which relates to the Z80's predecessor.

Why this is provided and not a full-blown Z80 assembler we don't understand. DUMP will dump a HEX file to the current display. ED is an Ascii text file editor used on its own or as the source creator for the assembler. It supports insertion and deletion of text, string searches, appending of text etc.

FORMAT initialises new discs. COPY

**'CP/M, a truly flexible  
and powerful  
disc operating system'**

called M-Basic and is the standard Microsoft 5.0. It has more commands than Applesoft. For example, AUTO provides line numbers automatically and RENUM allows for renumbering without running another program, as in Applesoft.

Variables may be double precision and have a 40 character significance. Two variables may be swapped via the SWAP command. There is a PRINT USING command and IF . . . THEN . . . ELSE and the WHILE . . . WEND structures are available.

Errors are flagged via two variables — ERR, which returns the error code, and ERL, which returns the offending line number.

LLIST prints to the printer as long as it's in slot 1. The usual Apple facilities of low resolution graphics and speaker access have been provided via simple commands such as GR, BEEP, BUTTON, but high resolution graphics are not provided, mainly because of memory constraints.

If you require high resolution you have to use G-Basic which is essentially

identical to M-Basic but with an HGR and a HPLLOT command. These are not quite the same as in Applesoft but should give no trouble. The only problem with G-Basic is that precious little memory is left for program storage, and it is only available to 16 sector disc drives.

You can gain some memory by using a language card since CP/M can be reconfigured to provide a 56k system simply by running a program provided on the discs.

A partially-compiled Basic called C-Basic is not supplied with the card but is available separately for £94. It is a semi-interpretive implementation which produces intermediate files which must be executed under the control of a run-time monitor.

There is also a truly-compiled Basic called CB80 (it costs £314) which takes CBasic source code and, with virtually no modification, produces object code.

To fully exploit the wealth of CP/M programs available an 80 column card is virtually a necessity. The choice of 80 column card is also important (see last

months' *Windfall*).

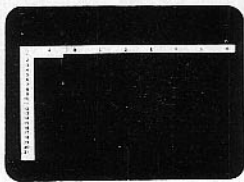
In view of the memory consumption by the Basic interpreters, a language card is also extremely useful, particularly if you are planning to use G-Basic.

Because of the complexities in running a dual processor system, the updating of the Apple mother board has resulted in some problems of compatibility between some Z80 cards and some versions of Apples. Some printers and interfaces require changes in the CP/M BIOS (Basic Input/Output System) to operate satisfactorily.

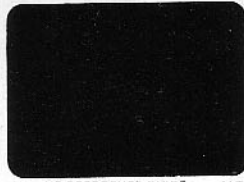
Because of these inconsistencies, we strongly recommend that you try out your intended system before committing yourself to purchase one particular product.

Perhaps the easiest way to do this is to assemble the total system and run a package such as Wordstar. Remember that at present the only Z80 cards which come complete with CP/M are ALS card and the Microsoft Softcard (which comes with comprehensive documentation). 🍏

## VisiCalc® was a good idea for then.



INTERACTIVE ELECTRONIC WORKSHEET YES



ON LINE REFERENCE GUIDE NO



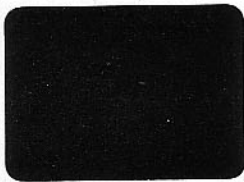
PLAIN ENGLISH PROMPTS NO



EXTENSIVE FORMATTING CAPABILITIES NO



SORTING CAPABILITY NO



INDIVIDUAL COLUMN WIDTHS NO



PROTECTED CELLS NO



"NAMING" OF CELLS OR AREAS NO



MULTIPLE LINKED WORKSHEETS NO

**The next generation.** First generation electronic worksheets were a good idea. They were early software management tools that could eliminate a lot of hours with a spreadsheet, calculator, pencil and eraser. Enter Multiplan, the next generation electronic worksheet that's as easy to use as it is useful.

**Make comparisons.** Compare Multiplan to any of the earlier electronic worksheets. We've given you some "prompts" above.

**Compare learning time.** Multiplan's tutorial book brings you up to speed. Fast. But Multiplan doesn't stop there. Multiplan's On-line Reference Guide gives you instant help if you have questions. It knows where you are in Multiplan and offers information related to your problem, right on the screen.

**Compare ease of use.** All Multiplan prompts are full length words or phrases. And Multiplan provides "naming," the ability to assign a plain English name to any

cell or area. "Gross Profit = Sales—Cost" rather than "AA44=AZ23—BK154." Which means you can work more intuitively. And faster.

**Compare utility.** Multiplan lets you link related worksheets so that information is transferred between them automatically. For instance, you can keep regional sales forecasts on separate sheets but link them with your overall company forecast. Then, just change the forecast for any region, and the company forecast sheet is updated automatically. Something you can't do with first generation worksheets.

**Compare reports.** Not just the work you can do, but the way you can present it. Multiplan's flexible formatting options allow you to produce presentation-quality reports. And its sorting capability lets you sort by either alphabetic or numeric order. So a sales manager who normally lists sales regions alphabetically could sort by amount sold and conveniently rank by sales performance. The