

The International  
**apple computer**  
users' magazine

# Windfall

Volume 3. No. 1 July 1983 £1

**Two new business packages reviewed:  
Micro Planner and The Spreadsheet**

**Beginner's guide to the world of  
PEEKs, POKEs and CALLs**

**Integrated computing  
for insurance brokers**

**How to save space  
with sequential files**

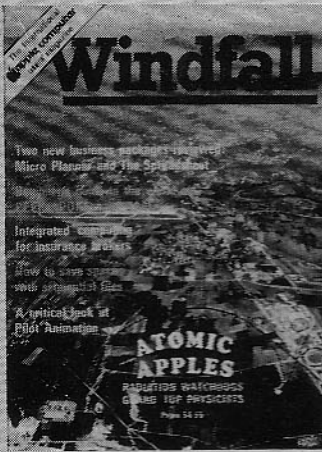
**A critical look at  
Pilot Animation**

**ATOMIC  
APPLES**

**RADIATION WATCHDOGS  
GUARD TOP PHYSICISTS**

Pages 54-55

AUG. E.



Vol. 3 No. 1 July 1983

*Managing Editor*

**Derek Meakin**

*Art Editor*

**Peter Glover**

*News Editor*

**David Creasey**

*Technical Editors*

**Peter Brameld**

**Cliff McKnight**

**Max Parrott**

*Advertisement Manager*

**John Riding**

*Advertising Sales*

**John Snowden**

**Peter Nowell**

*Marketing Manager*

**Linda Dobson**

Tel: 061-456 8383 (Editorial)

061-456 8500 (Advertising)

Telex: 667664 SHARET G

**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 (IR £16)

£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  
Wells, Gardner, Darton & Co Ltd, Fay-  
gate, Horsham, West Sussex RH12 4SU.  
Tel: Faygate 444.

Writing for Windfall: Articles and pro-  
grams 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. Unless  
otherwise agreed, material is accepted on  
an all rights basis.

© 1983 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 Com-  
puter is not responsible for any of the articles in  
this magazine, nor for any of the opinions  
expressed.

# LISTING

17	<b>WHAT'S NEWS</b> Inside look at the Apple world
21	<b>APPLE '83</b> Big show brings big orders
25	<b>THINK TANK</b> Apple Pascal and Fortran directory
28	<b>APPLETIPS</b> Make programming easier
31	<b>GAMESMANSHIP</b> Wouldn't you rather be Zorking?
35	<b>PROGRAMMING</b> Directing output to the printer
38	<b>MICRO PLANNER</b> Using critical path analysis
40	<b>VISICALC</b> Spreadsheet manual makes heavy going
44	<b>FEEDBACK</b> Hang on to your variables
47	<b>BEGINNERS PLEASE</b> PEEKs, POKEs and CALLs
50	<b>BUSINESS</b> Turnkey makes life simpler
54	<b>CERN</b> Apples in atomic research
56	<b>SPACE SAVER</b> File organisation methods
59	<b>INSURANCE</b> Keeping the competitive edge
61	<b>COMPUCOPIA</b> The latest soft and hardware
69	<b>APPLECART</b> Pilot Animation reviewed
73	<b>DISC DOCTORING</b> A switch will cut costs

# GOT A FUNNY DISK? ... WANT TO KNOW MORE ABOUT IT?

Then you  
need the



(Confidential  
Information  
Advisors)

## CAN YOU...

- \* edit normal or protected disks?
- \* quickly find and recover any intact file, however badly the disk is corrupted?
- \* list programs directly from any disk - protected or not?
- \* examine textfiles directly from any disk - protected or not?
- \* analyse the formatting of normal or protected disks?
- \* decrypt commercial software - or encrypt your own?
- \* rapidly auto-search normal or protected disks for anything you like?
- \* understand & use the latest copy protection methods?
- \* use your Apple as a powerful document retrieval system?
- \* make use of an exhaustive knowledge of disk lore?

**YOU CAN NOW** — with a little help from these 5 sophisticated disk utilities:

**TRICKY DICK** examines, records, deletes, and edits. It can: (1) read individual sectors from normal and most protected disks, (2) list their contents in BASIC, assembler, ASCII, or hex, (3) edit them; (4) write them back to the disk. Tricky Dick cunningly bypasses most protection systems, allowing you to work on disks with nonstandard formatting, half-tracks, and altered DOS marks. It is also a chief executive program that directs the following undercover agents:

**THE LINGUIST** reads in a trackful of raw data for your scrutiny, translates all the address information, and allows you to inspect the track's formatting. It also translates all 3 types of DOS encoding (6 & 2, 5 & 3, 4 & 4), and works with Tricky Dick to list and examine programs or textfiles on any protected disk. You can use The Linguist to recover valuable files from blown disks, improve your programming skills by studying commercial software, and analyse standard or altered formatting.

**THE TRACER** rapidly searches normal and most protected disks for up to six strings of your choice simultaneously (specified in ASCII or hex). The Tracer also verifies disk formatting, and sniffs out all hidden catalog or VTOC sectors. When it finds something, it transfers control to Tricky Dick and puts the cursor over the object of your search. A few further keystrokes allow you to make any necessary changes and write the sector back to the disk.

**THE CODE BREAKER** keeps your programs and textfiles from prying eyes by enabling you to translate them into a "secret code" during disk storage. This utility also deciphers encrypted

commercial programs, allowing you to use Tricky Dick to read, list, and edit software never before accessible to any disk utility.

**THE TRACKER** closely shadows the disk drive arm, carefully recording all its movements and operations. The Tracker's job is to display, on either your screen or printer, a list of every track and sector accessed during a LOAD, RUN, SAVE, or any other DOS operation. This utility also tells you exactly where a read or write occurred during any disk access. Use The Tracker's services to locate the precise trouble spots on a clobbered disk, to determine sector skew patterns, to discover the location of hidden "nibble-count" tracks on protected disks, and to learn much more about how DOS works. You'll be surprised to see just exactly where the disk arm really does go!

What's more, you get permanent access to:

**THE CIA FILES**, a 50,000+ word book designed to turn you into a disk expert. In addition to complete instructions for the 5 CIA utilities, the book contains an easy-to-follow hand-holding tutorial (written in plain English!) on all aspects of the Apple disk. Using the CIA utilities as your personal guides, you progress step-by-step to total disk mastery. You'll acquire a wealth of skills and information relating to disk repair and file recovery, DOS patches, copy protection, disk formatting, program encryption, and other vital topics. Much of the material has never before appeared in print.

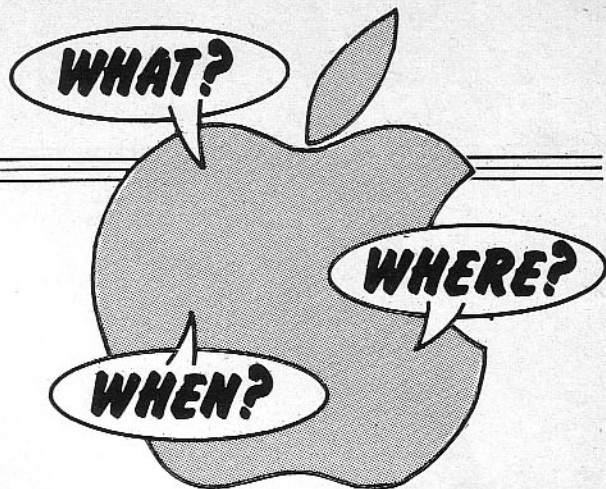
All programs are UNPROTECTED, and hence can be copied, listed, and modified at will. (special patches are described in the manual). They require one drive, DOS 3.3, and 48K of RAM.

TO GET THE CIA ON THE TRAIL OF YOUR DISKS, SEND £35.00 TO:

**GOLDEN DELICIOUS SOFTWARE LTD. Penthouse, 7, Sloane Ave., London, SW3 3JD**  
Dealers enquiries invited

# WHAT'S NEWS...

By David Creasey



## Copyright ruling stiffens the fight against pirates

APPLE'S fight against lookalike and work-alike machines received a major lift recently with a ruling by a Californian judge that ROM-based software programs should be protected by copyright laws.

It is an important judgement which could have a major bearing on the ability of software companies to fight back against so-called pirates.

In the case Apple Computer was awarded a preliminary injunction against a Los Angeles distributor of the Pineapple – an Apple II lookalike machine imported from the Far East in kit form.

Judge Irving Hill forbade distributors Formula International from copying or selling any of Apple's copyrighted software programs in the computer's memory circuits or discs. He specifically included the autostart ROM, Applesoft, DOS 3.3 and Integer Basic programs.

Judge Hill concluded that US copyright statutes cover programs stored in a computer's ROM components whether in source or object code. He said Congress intended to include operating system programs such as Apple's under the protection granted by copyright laws.

"Apple seek here not to protect an idea, but rather one's particular expression of that idea in the form of a specific program," he said.

Meanwhile two supposedly Apple work-alike machines have arrived on the UK market and Apple's legal beagles are no doubt hard at work to establish whether they contravene copyrights.

They are the Basis 108 from BCD Systems and the Franklin Ace series which has been taken on by CK Computers.



*Peter Cobb (left), managing director of Apple Computer (UK), presenting an Apple II computer system and £250 worth of books and software to Keiran Walshe, the first Young Apple Programmer of the Year*

## 1983's Young Programmer

THE first Apple Young Programmer of the Year award is to go to Keiran Walshe, a 19-year-old computer science student at Manchester University.

His winning program simulates the running of a retail business using an Apple. It was designed to give young offenders at a retraining centre a sense of responsibility.

The competition was sponsored by Apple UK, John Wiley and Sons and Sigma Technical Press. It was held to provide a sounding board for talented, young software writers and to stimulate general interest in the technology and applications of computers.

The judges were looking for original programs with an educational theme incorporating something unusual that would interest young people.

Keiran's program was written when he was 18 for the London Borough of Hillingdon Social Services Department.

Six to eight "players" participate, taking on management, staff or customer roles. They study the effect on a retail business of various external factors and

internal decisions. Problems such as stock shortages and strikes have to be negotiated and players score points in relation to the economic performance of the shop.

The competition judges said that the program instilled a sense of role playing, and they praised its "interesting and thorough use of Apple II colour graphics."

Keiran says he will use his prize, a complete Apple system with a colour monitor, to write software for physically and mentally handicapped people.

## On view to 10 million

APPLE showed Lisa to about 10 million people on BBC Television's "Tomorrow's World" programme.

The micro wasn't actually identified and didn't spend more than a few minutes on air – naturally, the BBC didn't want to show bias or commercial interest by mentioning the name of a single company – but apparently the BBC switchboard received many calls from people asking "what was the name of that machine?"

## Sales rise by 74 per cent

APPLE'S remarkable success story keeps on hitting the headlines. The company recently reported a 74 per cent increase in net sales and a 73 per cent increase in net income for the second quarter of 1983, compared to the same period a year ago.

Worldwide net sales reached a record \$228 million from \$131 million a year ago, and net income increased to \$23.9 million, compared to \$13.8 million.

Total sales for the first half of 1983 were \$442.3 million.

## Agatha – the Russian Apple

WILL Apple Inc take on the Russians? The company has already pledged to fight to the bitter end the manufacturers and importers of lookalike and software compatible machines threatening both its business and its reputation for quality.

Now we've heard that the Soviet Union has developed its first personal computer, nicknamed Agatha – and the trusty Apple II served as a development prototype.

According to *International Business Week* the first 1,000 units are to be installed in schools in Moscow and Novosibirsk, a major computer training centre, in September.

Each classroom will have 18 colour terminals with two students at each machine, and current plans involve the installation of Agathas in 5,000 classrooms by the end of the decade.

The Agathas are being made at Zelenograd, near Moscow.

## Two-year lead by Lisa

WHEN Lisa was first announced early this year some people thought that the VisiOn operating system, being released this summer on the IBM, was the same thing.

However, Brian Reynolds, product manager for Lisa in the UK, recently had his first look at VisiOn and told us: "What VisiOn has now is exactly what we had on Lisa two years ago. It has taken us two years to reach our present state of development."

Peter Cobb, Apple UK managing director, said that VisiOn and Lisa are two completely different concepts. While Lisa

was revolutionary in that it employed graphics and the mouse, giving a picture of a filing cabinet for example, VisiOn fashioned a mouse linked to an old fashioned concept of text.

"The techniques are very different, and the more the difference is blurred in the public eye the more VisiOn will be seen as a potential rival to Lisa," he said.

Meanwhile John Ash, managing director of Asolv and immediate past chairman of the UK Pascal Users Group, told delegates to the Apple '83 convention last month that VisiOn offered users of conventional micros the first step on the road towards a Lisa capability. It used the main concepts of existing, conventional micros, which were limited to handling only 80 columns on screen.

"You can't get something for nothing," he said. "You can't expect a conventional screen which handles only 80 columns to cope with Lisa quality graphics.

"VisiOn offers a software environment for existing micros. As they improve, so will the software be upgraded."

Ash also noted that the mechanical mouse, used as a control unit in Lisa and VisiOn, is multiplying. There were already 11 "mouse houses" he said, and he predicted that within six to eight months every micro released will use a mouse device.

## Medi-micro on duty

CLIPBOARDS and charts at the foot of hospital patients' beds could start disappearing as word gets around about what they do in St Paul, Minnesota.

The intensive care unit at a children's hospital there now has 10 Apple computers – one for each bed, one to monitor respiratory functions and one on the duty nurse's desk.

The bedside Apples keep a constant check on vital matters like blood pressure, temperature and pulse rate.

The hospital first looked at large mainframe systems specially designed for intensive care wards. But they found that not only were they too expensive but would mean that if the computer was down the whole of the ward would be down.

Said paediatrician Dr John Tilelli: "We wanted a simple design, software that was easy to write and a system that would make it easy to communicate between the various monitoring instruments and the computer."

The Apple, they found, would give them all they wanted and more for almost a third of the cost of a mainframe.

But Dr Tilelli hasn't stopped finding

uses for the Apple. He is a specialist in echo-cardiographs, which use sound waves to produce an image of the heart, and is developing a method of converting output from the machine into a picture on the Apple's monitor.

A joystick can be used to measure sections of the heart, working out which parts are not functioning normally and allows treatment to be started much earlier than is usually possible.

In fact the Apple's bedside role could expand considerably, claims Dr Tilelli. "It should be possible to automate treatment", he said. "For example, the computer could sense a 10 per cent change in blood pressure and make the necessary adjustment in treatment.

"I can even see the day when the Apple would supervise the majority of monitoring and therapeutic tasks."

## Visicalc marches on

THE popularity of the Visicalc program continues unabated. The package recently won a special award from International Computer Programs in recognition of topping the \$25 million dollar mark in sales – only a year after IPC singled it out for passing the \$10 million sales mark.

The award was made to Software Arts as creator of the Visicalc program and to Visicorp, which markets it.

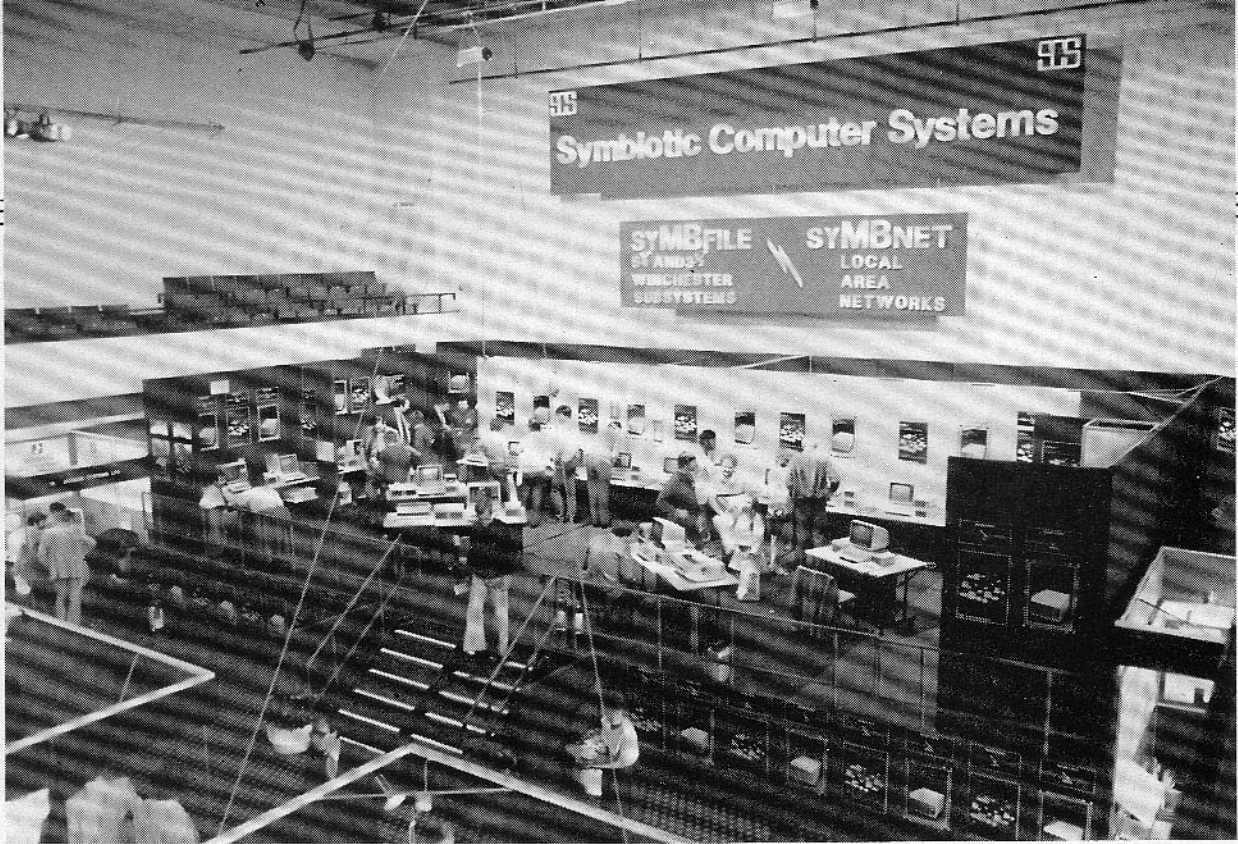
IPC introduced its Million Dollar Awards ceremony in 1972 to honour the producers of software products that had achieved sales of over a million. Since then new award levels have been added to reflect the growing market for software.

Meanwhile another American company, Softalk Publishing, says the Visicalc Advanced Version was the most popular program for the Apple III in 1982. A poll of Softalk readers showed that Apple III users were almost unanimous in the selection of Visicalc.

## Apple's new Euroboss

APPLE gets a new European boss this month. He is Michael Spindler, whose promotion from European marketing manager and director of marketing to the post of general manager takes effect on July 4.

Spindler succeeds Ken Zerbe, who moves up to become Apple's vice president and general manager for inter-continental operations. His area covers the world outside North America and Europe.



# Apple '83 -the big show was brighter and better than ever

LAST month's Apple '83 was billed as bigger and better than Apple '82 – and so it proved. The three day get-together at Slough was an outstanding success from whatever angle it was viewed.

Exhibitors in particular were delighted with what they called the "quality" of the visitors. Everyone there had an interest in Apples, and where a bargain was to be had, seemed ready to spend money.

Most of the standholders did a lot of business on the spot while others said they were extremely happy with the level of enquiries.

"We received more than £20,000 worth of orders", said a spokesman for U-Microcomputers, "and we've since been getting orders in over the telephone that can be directly attributed to the show.

"We also met a lot of dealers. For us the show was even better than last year."

Symbiotic Computer Systems took a multi-thousand pound order within an hour of the show opening and managing director Rodney Cox commented: "In the three months following Apple '82 we sold around £250,000 worth of equipment as a direct result of the show.

"With Apple '83 it wasn't a case of generating the business – people actually placed orders on the spot." He said his company took more than 50 orders and also appointed 20 new dealers and a French distributor.

Symbiotics used fibre optic cable to network the 24 Apples on its own stand with those on 11 other stands, enabling the whole exhibition to be linked together electronically.

From a visitor's point of view there was something to please everyone – from the free sticks of "apple" rock handed out by Pete and Pam to a wealth of information about present and future trends in networking, a huge variety of software and books and the chance to share experiences and to find answers to problems from the assembled Apple experts.

For many people it was also the first



chance they'd had to see the new Lisa in action — with Apple UK running a continuous half-hour demonstration throughout each day to a packed audience.

*Windfall's* own technical stand did a roaring "trade" answering readers' queries. Visitors liked having someone independent to turn to for advice and problem solving, and enjoyed meeting our writers in person.

At any exhibition one expects to find many products from independent manufacturers. On this occasion the additions weren't limited to peripheral devices. BCD Systems, for example, were showing their dual processor Basis 108, a CP/M machine which is also claimed to run most Apple software.

The Apple users' convention, which formed part of Apple '83, also had a considerable spin-off value for exhibitors. The benefit of having experts who know their subject and can make unbiased recommendations was illustrated by the very rapid disappearance of a particular Visicalc book from the shelves of Haigh

and Hochland's book stand minutes after a convention lecture on Visicalc had ended. Twenty-seven copies of the recommended volume were sold.

This demonstrated not only the effect of the convention but also of the well stocked nature of the exhibition stands — no one could anticipate the demands the convention delegates would make on their stocks.

Standholders, who arranged numerous dealerships and distributorships, said they liked the fact that it was a friendly show, that it was Apple-specific, and that the quality of people attending was so good. Everyone we spoke to said they would come back next year.

Even Apple UK, with a hardened team of exhibition staff, was impressed. "The exhibition provided a very good platform from which to learn and was an ideal venue for the exchange of ideas and for communicating with Apple dealers, software developers and others," said a spokesman.

"It gave us an excellent opportunity to find out more about Apple users and to

discuss their requirements first hand."

Data Efficiency said the show had "certainly been well worthwhile" and that it could have sold Apple systems off the stand had it thought to do so in advance. However a spokesman said that there weren't as many actual dealers attending as he had hoped.

In direct contrast was DMS Electronics whose managing director, Steve Alsop said he had arranged distributors for his colour cards, bar code readers and Presfax key pad in Italy, Sweden, Austria, Australia and Saudi Arabia.

"It was a very worthwhile show," he added, "and if orders come in as we expect we should move between 10 and 15 thousand units here and abroad.

"We didn't have time to eat, drink or even see friends," he added.


Nick Justice of Strategic Systems said the exhibition gave his stand, and product, just the right sort of exposure.

"It wasn't a roaring business, but that wasn't my objective in taking the stand. We wanted to meet distributors — and did so from Sweden, Switzerland, Germany, France and North America. Dealers there were looking for new products."

It was the first time at any exhibition for Number One Computers of Jersey. Managing director Terry Crouch commented: "It is fair to say that we got out of it everything that we wanted to achieve.

"We learnt about the need to sort out customs documentation (our car was impounded), the need to pack a stand high and sell cheap if you are to succeed, and many other things that we learnt by doing. We will almost certainly come back next year."

Pete and Pam, one of the largest importers and distributors of Apple software and hardware, doubled its stand space compared to Apple '82.

A spokesman said the overall standard of the show was even better than it had been a year earlier. He added: "The quality of visitors was better, it was more professional, and even the exhibitors were more concerned with their image." 



# Wide ranging convention had a business slant

THE Apple '83 Users' Convention was a great success. Speakers covered a wide range of subjects with something for everyone, but with a particular slant towards business, and there was a good attendance over the full three-day programme.

The main function of the convention, apart from drawing together a pool of experience for users to dip into, was to present the computer not as a miracle of modern technology, but as simple to use.

Examples of what the hardware could achieve in business were given by a variety of speakers ranging from Dave King of Apple UK – who should know something about Apples – to Malcolm Harby and Tony Pearce who had each independently chosen the Apple to perform a range of business functions, and who explained how they had achieved satisfactory results.

The three men were excellent examples of the high standard achieved by the convention speakers.

The spreadsheet session started with an introduction to the superb aid, Visicalc. Fran Teo illustrated the relationship between the commonly encountered financial schedule and the electronic representation provided on screen by Visicalc. The smoothness of her presentation gave no hint of the dramatic events to follow.

Nick Levy, who has built an excellent reputation with his Visicalc articles in *Windfall*, followed Fran with a dynamic demonstration of "destructive" Visicalc techniques.

His particular approach was applied first to the microphone and then, with even greater impact, to the on-stage Apple system itself! Ten minutes into his presentation he trod on the microphone lead and the mike fell to the ground and broke.

En route to the second microphone (with chairman Brameld on his knees in front of the guru of Visicalc, trying to mend the equipment) Nick tripped over the Apple mains lead.

The lecture was now without sound or an Apple screen display, and to make matters worse, the final visual aid, an Apple poster, became unstuck and slid off the lectern.

Nick carried on with his presentation apparently undaunted while the equipment was restored (whereas most people would have been rooted to the spot.) All credit to those prepared to live dangerously.

Steve Mandelin was faced with the difficult task of following Nick and rose to the occasion by outlining many of the new facilities available on the latest version of Visicalc known as Advanced VisiCalc. Also covered were MultiPlan and MagiCalc, sometimes known as The Spreadsheet.

There was considerable interest in Mike Glover's presentation on software utilities and Peter Gorry's use of the Apple

in laboratory interfacing. Both sessions took on a workshop atmosphere – there was plenty of audience interaction and considerable demand for "more of the same" next year.

So too with the lecture on portable software – John Ash's stimulating look ahead at software for the '80s.

Roger Stemplis made the long journey down from Hamilton, Scotland, to tell us of his school's fairly recent experience with networked Apples, and how they raised money to pay for them.

His talk was well contrasted by Dave Brunner of London who has been using Apples in school for some years. He generated considerable interest in his description of the software library his school accesses by modem.

A possible way forward with programming languages in education was suggested by Janet Rothwell, an impressive speaker who in turn was impressed with Superpilot and Prolog on the Apple.

Paul Rayner gave a thorough account of what a micro-based database package is and should do. It was enlightening for

businessmen in the audience new to the concept and entertaining, if general, for the more experienced users.

As Paul explained his talk was based on the nightgown principle. "It covered everything lightly, highlighted certain areas and left you wanting to know more."

Networking featured on many of the exhibition stands at the related Apple '83 show. Paul Wright's lecture on the subject gave a timely summary of its development and a look at where it is likely to fit into various levels of business in the future.

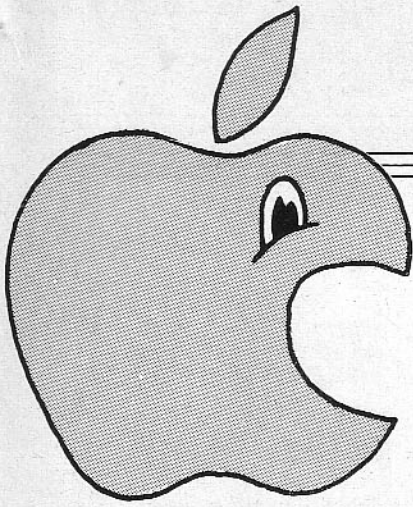
The general reaction from delegates was that they had been served a varied and excellent fare.

Some asked that next year we make the lecture sessions longer (more time for questions), publicise them earlier and have more on the scientific, hardware and education side.

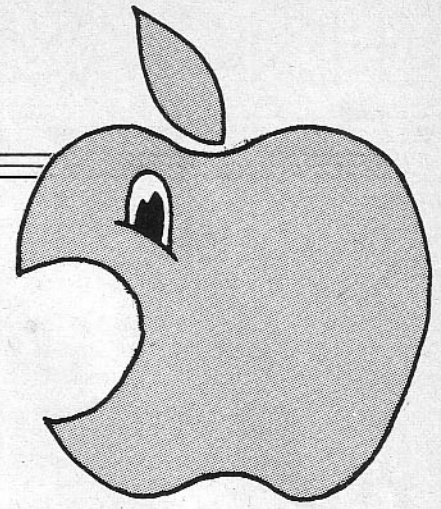
As it was a users' convention, we would be very interested in readers' comments. Please let us know what you would like to see and hear at Apple '84!







# THINK TANK



... the *Windfall* platform for anyone wishing to agree with, improve, disprove or generally discuss specific articles in *Windfall*. Write to: Think Tank, Windfall, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

## Apple Pascal and Fortran Directory format

WHILE using the Pascal language system I tried to update the work file with my current text, only to find that the Directory had crashed. Despite backup, if you are currently working on a program, losing the SYSTEM.WRK.TEXT can be disastrous.

Being the simple minded sort (the information on disc was too valuable to lose), I decided to reconstruct the Directory, and found that only the first sector had been scrambled. I re-initialised the sector, and started hunting through manuals and magazines to find out what it should contain besides the zeros it now had.

After a few hours searching I came to the conclusion that *technical* information was totally missing from Pascal manuals and that, though there is plenty of literature about DOS, I couldn't find any-

By T.N. THOMPSON

thing about the nuts and bolts of Pascal.

To rectify that, and to assist anybody with a similar problem, either with Pascal or Fortran, I have found that the following format will get the disc running and recover your files. However you must know the start and end blocks of your files on the disc, or be able to make a good guess.

Text files are easy enough to find: the first four sectors (two blocks) are mainly empty, being used purely for the information of the Editor; the first sector of Code files contain a heading section from byte \$40 (hex) to byte \$BF (hex); Data files are much more difficult to find, as they merely start with data.

On my copy of Pascal the Directory starts at Track \$00, Sector \$04, and ends at Track \$00, Sector \$0B.

Regardless of where it starts the format is the same, and is listed in Figure 1.

When using the standard 40 column display the number of bytes used in the last block is suppressed. This figure is displayed in an Extended Directory listing, when the screen width has been adjusted to 80 columns by means of the APPLE3:SETUP.CODE program. This should only be done if you are using an 80 column card or an external terminal with 80 columns width.

The two date bytes are a beautiful example of data compacting. They are formed as follows:

The month is the low nibble of the first byte. The last two figures of the year ('82' from '1982') are multiplied by two, giving

an even number which forms the basis of the second byte. (An even number has a zero in bit zero.) The day of the month is multiplied by 16 and forms the high nibble of the first byte.

Obviously this only allows for 15 days. If there are more, then one is added to the second byte and the number of days, multiplied by 16 MOD 256, is stored as the high nibble of the first byte.

It then follows that:

```
(18 A0) is 01-AUG-80
(FB A0) is 15-AUG-80
(18 A1) is 17-AUG-80
(FB A1) is 31-AUG-80
```

This sounds very complicated in normal language, but in assembly language it is very easy. (See Figure II.)

Once the sector has been re-initialised the following entries are vital: the length of the volume name and the volume name, the number of Directory entries, and the Directory entries. When accessing

Byte (hex)	Use
0-1	Not used (normally \$00 00 00)
2	Normally \$06 (changing doesn't seem to have any effect)
3-5	Not used (normally \$00 00 00)
6	Length of Volume name
7-0	Volume Name
E-F	Number of blocks on disk (Lo/Hi)
10	Number of entries in Directory
11-13	Not used (normally \$00 00 00)
14-15	current Volume Date
1A	Byte 00 of first Directory entry

All subsequent Directory entries are of \$1A bytes in length, and take the following format:

Relative Byte (hex)	use
0-1	Start Block number (Lo/Hi)
2-3	Block following program (Lo/Hi)
4	File type
5	Not used (normally \$00)
6	Length of Filename
7-15	Filename
16-17	Bytes used in last Block (Lo/Hi) see below
18-19	Date stamp for entry. See below

File types are text = 03, Code = 02, Data = 05.

Figure I

```

LDA YEAR
ASL          ;Multiply by 2;
STA DATE HIGH ;And store it

LDA DAY
CLC          ;Set the carry to zero
ASL          ;Moves the day into the
ASL          ;high 4 bits (multiply by 15)
ASL          ;if the day was greater than
ASL          ;15 the carry is set

;
BCC ADDMONTH ;if carry clear continue

;
INC DATEHIGH ;if not add one to year
CLC          ;and set carry to zero

;
ADDMONTH ADC MONTH ;Add month to day
STA DATELOW  ;and store it

```

Figure II

# THINK TANK

the Directory Pascal looks for the information you require for the number of entries at byte \$10. You can have as many as you like beyond that number but Pascal will ignore them.

After the first section of the Directory has been entered you must construct the file entries on the format given. The date is not essential, but the prologue is. To delete a file, merely enter 00 at relative byte \$06 (length of file name) and Pascal will delete it during the next pass.

Using this format I was able to recover 90 per cent of the files, which is reasonable, considering I started with nothing.

I should hope that the complete loss of the Directory doesn't happen to too many people. The most likely problem to be encountered is the accidental deletion of a file. Provided the disc hasn't been K)runched after the deletion, the best way to recover the deleted file is by making

use of the M)ake command. Repeated use of this should, eventually, leave you with the file recovered.

When the M)ake command is used, Pascal automatically reconstructs the Directory in order to fit the new files in the correct places. If you are M)aking a file that takes the first block, the whole Directory will be renewed.

The only time this will not work is when two deleted files occupied adjacent blocks on the disc. In which case, it's back to the byte editor.

Having gone through this detective work for Pascal and Fortran, I decided to check that I had the information to recover a CP/M disc if one crashed. I didn't, so, while in the mood, I set to and found it a lot easier to understand.

The CP/M disc is set out in 1k blocks, not 128byte records as STAT would have you believe. The first three tracks (00-02)

are reserved for the system. From there on the tracks (\$03-\$22) are split up into four blocks, each 1k long. These blocks are numbered consecutively working up from track 03 sectors 00-04 which is block 00.

Blocks 00 and 01 are reserved and contain the directory, giving sufficient space for 32 entries. The numbering of these blocks is important as CP/M uses them instead of track/sector lists to find the needed information. Free directory space following the last entry is filled with \$E5.

Each directory entry is \$20 bytes long, and a complete breakdown is given in figure 1. As can be seen, there is only sufficient space in each entry for 16 block numbers, giving a file length of 16k. For longer files, extra file entries are made with *exactly* the same file name and type but with an extent number one higher than the previous extent.

## Timing recursive routines

WHEN making programs containing recursive routines, it is important to know how fast a particular part of the program is, writes **M. Keesen** of Enschede, Holland. Especially when a routine is used a lot of times this is interesting.

A way to measure the time is to execute such a routine for say 1000 times

and check the time with your watch. This is, of course, not a very accurate method.

I found a nice way to measure the time which it takes to execute one or more statements. For this program I use the annunciator outputs of the game I/O connector. They are set and cleared by extra statements which you place in the

program under test. The time intervals between these statements are measured with an oscilloscope.

To set and clear the annunciator outputs takes some time which is different for the applied language. The following examples each generate a pulse on annunciator output number zero:

APPLESOFT BASIC :

```
10 POKE 49240,0
20 POKE 49241,0 (6.4 mS)
30 POKE 49240,0
40 GOTO 10
```

```
10 POKE 49240,0 : POKE 49241,0 : POKE 49240,0 : GOTO 10 (6.2 mS)
```

```
10 A = 49240 : B = 49241 : C = 0
20 POKE A,C
30 POKE B,C (1.6 mS)
40 POKE A,C
50 GOTO 20
```

```
10 A = 49240 : B = 49241 : C = 0
20 POKE A,C : POKE B,C : POKE A,C : GOTO 20 (1.5 mS)
```

```
10 A% = -16296 : B% = -16295 : C% = 0
20 POKE A%,C%
30 POKE B%,C% (2.2 mS)
40 POKE A%,C%
50 GOTO 20
```

```
10 A% = -16296 : B% = -16295 : C% = 0
20 POKE A%,C% : POKE B%,C% : POKE A%,C% : GOTO 20 (2.1 mS)
```

INTECER BASIC :

```
10 POKE -16296,0
20 POKE -16295,0 (0.92 mS)
30 POKE -16296,0
40 GOTO 10
```

```
10 POKE -16296,0 : POKE -16295,0 : POKE -16296,0 : GOTO 10 (0.89 mS)
```

```
10 A = -16296 : B = -16295 : C = 0
20 POKE A,C
30 POKE B,C (1.2 mS)
40 POKE A,C
50 GOTO 20
```

```
10 A = -16296 : B = -16295 : C = 0
20 POKE A,C : POKE B,C : POKE A,C : GOTO 20 (0.96 mS)
```

MACHINE LANGUAGE :

```
LABEL : LDA $C058
LDA $C059 (9.6 uS)
LDA $C058
JMP LABEL
```

```
LABEL : STA $C058
STA $C059 (9.6 uS)
STA $C058
JMP LABEL
```



## LONG WAIT

**t** The Apple Reference Manual says that the only way out of a WAIT statement, apart from the conditions being fulfilled, is to press RESET. This can cause a bug in some cases, for example:

```
100 FLASH
105 PRINT "PRESS ANY KEY TO CONTINUE"
110 POKE - 16368,0
115 WAIT - 16384,128
120 REM REST OF PROGRAM
```

If RESET is pressed to terminate this WAIT then a bug is introduced into the edit mode, so that if you now try to edit a line of program, it will overprint it with garbage. (in fact it adds 32 to the original character-code.)

This seems to happen with all WAIT statements that follow a FLASH statement. The simple solution is not to press RESET or do not use the flashing mode with a wait instruction.

**John J. Grieve**

● Well done. A very annoying fault well spotted. In fact this particular phenomenon manifests itself on many occasions, not only when one is using the WAIT command.

If you have been doing fancy, flashy or inverse presentations on the screen just before a program failing due to the ubiquitous SYNTAX ERROR etc. then always type NORMAL before attempting any editing. This will ensure that the editor works as expected and does not produce garbage on the screen.

**Peter Brameld**

## SHIFT LOCK SNAG

**t** A very small percentage of Apple II software won't run on the IIe. However, several games appear to run OK (insofar as they boot correctly) but there is no response to movement commands input from the IIe keyboard.

Before giving up in disgust and cursing the apparent waste of money on a game, check that the IIe shift lock is engaged. I've run several games successfully and then suddenly had an apparent keyboard failure.

This was caused by my accidentally dis-engaging the shift lock.

**David Creasey**

## Smarten up your programs

```
***** SETUP *****
5 HOME : PRINT "THIS PROGRAM HEL
  PS IN LISTING YOUR "
6 PRINT "PROGRAMS IN INTELLIGENT
  SECTIONS ON THE"
7 PRINT "PRINTER."
8 PRINT : PRINT
10 INPUT "HOW MANY SECTIONS HAS
  YOUR PROGRAM? ";N
15 DIM A$(N),B$(N)
20 FOR J = 1 TO N
30 PRINT "NAME OF SECTION ";J;"
  ? ";
40 INPUT A$(J)
50 PRINT "POSITION OF SECTION ";
  J;" E.G. 100-200 ";
60 INPUT B$(J)
70 NEXT J
80 S$ = " ***** "
```

```
***** DISC HANDLER *****
90 PRINT "OPEN LAYOUT"
100 PRINT "DELETE LAYOUT"
110 PRINT "OPEN LAYOUT"
120 PRINT "WRITE LAYOUT"
150 FOR J = 1 TO N
160 PRINT "PRINT" + CHR$(34) +
  S$ + A$(J) + S$ + CHR$(34)
170 PRINT "LIST";B$(J)
180 NEXT J
185 PRINT "PR#0"
190 PRINT "CLOSE LAYOUT"
```

**t** HERE is a short program which will allow you to produce a smart looking final copy of your Basic programs, neatly split into headed sections.

When running, it asks you how many sections you want your program to be in, and then requests a title and range for each section. Note particularly that the order you put the sections in does not have to be the order they appear in the program.

When it has all the data, the program creates an EXEC file called LAYOUT containing these titles, and suitable LIST instructions to send to Applesoft, so that when you load your program, set up your printer, and EXEC LAYOUT, you get a listing styled like the one below.

NB: The strings printed in lines 90 to 120, and 190 all start with a hidden CTRL-D.

**J.P. Lewis**

```
***** INSTRUCTION *****
200 HOME
205 PRINT "LOAD THE REQUISITE PR
  OGRAM,"
210 PRINT "SET THE PRINTER TO YO
  UR FAVOURITE OUTPUT"
220 PRINT "THEN TYPE IN 'PR#1 '"
230 PRINT "AND FINALLY 'EXEC LAY
  OUT'"
240 END
```

## Error resolved

**t** C.J. Palmer (Windfall, May 1983) asks about the solution to the NOT DIRECT command error when attempting to re-enter a program in Basic using GOTO.

If the error is caused by the program's use of disc I/O (DOS commands being illegal in DIRECT mode), then the magic POKE is: POKE 51,6 : GOTO xxx (NB all on one line).

**Michael King**

## WAY OUT OF THAT SNOWSTORM

**t** As a follow on from the very good Appletip in the December edition of Windfall on the "Unexpected bonus" from the use of the new Microvitec RGB monitors, the one problem of using the POKEs

provided (and incidentally I prefer POKE - 16143,0 : POKE - 16135,15 : POKE - 16134,15) is that during the loading of the program you are blinded by the white on white effect given by the above mentioned POKEs.

To get round this unpleasant effect try the routine on the right as a preloader to your main program, or incorporate it into an existing program as a subroutine before loading the main program.

Peter F. Wilson

```
10 POKE - 16297,0: POKE - 1630
  2,0: POKE - 16304,0: POKE -
  16144,0
20 POKE - 16143,0: POKE - 1614
  2,0: POKE - 16139,0
30 POKE - 16143,0: POKE - 1613
  5,15: POKE - 16134,15
40 PRINT CHR$(4);"RUN FILENAME
   " or
40 PRINT CHR$(4);"BRUN FILENAM
   E"
```

**t** IF you've ever had the annoying experience of owning a disc which is readable but won't boot properly you may find the routines

```
*9100.912F
9100- A2 8F 8E 29 91 A2 0F 8E
9108- 25 91 A9 91 A0 20 20 D9
9110- 03 CE 29 91 CE 25 91 10
9118- F1 60 00 01 EF D8 FE FE
9120- 01 60 01 00 11 0F 1A 91
9128- 00 7F 00 00 01 03 FE 60
```

Figure 1

```
10 PRINT "BLOAD TRACK-COPIER"
15 RWTS = 37120: REM #9100
20 FOR K = 0 TO 2
25 POKE RWTS + 36,K: REM TRACK
30 POKE RWTS + 34,1: REM DRIVE
35 POKE RWTS + 44,1: REM READ
40 CALL RWTS
45 POKE RWTS + 34,2: REM DRIVE
50 POKE RWTS + 44,2: REM WRITE
55 CALL RWTS
60 NEXT K
65 END
```

Figure 2

below rather useful. The machine code shown as a hex dump in Figure 1 contains a copy of the RWTS tables described in the DOS manual, and a short loop that uses these tables with a call to RWTS, namely \$3D9, to copy a track at a time to \$8000-\$8FFF.

The Applesoft in Figure 2 is the absolutely bare outline of a routine to make it easy to copy the first three tracks from drive 1 to drive 2. This is usually sufficient to revive a non-booting disc if it is suffering only from lost data and not media corruption.

J.P. Lewis

## What boots it to worry?

## COPY STOPPER

**t** Many people wish to stop others copying discs. This method, while not perfect protection, does slow down the process:

Type "POKE 44033,16"  
Put a new disc into your drive and INIT it.

What you have done is change the track of your CATALOG from 17 (normal track) to 16. Other numbers, 3-15, will work too when POKEd into 44033.

To transfer programs from a normal disc to your new disc:

Boot a normal disc  
Load a program from that disc  
Type "POKE 44033,16" (assuming you used 16)  
Insert your protected disc  
Save the program  
Type "POKE 44033,17"  
(normal CATALOG track)  
Insert your normal disc and go to step two.

Allan Dubost

# GAMESMANSHIP

If you're a bumper-sticker freak, you'll already know that divers do it deeper, wind-surfers do it standing up and anglers tell lies about how big it is. Did you know, though, that Zorkers do it under the rug?

In these days of adventure games replete with hi-res graphics and animation, Zork is a phenomenon in a class of its own. It is a text-only adventure game interesting enough to have achieved something approaching cult status in America. After a few minutes of playing it is not hard to see why, because the game is extremely well written and not a little difficult.

In fact, Zork is not a single game but three separate ones – or a trilogy, you might say. There is some continuity between them but you can do any without knowledge of the others. You might not appreciate why something was described as "strangely familiar" but success in the game would be unaffected.

Zork I is "The Great Underground Empire" in which you are striving to discover the Twenty Treasures of Zork and escape with them and your life. As the name suggests, most of the action takes place underground as you search for the treasures. Finding them is not sufficient, you must put them in the trophy case, otherwise a roving thief will lighten your load for you.

There is plenty to explore, and you might even find yourself at the entrance to Hades. Mind you, after working for *Windfall*, Hades seems a pretty tame place. You have to do some fighting too, but most adventurers are used to the cut and thrust of exploring. For the mechanically-minded, there is a great piece of equipment to fiddle with at the dam. Given the position of the machinery, you can probably imagine the result of indiscriminate fiddling.

Zork II, "The Wizard of Frobozz", takes you into new depths of the underground realm. The wizard has an interesting style, including a nasty habit of laughing at you when you are stuck somewhere. There are also some interesting geophysical features, like a glacier in a volcano!

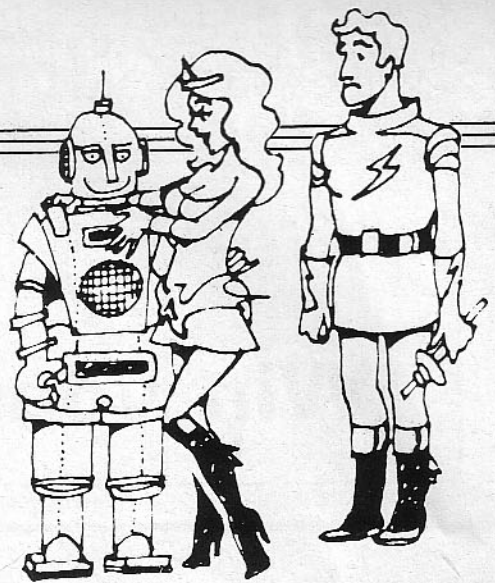
Mythical creatures are well represented. There is a unicorn with strange dietary habits and a dragon which seems to be a close relative of Smaug. The bank is extremely difficult to make withdrawals from, but then I always have that trouble with banks.

Zork III, "The Dungeon Master", is described as the final test of your courage and wisdom, which seems to rule out the possibility of Zork IV. I was delighted to find another piece of machinery to play with, although once again I was less than enthralled with the results of my meddling.

Even death is not straightforward in Zork III. In some cases you are given another chance, but at other times shuffling off the mortal coil results in complete rigor mortis, requiring a reset or reboot.

After the two-word sentences of many adventure games, communicating with Zork may come as a shock. It felt a bit like having been abroad and then coming home where people understand what you're saying. Sentences like "take all but the knife and the lamp" are understood, but common commands like directions can be abbreviated.

While you roam about, the number of



## But wouldn't you rather be Zorking?

moves you have taken is displayed along with your score, hence you can tell when you have found a treasure by keeping an eye on your score. Needless to say, items like the brass lamp which are readily available don't increase the score.

The first time you visit a location a full description is given. On subsequent visits

---

By CLIFF McKNIGHT

---

you are simply given the title of the frame – "forest glade" or "studio". The full description can be recalled using LOOK, or you can opt for VERBOSE, which always gives the full description. If you have a printer you can also print a transcript of the game as it proceeds.

All the usual features are present ... SAVE and RESTORE, INVENTORY, and so forth. There are also commands like AGAIN which effectively repeats your last sentence, and DIAGNOSE which gives you a medical report on your physical condition. As you can imagine, both are useful commands if you're doing any serious fighting.

Without doubt Zork is the best text-based adventure I have ever played. This is presumably a view shared by many people, because there is a Zork Users Group in America. This group is completely independent of Infocom and markets a whole range of Zork products.

They range from maps (which they warn may give hints) and books of Invisiclues (with a special developing pen – read only the ones you want to read) to badges that proclaim "I am a Zork groupie".

Although it is the Zork Users Group, it also offers support for other Infocom games like *Deadline* and *Starcross*. From the literature enclosed with Zork it looks as though you could buy the products from the UK because there is an instruction to add an extra \$2 postage outside the USA and Canada.

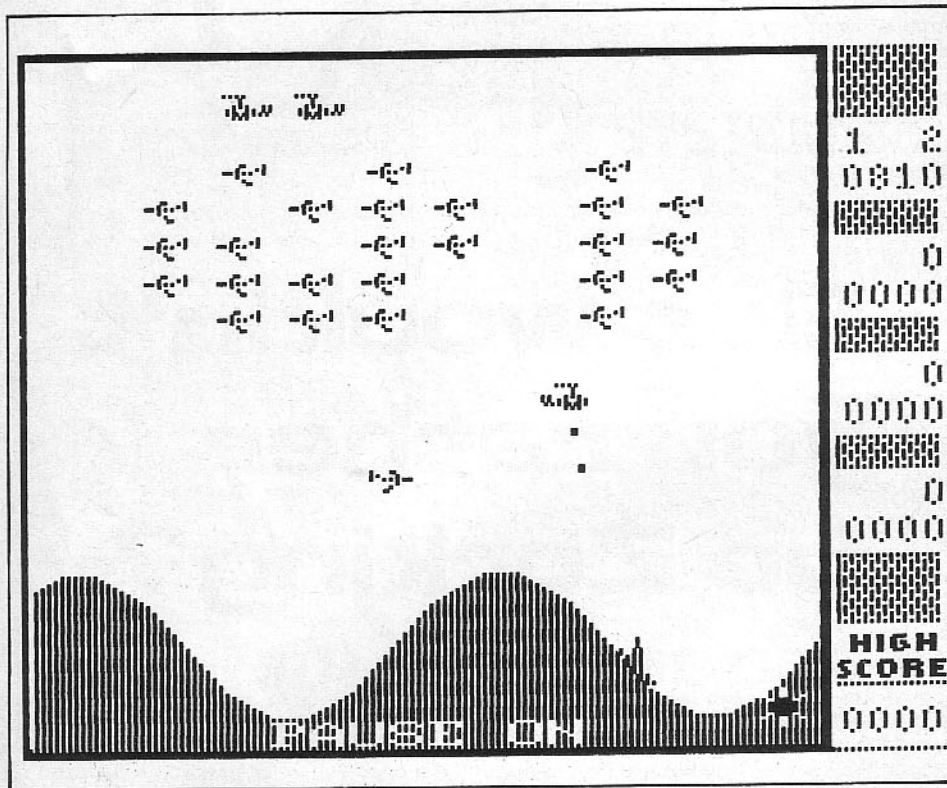
I don't know what it is about Zork, but everyone I know who has played them has enjoyed them. If you haven't played any of them I would recommend starting with Zork I. The later two are self-contained, but you miss out on all the little references which make you feel at home.

Any of the games will last you a long time and all are therefore good value for money. Zork II and III have the added dimension that even your task is not immediately obvious at the outset.

When the world is beginning to weigh heavy on your shoulders, when you're stood waiting for a bus in the pouring rain, when the office bore is telling you why computers can't be as clever as people, just think to yourself "I'd rather be Zorking".

Title: Zork I, II and III  
Author: Not credited  
Publisher: Infocom  
Requirements: Apple II

## Win your stripes, sailor – but try not to get seasick



AVAST and belay, me hearties. Yo ho ho and a bottle of rum, and all that stuff. Well, I mean, with a name like Wavy Navy, it's got to be that or "Hello sailor".

As the name suggests, the context is nautical. You are on a PT boat which is being tossed by 30 foot waves and the sky is full of enemy bombers, Kamikaze fighters, and helicopters firing machine guns... and that's only level 1!

Your task, despite your rank as galley slave, is to blast all these enemy planes out of the sky. Presumably you throw last week's sausages at them! If successful you are promoted to boatswain (or bo's'n, as we sailors with O-level English know it) and move to level 2.

There we find all the old favourites, with the addition of a mine which drifts across the sea. You have to stay at the top of the waves to miss it, thus making

## Tour the (text-only) galaxy

HAVING listened to the radio series, read the books and watched the TV series, I suppose it was inevitable that I should try the adventure game of The Hitch-Hiker's Guide to the Galaxy. I particularly enjoyed the radio implementation because it allowed me to imagine the characters. However, I thought the TV series was quite well done, particularly some of the special effects like Zaphod's second head.

How, then, could an adventure game recreate something as lavish as the TV series? The answer in this case is that it doesn't try. The game is a text-only adventure game and is therefore more like the books or radio shows. The version I saw was an advance copy with no documentation, so I can't say what the packaging will be like. However, Eddie the on-board computer tells you that he thinks the idea is to zoom around collecting various objects. This is more than Arthur was told, so I can't complain.

One of the things I *could* complain about was that some of Eddie's replies were presented so briefly as to be sub-

liminal. However, I quickly learned that a flash of the screen usually meant "right, guys". This may be ironed out in the final version.

If you've seen/heard/read Hitch-Hiker and you've played a text-based adventure game, you should have a good idea of what to expect from this game. If you haven't done the former, I congratulate you on successfully burying your head in the sand for the last few years. If you haven't done the latter, then you might like to try this one.

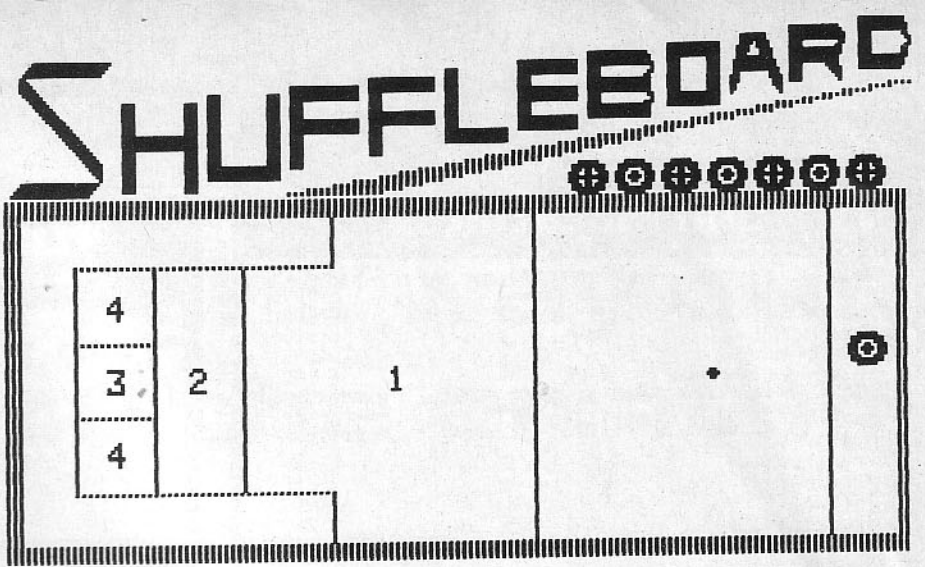
One major way in which the game differs from others is in the style of the interaction with the environment. Rather than being able to Talk, Give, Buy, etc., you mostly have to Drop one of your possessions at the appropriate place. It was surprising to have such a limited vocabulary, particularly in a text game where lots of disc space isn't needed for pretty pictures.

One thing which distinguishes the game is that it comes from a British software house. I'm not one to wave the

flag because I like too many of the American games, but it is nice to see a British product now and again. Estuary are also advertising the game for the 48k Spectrum, but maybe *Windfall* readers don't want to know that.

I was in a health food shop recently, and rather than search the shelves for lettuce-burgers or whatever I asked the assistant if they had any. "No", she replied, "but we sometimes have them. The manager gets them because he's a Vogan." She probably meant 'vegan'... at least I hope she did. On the other hand, there are some pretty awful poets around these days.

Title: *The Hitch-Hiker's Guide to the Galaxy*  
Authors: Bob Chappell and Michael Foy  
Publisher: Estuary Software Products  
Requirements: Apple II



## as well

yourself more vulnerable. Mind you, you stand the chance of being promoted to cook. I knew you'd be impressed.

Being an American game, you have the chance to fight your way up through the ranks and achieve the lofty status of President. Personally, I've never made it higher than deckhand because the Exocet missiles on level 4 usually finish me off.

Up to four people can play Wavy Navy, each being prompted when it is his or her turn. The high score for the current session is displayed, but none are saved to the disc.

Basically, Wavy Navy is a shooting gallery type game. However, as we've come to expect from Sirius Software, it is very well put together. The waves complicate the usual motion across the bottom of the screen, and the enemy doesn't just fly past waiting to be shot down.

I was delighted with the quality of the tunes played at the end of each level and notice that they were created using Electric Duet from Insoft (reviewed in *Windfall*, December 1982). As you might imagine, the tunes have a suitably nautical theme. All action during the game is accompanied by various noises, and all the sounds can be switched to the cassette port (effectively turning them off).

Another good feature is the pause facility. What is unusual about this one is that when you press ESC to restart the game it doesn't restart immediately. There is a short pause with the words "get ready" on the screen in place of "pause on" and this gives you time to sit back from the keyboard with your joystick or paddle.

Wavy Navy is a good game, hard enough and with enough levels to last you a long time, assuming you don't succumb to sea sickness after half an hour of the 30 foot waves. Right, then - I'm off to splice the mainbrace. You can find your own preventative medicine.

*Title: Wavy Navy*  
*Author: Rodney McAuley*  
*Publisher: Sirius Software*  
*Requirements: Apple II*

WHEN you've finished typing in Max Parrott's great darts game (*Windfall*, March) and you've got the feel for Apple-based pub games, two offerings from IDSI might interest you. One is a version of shove ha'penny called Shuffleboard, and the other is a pool package called Trick Shot. Buy me a drink and I'll tell you about them.

Cheers! Shuffleboard offers you two games, Tally All and Cut Throat. Either can be played against another player or against the computer.

Play involves three operations - positioning your puck at the back of the board using U and D keys, moving a dot around your puck to determine the direction of motion, using the --> and <-- keys, and shoving the puck.

When you are ready to shove, a scale is displayed from "slow" to "fast" with a point moving continuously back and forth between them. Hitting the S key shoves the puck at a force proportional to where the point was on the scale when you pressed.

It sounds a little awkward, but after a few tries it is actually quite easy to do. Also, if you decide you're not quite ready to shove you can move back to the aiming phase. The current score and all the options are held on the text screen and you can toggle back and forth between the text and graphics screens via the ESC key.

Playing against the computer doesn't automatically mean that you'll lose. The program has obviously been given the machine equivalent of a few pints, so while it is a good player it is not perfect. The two games differ in the scoring procedure used, and hence may call for different strategies. You can get some idea of the strategy required by watching how the computer plays.

I must admit I quite enjoyed Shuffleboard. It is not an exciting game - in fact it has an almost relaxed air to it, though not as relaxed as the original wood-and-copper version. It is demanding enough to maintain interest, but not too wearying on the grey matter.

Yes, thanks, mine's a G and T, squire. Now, where was I? Oh yes, Trick Shot is a pool package for one to four players. There are four different games available, plus a facility to set up (and save) trick shots. The four games are billiards, snooker, three ball, and open table (which is basically a practice mode, since there are no rules and no scoring).

You begin each shot in "aim" mode, with 128 directions available and an aiming line to show you where the cue

## More pub games for the Apple

ball will go. A ghost ball at the end of the aiming line helps you to see how the cue ball will strike the target ball. Once aimed, you can then select from one of eight speeds for the ball and can then choose to add "English" to the shot. You can achieve all of this from either keyboard or paddle.

Of the four games, I'm most familiar with snooker, so I've spent most time playing that. This version contains some non-standard facilities which might raise an eyebrow or two at the Crucible.

For example, you can position the cue ball anywhere on the table to break off. Also, despite what the manual says, the program failed to re-spot an illegally potted colour on one occasion. The fact that there is no penalty for illegal shots seems a strange omission, and one which spoiled the game for me - with no penalty there is no incentive to avoid such shots.

In addition to the game disc there is another which contains some pre-programmed trick shots. These are fairly amazing, but you can also use the package to build and save your own attempts to emulate Minnesota Fats.

The manual describes Trick Shot as "an exciting extension of Pool 1.5", and it certainly is an extension. There is no overlap between the two packages since Pool 1.5 contains four different games including straight pool, if memory serves me correctly. If you liked Pool 1.5 you'll like Trick Shot, but I think I prefer playing my snooker on a table.

*Title: Shuffleboard*  
*Author: Not credited*  
*Publisher: Innovative Design Software*  
*Requirements: Apple II*

*Title: Trick Shot*  
*Author: Not credited*  
*Publisher: Innovative Design Software*  
*Requirements: Apple II*



## Right way out to the printer

THERE comes a time in the affairs of Apple programmers when they want to send characters to devices other than the screen, or to alter the output in some way before it appears on the screen.

To achieve this, there is a vector in zero page at \$36/\$37 called CSW. This normally contains the address of the screen output routine, \$FDF0, (COUT1) so that all output goes to the screen.

The monitor routine \$FDED (COUT) does an indirect jump (\$6C) via CSW, and not a JMP (\$4C) as in the control "Y" user-vector at \$3F8. CSW is in the normal low/high address format, ie low byte in \$36 and high byte in \$37.

To send the output somewhere else, just put the start address of your routine in CSW. When DOS is in operation however, things are not quite so simple. DOS maintains its own output hook, separate from CSW, at \$AA53/\$AA54. The method of altering the hook is explained in the DOS manual, page 105.

The new output address is put into CSW as usual, and then the DOS "reconnect" routine at \$3EA is called. This moves the address from CSW to the DOS hook, and restores the DOS output to CSW, and then returns. This is so that DOS can monitor all output, looking for DOS commands.

To change the output routine from within a program, several methods can be used. The following examples show how to switch output to a printer in slot 1 (\$C100), and assumes that DOS is in use.

```
LDA #$00 ;Low byte
STA $36 ;CSWL
LDA #$C1 ;High byte
STA $37 ;CSWH
JSR $3EA ;Reconnect DOS
```

Using the output routine \$FDED (COUT) the contents of the "A" register are now sent to slot 1. It can be returned to the screen by using \$FDF0 instead of \$C100.

The equivalent of the Basic (Beginners Awfully Slow Idiot Code) PR# command can also be used from machine language programs as follows, remembering that all DOS commands must start and end with a "CR"

```
LDA #$8D ;Carriage return
JSR $FDED ;Output routine
LDA #$84 ;Control 'D' for
DOS command
JSR $FDED ;Output routine
LDA #$D0 ;"P"
JSR $FDED ;Output routine
LDA #$D2 ;"R"
JSR $FDED ;Output routine
LDA #$A3 ;"#"
JSR $FDED ;Output routine
LDA #$E1 ;"!"
JSR $FDED ;Output routine
LDA #$8D ;Carriage return
JSR $FDED ;Output routine
```

There is a simpler way, however, by using the routine in the monitor ROM at \$FE93 (OUTPORT). This performs the equivalent of a 'PR#0' and re-sets the output to the screen. If it is entered at \$FE95, bypassing the LDA # \$100, with

**TO direct output to a printer rather than the screen when DOS is active sometimes causes problems for Basic programmers let alone assembly language users. This article, by IAN PAWSON, will point users in the right direction.**

the high byte of the required output address in the 'A' register, it will perform the CSW change.

```
LDA #$C1 ;High byte of
slot #1
JSR $FE95 ;Switch output
JSR $3EA ;Reconnect DOS
```

To return output to the screen,

```
JSR $FE93 ;Set video
JSR $3EA ;Reconnect DOS
```

```
JSR $FDF0 ;Prints on the
screen only,
not slot #1.
```

Rather than using JSR \$FDED after each LDA for printing Ascii text, there is a 'standard' print routine that uses the stack to store the 'return' address. This has the advantage that the text can appear in the program where it is used and can be any length. The text must end with a zero. The print sub-routine is called MSGOUT and is used as follows:

```
JSR MSGOUT ;Do print routine
ASC "THIS IS A MESSAGE"
BRK ;This assembles as
zero
XXX ;Next code. MSGOUT
returns here
```

Rest of program

The message printer subroutine is shown in the panel below:

```
MSGOUT FLA ;Get low byte of RTS off stack
STA TEMP ;and save it
FLA ;Get hi byte
STA TEMP+1 ;and save it
TYA ;Transfer 'Y' to 'A'
PHA ;and save it on the stack
LDY #$00 ;Set index to zero
LOOP INC TEMP ;Inc. RTS address for each char
BNE SKIPADD ;If it's zero
INC TEMP+1 ;Increment hi byte
SKIPADD LDA (TEMP),Y ;Get a char from ASCII string
BEQ MSGRTS ;If it's zero then end of string
JSR $FDED ;Output the char
JMP LOOP ;Get a new char
MSGRTS FLA ;Recover 'Y' data
TAY
LDA TEMP+1 ;Get hi byte of RTS
PHA ;Push it on the stack
LDA TEMP ;Get lo byte of RTS
PHA ;Push it on the stack
RTS ;Return to main program
```

These monitor routines scramble the registers so, to keep their values, use the register save routine at \$FF4A (IOSAVE) before the switch and the register restore routine at \$FF3F (IOREST) after the switch.

When CSW is changed output can still be sent to the screen only, by using the screen output routine \$FDF0 (COUT1). This will send the contents of the 'A' register to the screen and not to where CSW points. Thus, using the example above to switch out output to slot 1, the following shows how output can be sent to the screen only.

```
LDA #$C1 ;ASCII "A"
JSR $FDED ;Sends to slot #1
LDA #$C2 ;ASCII "B"
```

The location TEMP can be any zero page location. I use \$06 as this does not conflict with Apple zero page usage.

To show how this can save on code, it can be used for the 'PR#1' routine shown above.

```
JSR MSGOUT
HEX 8D84
ASC "PR#1"
HEX 8D00
```

Rest of program.

If your assembler does not support the HEX pseudo-op, then use DFB, separating the HEX digits with commas and using '\$' before each number.

*Don't forget to close the door quietly on the way out!*

**IN this two part article, PETER THOMASON reviews Micro Planner, a critical path analysis package for the Apple. Many readers will be familiar with the ability of Visicalc to exploit the inter-relationship between numbers. Micro Planner takes things a stage further and allows the user to explore the relationship between a variety of previously defined real life activities.**

**This technique has been used for some time on minis and mainframes, but is comparatively new to micros. It is said that the use of critical path analysis programs shortened the development of the US Navy's Polaris project by two years, which is some indication of the potential of this approach.**

**This month we look at the hardware required and how user input is handled. Next month we will examine how the program presents its output.**

**Critical path analysis is a highly complex area. The review does not set out to explain in depth the inner workings of this technique, but concentrates on how the end user interacts with the product.**

# Critical path analysis and the end user

*Who would be interested in Micro Planner?*

The user is typically every business manager, in every industry and anything in any place that has to plan nearly anything.

*Why would they be interested?*

Most subjects which require a planning and control ability have little complicated things, things which are not quite as simple as they appear on the surface, things which manage to get more complicated the more the manager thinks about them.

Complications come in many forms – in time, resources, inter-relationships, detail and so on. It is to handle these complications that any micro (mainframe) is of use. In Micro Planner's case the greatest benefit is that a manager can depict nearly every possible situation, model it, simulate it and change it, interactively.

*What would they have to do to use it?*

By all accounts – according to the manual – not to read the manual! This is because Micro Planner is an "interactive" program and, as such, has the ability to ask for the information that it needs in order to carry out its various routines.

Continuous use of the manual will only slow down the user's communication with the computer and, in consequence, may retard the learning process.

Having said that, we do not live in a perfect world and there will be times when you and Micro Planner will not be able to understand each other – in those circumstances you will want to fall back on the manual for purely reference purposes.

In preparing Micro Planner, the program writers have taken great care to make the program as robust as possible and new users will find it possible to shed any inhibitions with which they may approach the computer. After trying the different options (deliberately somewhat irresponsibly) I found that Micro Planner will stay cool in most crises.

Micro Planner is a powerful project planning program using the principles of critical path analysis in a flexible interactive environment. Using the basic network

analysis technique, it adds a number of enhancements and extensions which provide a complete package for the precise modelling, effective planning, progressing, analysis and control of complex projects. The program is more than just another version of P.E.R.T. (Program Evaluation and Review Technique).

Once the user has prepared a network diagram illustrating the logical relationship of operations, or activities, to be carried out for any project, full information about the operations can be fed into the system enabling Micro Planner to construct a "model" of the project. Information can also be entered defining and quantifying the resources available to the project.

Using the Time Analysis facility the program can now establish the earliest and latest dates on which each operation can start or finish. With this information, Resource Analysis can calculate optimum start and finish dates for all operations using the available resources.

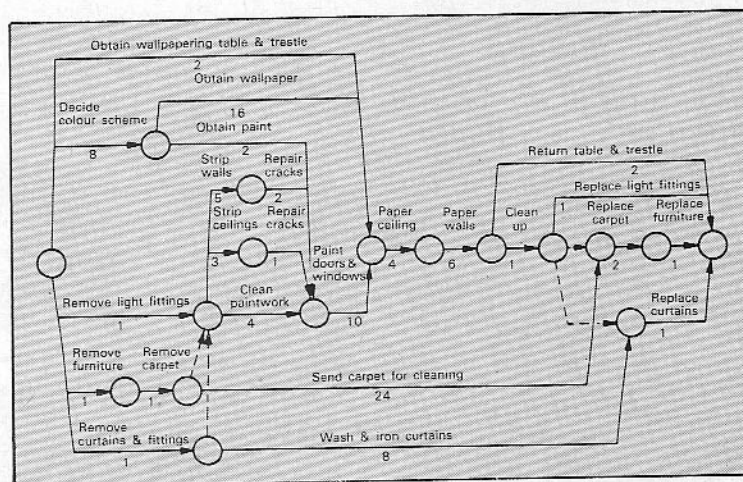
Micro Planner can print the results of these analyses in simple, easy to read charts or schedules.

Micro Planner will run on its own, relying on the Pascal operating system only for organisation of data files. The program resides on two standard 5¼in discs, both of which can initiate any function of the program (identical main menu on each). As one disc drive permanently contains a "work disc", the system works most effectively with three drives. If, however, the third drive is not available then the user is prompted when program disc changes are required.

The policy of Micro Planner Services is to provide a full back-up service to users. A small annual maintenance charge is made to cover administrative costs and issues of updates. Program discs are issued with "security back-up" copies.

The Micro Planner system consists of the following:

- 48k Apple II (or IIe).
- Language card with 16k memory (Apple II).
- Printer interface card.



*Typical network diagram*

Real time clock card (optional, it doesn't work with an 80-column card).

Three disc drives with control card (only two in two drive version).

Video monitor.

Printer.

All necessary interfaces and cables.

The program will support an 80 column card. I used a Videx although other cards may be supported - check with supplier.

Micro Planner will operate using two or three drives, but when just two drives are fitted, it will only run projects with three subprojects or less. It is essential that the system includes Apple Pascal system software and documentation.

Any printer which is compatible with the Apple Pascal operating system can be used but a printer of at least 132 column width should be chosen in order to use all the outputs available. It is not sufficient to ensure that the printer is compatible with the Apple II - some printers and/or interfaces can work perfectly with DOS 3.3 but may not work with Pascal, others will work with Pascal but require different switch settings.

The Apple III Micro Planner System requires the SOS system software and documentation, a 256k Apple III, Profile hard disc and security chip.

Micro Planner uses the "Activity on Arrow" convention. Projects are built up using "subprojects" - the resulting network may contain a number of unrelated (or related) projects, that is subprojects, which are only common in their beginning and ending events.

The input programs are fully interactive. A great deal of error checking is carried out by the input programs and users will find that, when problems are encountered, Micro Planner is able to take care of itself and to retrieve such situations in an almost friendly way.

The Input System operates as follows:

## Project data

It establishes or modifies basic criteria for calendars and timing:

- One, two or three levels of time reporting may be specified by the user for each project; major, minor and secondary, for example, weeks, days and shifts.

- Project base date. While Micro Planner will recognise any date between 1950 and 2049, the base date defines the start of the calendar for each project in question.

- Time now date - The date from which current analyses will commence.

- Normal weekly cycle of working and non-working time periods.

- Holidays, that is selected non-working periods.

- Overtime periods, that is selected working periods.

## Network data

- Using a form filling technique, this module establishes or modifies a file of network data including all activities and "special events" - that is, logical starts,

logical finishes, interfaces etc (normal events are accepted by default). Event numbers alone can be changed, this avoids the necessity to delete and re-add activities when network logic changes and eliminating one of the major disadvantages of arrow networks as against precedence networks.

## Activities data

This routine allows the data to be specified as follows:

- Record identification (preceding and succeeding event numbers).

- Uniqueness identifiers where more than one activity shares the same preceding and succeeding events.

- Activity or event types and calendar specification.

- Duration.

- Imposed dates and time restrictions.

- Varied progress reporting including:

  - Date when actually started.

  - Date when actually finished.

  - Balance of duration yet to be carried out.

  - Percentage of duration completed.

**Special events may be specified in a similar way.**

Resources data establishes or modifies a file of resource availability and uses a form filling technique similar to that used for activities and events. Resource levels can be set to change at any date. There may be up to 20 changes of resource level throughout the calendar.

The time analysis program carries out the usual calculations. Micro Planner has these additional features:

- Negative durations catered for (but not if resource analysis is used).

- Reverse logic (otherwise called short-path indicators) events, that is where the event can be achieved when the first preceding activity is completed.

- Interfacing of subprojects at any event and with more than one interface between any two subprojects.

- Progress accepted as any of: start date, finish date, event achieved, time to completion (remaining duration), percentage completed.

- Hammock activities, that is activities which adjust their durations to span from start to finish events (mostly used for depiction of project overheads).

- Results of analyses can be stored in one of four schedules - three of which can retain historical data unchanged by future time analyses.

## Additional features include:

- Progress report: compares previous and current schedules in order to evaluate any variation and to highlight progress "slippage" (any two of the four schedules can be compared).

- Full size bar chart. Displays any schedule in bar chart form. Bar charts are assembled from vertical strips of printed output which are joined together to form full sized wall charts.

- Key events. A listing of key events comparing current forecasts with earlier

programmed dates.

- Resource and cash flow histograms.

Activities are selected, rejected or sorted on the following criteria: Event names, float values, time analysis or resource analysis dates, sort codes and responsibility or zone labels or codes.

A hand-pick selection option is also provided allowing only the selected records to be printed. Listings of file contents are available.

## Program capacities (Apple III in brackets)

Maximum number of:

Activities per project guide - 850 for basic configuration (4000)

Activities and events in subproject: 150 (512)

Subprojects: 25 (27)

Calendars: 6 (6)

Resources: 35 (35)

Resources allocated to any one activity: 20 (20)

Calendars: 2144 smallest time units (2144)

Activities to be listed in any one output: 850 (4000)

Activities to run "in parallel" in resource analysis: 100 (500)

## Ladder activities

Many activities in a project are made up of repetitive phases. Because of this, it is often possible for one activity to start before the preceding activity is finished. Where this kind of relationship between activities exists, they are known as progressive feed activities, since each completed phase of the first activity enables the corresponding phase in the succeeding activity to begin.

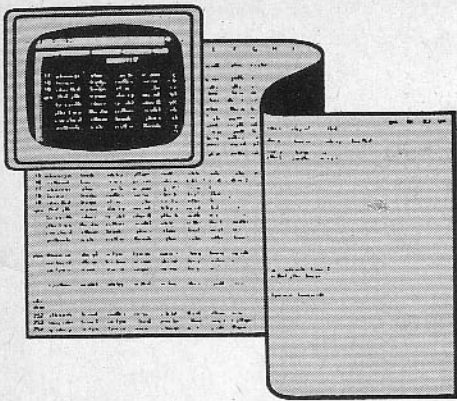
The ladder activity is a network convention for depicting progressive feed activities without increasing the complexity of the network, yet allowing the computer program to interpret the network accurately. The rungs of the ladder represent the overlapping activities. The left hand side indicates the time between the start of one activity and the next, the "lead time". The right hand side indicates the time between the end of activities, the "lag" time.

On the Apple III version a project edit facility is available. This is the equivalent of a normal textfile editor with the addition, deletion and renaming facilities one might expect.

The Micro Planner project editor is designed to offer flexible transfer of logic data from one project to another or within a single project.

Users of Micro Planner will know that projects can be built up of one or more subprojects. Within the limits of the maximum subproject size, subprojects can consist of any number of activities. The function of the project editor is simply to transfer all or part of a subproject from any subproject to any other subproject. This facility will be of great benefit to regular users.

**NEXT MONTH: How resource scheduling works.**



# This latest VC manual makes for heavy going

ONE of the most exciting things that has happened to me recently, professionally speaking of course, has been a chance encounter with The Spreadsheet program by William Graves, published by A.P.P.L.E. (Apple Pugetsound Program Library Exchange).

For all intents and purposes The Spreadsheet is a menu driven Visicalc program with the added appeal that it has many of the features which up to now were only available on the Apple III's Visicalc advanced version - for example, protecting cell from being accidentally overwritten, having a VC model with columns of variable width, hiding the content of cells containing confidential information, rounding numbers to a desired number of decimal places, replicating VC models as a block instead of having to replicate them line by line or column by column, etc.

The program also has a number of utilities which are currently only available on peripheral VC discs such as locking VC data files. I have no doubt that when you get to know The Spreadsheet, many serious Visicalc users will prefer to load, save and manipulate their current VC data discs with The Spreadsheet program rather than with the original VC one.

At this point I must declare my interest in the subject: I liked The Spreadsheet and the CDEX Training Pack for Visicalc (mentioned in an earlier article) so much that I have decided to become a distributor for them. I hope however that you will not, because of that, regard my comments on these packages as purely sales talk.

I have only been using The Spreadsheet on and off for about a couple of weeks, and I was delighted from the outset to discover that on loading the program it immediately recognised my 128k Vergecourt Ramex card so providing me with 158k user memory. (A simple calculation shows that The Spreadsheet occupies 34k of the Apple's memory, compared with 30k required by the Visicalc program.) If you use the program with a Ramex card in a 48k Apple then the user memory that will be available to you on loading the program will be 142k.

The Spreadsheet will also either recognise, or can be configured to recognise, your 80 column card. If on the other hand

you do not have an 80 column card but you still wish your monitor could display on the screen a wider portion of your Visicalc models, then at the stroke of a key The Spreadsheet has the facility to do just that, except that the display will be only 70 columns wide and it will cost you 12k of user memory.

So if you want to load Spreadsheet displaying 70 columns on a 48k Apple (without an 80 column card) then do not expect to be left with more than 2k user memory. But if you have a 64k Apple without an 80 column card, you can get a

put on a disc the various VC models which appeared since June 1982. Well, I started work on this project but I did not like the end product. First of all I now have new ideas on how to improve and either simplify or elaborate on many of the models discussed in the earlier articles. Secondly the commentary accompanying such a disc would have to be different from the relevant earlier articles, because those articles were primarily addressed to readers who did not possess a disc.

The necessary changes are almost as if a DIY programme was transferred from a radio to a television studio. You still need the commentary, but it is bound to be completely different.

In order to overcome this problem in the future, the publishers of *Windfall* and I are considering the proposition that if we are to make available to the readers of the magazine a VC data disc containing the VC models and datagrams being discussed on any particular month, then part of the article will inevitably have to be written in a way that will be more meaningful to readers who can access the disc, ingful to readers who have the disc, then to readers who cannot access it.

The acquisition of the disc will, of course, be optional, and if the scheme goes ahead it is also envisaged that future articles in the Visicalc series will contain material not aimed only for readers wishing to send for a supplementary disc.

Readers applying for any discs accompanying the Visicalc column in

---

By NICK LEVY

---

70 column VC display on your monitor and still be left with 18k of user memory.

The Spreadsheet does not have any of the trigonometrical functions which are available in Visicalc, but has other functions such as @COL and @ROW which return the numerical value of a column or the row, as well as a @ ROUND function as mentioned earlier. At the moment I have a problem with using The Spreadsheet for writing datagrams, but the program's other attributes more than make up for what I hope would be only be a temporary irritation. The Spreadsheet comes with a 160 page manual and costs £100.

Readers will remember my promises to

CASHFLOW SUMMARY  
FOR PERIOD 7

A.B. JONES LTD, ENGINEERS, MILLERTON, YORKS.

RANGE OF FORECAST:

FROM: AUGUST 1983 TO: JULY 1984

DATE OF F'CAST: MARCH 1983 BANK BRANCH MILLERTON YORKS.

DATE OF PERIOD: DEC 1983

	CURRENT PERIOD				YEAR TO DATE			
	BUDGET	ACTUAL	VARIANCE AMOUNT	%	BUDGET	ACTUAL	VARIANCE AMOUNT	%
<b>-RECEIPTS</b>								
REVENUE ITEMS	21214	22063	849	4.00	145130	150982	5852	4.03
CAPITAL ITEMS	1635	1575	-60	-3.67	23119	22211	-908	-3.93
<b>TOTAL RECEIPTS</b>	<b>22849</b>	<b>23638</b>	<b>789</b>	<b>3.45</b>	<b>168249</b>	<b>173193</b>	<b>4944</b>	<b>2.94</b>
<b>-PAYMENTS</b>								
<b>--REVENUE ITEMS</b>								
TRADING EXPENSES	11817	12695	878	7.43	80339	86300	5961	7.42
ADMIN EXPENSES	5177	5235	58	1.12	35420	35800	380	1.07
SELL & DIST EXES	260	276	16	6.15	1799	1911	112	6.23
FINANCE COSTS	55	87	32	58.18	546	706	160	29.30
<b>--CAPITAL ITEMS</b>	<b>1436</b>	<b>1420</b>	<b>-16</b>	<b>-1.11</b>	<b>23698</b>	<b>24178</b>	<b>480</b>	<b>2.03</b>
<b>--OTHER PAYMENTS</b>	<b>2689</b>	<b>2722</b>	<b>33</b>	<b>1.23</b>	<b>19863</b>	<b>20015</b>	<b>152</b>	<b>0.77</b>
<b>TOTAL PAYMENTS</b>	<b>21434</b>	<b>22435</b>	<b>1001</b>	<b>4.67</b>	<b>161665</b>	<b>168910</b>	<b>7245</b>	<b>4.48</b>
SURP OR DEFCT (-)	1415	1203			6584	4283		
OPENING BK BAL	-413	-2594	BGT OD IR:	15.50	ACT OD IR:	15.75		
CLOSING BK BAL	1002	-1391	BGT CR IR:	12.50	ACT CR IR:	12.75		

Exhibit 1: A typical monthly summary

Windfall will probably charge the payments for them against the cost of learning Visicalc, which brings me to another topic – why is the VC manual inadequate as a self tutorial program and what you can do about it?

As my organisation provides training and consultancy in using Visicalc and other electronic spreadsheets, I must declare from the outset that this is a subject matter in which I have a vested interest.

When Visicalc made its first appearance, version 1.0 was accompanied by an excellent 77 page manual divided into four well structured and manageable lessons, each with a hierarchy of numbered sections. The manual was so highly thought of at the time that rumour had it that with such a manual you could learn Visicalc in half an hour!

Then came the current version of VC accompanied by a 200 page manual. What does the new version of Visicalc have which is not in the first version (beside being on a 16 sector disc as against a 13 sector one)? One new very useful command, /E (Edit), nine new invaluable logical functions (@IF, @AND, @NOT, <, >, etc.), and a couple of logical operators.

But the tutorial section in the latest VC manual does not make any reference whatsoever to these new indispensable functions, the use of which makes all the difference between rudimentary financial

modelling and the more advanced techniques which are necessary for modern business planning. They are only mentioned briefly in the reference section.

Furthermore, the section numbering in the latest VC tutorial, which covers the same topics under the same headlines as in the first manual – except the references to editing – has completely disappeared.

This has, in my opinion, made the tutorial much more difficult to follow, and many keen Visicalc students just cannot get on with the latest VC manual. Incidentally, let me assure you that as someone with vested interest in teaching Visicalc, I made no payments to Visicorp to make their latest VC manual less conducive to learning, compared to the first edition.

Both the earlier and the current tutorials in the VC manual concentrate primarily on teaching you how to use the Visicalc commands (that is, those instructions that you give to VC which begin with / like /B, /C, /D, /E etc.), and on the technical peculiarities of operating a teletype keyboard (for example, pressing ! recalculates the current spreadsheet, keying ESC or CTRL-C removes typing errors, having to type first the symbol " if you want to enter text starting with the symbol \$, etc.).

The tutorial makes scanty references only to the Visicalc functions which existed in the 13 sector version, but none to such functions beginning with @IF, @CHOOSE, @ISERROR, @TRUE, etc. Yet

the real art of using Visicalc and making the most of it lies in knowing when and how to manipulate those little-known VC functions which were left out of the VC tutorial.

Mastering the technique for replicating a range of Visicalc formulas, and knowing when to respond with R for Relative and N for No change, is something that with practice and patience you could learn from the VC tutorial, although you will probably be better off learning this technique if, at first, you had the opportunity of "sitting next to Nelly" and watching how the experts replicate VC formulas.

But when it comes to learning the VC functions you are on your own. Let me give an example of a typical modelling requirement which the VC manual will not give you even a remote idea on how it could be done.

Suppose you have a cell showing the variance between a budgeted sum and an actual sum. Suppose also that next to that cell you want to show one asterisk if the variance is more than 10 per cent favourable, three asterisks if it is more than 10 per cent unfavourable, and no asterisk if the variance is within plus or minus 10 per cent of the budget – not an unreasonable request, which Visicalc can be made to perform.

The formula for such a calculation will consist of two @IF statements nesting inside a @MAX function. Now you can

# VISICALC

learn from the VC manual how a @IF function is structured and how a @MAX function is structured, but nothing about imbedding @IF functions in a @MAX function, or the magic that you can perform by a @CHOOSE function nesting in a @COUNT function.

Furthermore, the VC manual does not tell you anything about datagramming, which is so essential if you want to use Visicalc to consolidate accounts. And even if you know how to write datagrams it's possible that you would not know what to do if your datagram is more than 254 lines long, unless of course you read R. Gears-Evans' excellent article on "A touch more magic" in the May issue of *Windfall*.

I must confess that from my point of view R. Gear-Evans has revealed what I consider a professional trade secret for which I would normally charge a consultancy fee. After all, if you mention that a word processor can be used to write datagrams for Visicalc you inevitably get involved in endless questions of how to use the word processor. But then I wonder how many readers took a note of his and some of my revelations? When in the April issue of my column (The economics of using electronic worksheets) the publishers printed by mistake only the left half of the VC models which accompanied the article, only two people

remarked to me that something in the presentation appeared to be wrong.

Returning to the Visicalc manual, the inevitable conclusion must be that the latest version is a rather cumbersome foundation course for learning Visicalc, and that the best way of learning how to use, especially the more advanced features of the program, is by studying closely the VC models of other more experienced Visicalc users, preferably under guidance.

Some time ago I bought the manual and the two discs accompanying the Visicalc templates on real estate management. More recently I acquired Cash Matrix, the Visicalc template for cash flow planning and analysis. I have no connections or dealings whatsoever with the real estate business, and I have no intention at the moment of using the Cash Matrix templates to project my cash flow. So why did I add these two packages to my library of software? And why did I spend hours and hours studying them in depth?

The reason is because I can learn more from these packages about using and manipulating Visicalc that I can learn from the Visicalc manual. A well thought out and well written VC application model or package fills me with fresh inspiration and provides me with new ideas on how to develop my own models, or what to advise other companies on how they

should develop and operate their own models (some people would probably call this unshameful plagiarising, but then, learning from each other is probably the best way to go if you want to learn more about Visicalc).

I was particularly awed by the creative brilliancy of the authors of the Cash Matrix package, Edwin Smith and Stephen Mandelin, who produced what is probably the first ever completely menu driven Visicalc application program (in this case a model for forecasting cash flow).

It incorporates, among other things, menu driven instructions to produce summaries of monthly and year to date variances analysis (*see Exhibit 1*), as well as menu driven instructions to print automatically the 63 row cash flow worksheet on either a 224 column page or on five 80 column pages (including a very neatly designed title page).

Even preparing the cash flow results for plotting on the Visitrend+Visiplot program is done via a simple menu. I know that this was not the intention of the authors, but I for one was much more interested in the process of how the program interfaced with the computer and vice versa, than in the actual results of the cash flow projections. ☛

● *Nick Levy is the principal of Interface Management Resources and gives specialist lectures on spreadsheets.*

## The **FORECASTER**

takes the

$$g(T) = \sum_{j=0}^{T-1} \beta^j x(T-j) f(-j)$$
$$= \left[ \begin{array}{l} \sum_{j=0}^{T-1} \beta^j x(T-j) \\ \sum_{j=0}^{T-1} -j\beta^j x(T-j) \end{array} \right]$$

out of forecasting

Available NOW on the Apple II,  
Ile and III

Cost £320 + VAT

Ask for details at your Apple dealer  
or contact MPD at this address

The first in a new generation of forecasting software which puts the full power of the latest mathematical methods at your finger-tips.

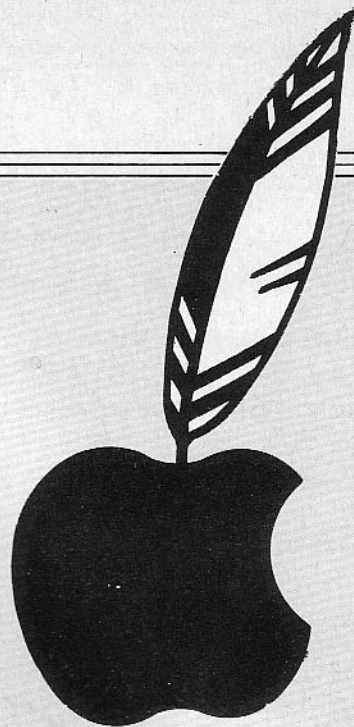
For everyone who needs to forecast the future: SALES, PROFITS, PRODUCTION LEVELS, PURCHASES, PACKAGING, FUEL COSTS, DAYS LOST THROUGH SICKNESS - the FORECASTER is the answer.

### m.p.d

MICROCOMPUTER PROGRAM DESIGN

**2 Hillside Place  
Newport-on-Tay  
Fife.**

# Hang on to your variables with this 'miracle POKE'



*IN the May issue of Windfall Mr C.J. Palmer wrote in with a problem concerning re-starting a program which uses indirect DOS commands with an immediate GOTO command.*

*The problem was that if a program was halted by CTRL-C or a STOP command or an error and the operator wished to restart without losing the values of the variables the way to do it would be to use the CONT or GOTO linenum immediate commands.*

*But this doesn't work, because as soon as the restarted program reaches an indirect DOS command execution will stop with the error message "NOT DIRECT COMMAND".*

*Mr Palmer asks is there a miracle POKE to stop this happening and the answer is, yes there is.*

*When DOS wants to know whether the computer is in immediate or deferred mode, that is whether or not a program is running, it does this by checking whether the last prompt character issued was (in Applesoft) a right square bracket. If it was, then the computer is in immediate mode. A value corresponding to the last prompt issued is stored in zero page memory location \$33 (51 DEC). When a RUN command is issued the value stored in this location is replaced with the value 6.*

*When the computer encounters an indirect DOS command it checks location \$33. If it contains the value 6 then the computer is running a program, so the command is accepted.*

*If program execution is interrupted in any way then, following the ERROR or BREAK message, the computer will issue a right square bracket prompt, thereby changing the value stored at location \$33. The immediate commands CONT or GOTO will not alter the value stored in this location, therefore any subsequent indirect DOS commands will not work.*

*The miracle poke is therefore POKE 51,6 but it is no use typing in POKE 51,6 then pressing RETURN as you would immediately get another square bracket prompt and be back where you started.*

*The answer is to use a multi-statement line consisting of POKE 51,6:CONT or POKE 51,6:GOTO linenum - and you will then find that your problems are over.*

*- Colin J. Davies, Liverpool.*

## Character generation

*I HAVE the program "Animatrix" and all various types of letter character sets, some 20 in all.*

*However how is it possible to use*

*various character sets with, say, Applewriter or for that matter for use in other programs? What instructions does one have to give to the Apple to make it use a particular character set as a normal writing and printing medium? - Frank E. Brooks, Bukit Baru, West Malaysia.*

● Animatrix, a character generation program contained on the Apple Toolkit Disc, is designed to produce characters on the high resolution screen. The letters are created by building up each character using a series of dots. The mechanism by which the dots are manipulated to form the character is not directly applicable to the mode of operation of a dot matrix printer.

The second of Mike Glover's series on Understanding the Epson in the June issue of Windfall covers the use of your own characters from within Applewriter II. This combination of soft/hardware may not directly match your own, but the techniques used should provide a clue to the solution to your problem.

## The need to copy

*I AM writing with reference to the article in Windfall concerning the copying of software.*

*I am in the fourth year at school and use an Apple II Europlus. Together with a friend, we are the main users of the Apple. The school is still very small (it does not yet have a fifth and sixth form as the buildings are not yet complete) so we don't have as much support as is really needed. This means that we don't have enough money to buy any reasonable software. We can obtain educational software, but most of this is in Basic and suffers from being slow and "crashable".*

*The only method my friend and I have*

*is to copy software from other sources. By doing this we can have Structured Basic, a compiler, bit copying programs and various other useful utilities.*

*We have been advised to stop copying software as it could ruin our reputation, though we came first in a national competition on the strength of a program on the Apple.*

*We didn't really consider it cheating the author to copy software that we wouldn't have bought anyway. The only way we could get the software was to copy it. Ironically, we were copying from another school. They had a large computing department and were able to buy the software.*

*Perhaps software houses should offer software to schools at a discount. In return the school would promise to buy all, or at least most, of its commercial software from that software house.*

*Where games are concerned, they are so cheap that if three or four pupils club together we can buy them, but being so cheap there seems little point in not copying it. The school would certainly not buy games even if it could afford them.*

*Perhaps the local authority should be able to buy software for distribution to all its schools. I am sure that a small business with 10 Apple computers would not buy 18 copies of a word processor at £200 each.*

*- Stephen Mehew, Strathaven, Lanarkshire.*

## Lower case generator

*IN the March issue of Windfall there was an advert for a lower case generator. Please could you inform us of the mode of operation of this device, and if it would be compatible with the Wordstar word processing and DMS record keeping packages.*

*We are also anxious to learn which chip*

on the motherboard this lower case generator would replace.

I should add that if the chip is capable of overcoming the problems of obtaining lower case in the above packages, we would be interested in ordering eight of them. — **M.S. Harris, Agricultural Training Board.**

● The chip would be compatible with Wordstar and DMS but these really require the use of an 80 column text display which this does not provide.

The chip replaces the character generator chip which is positioned on the motherboard under the keyboard nearest the operator. Installation necessitates unscrewing the fixing screws in the base, then rewiring the upper case together with keyboard, pulling out the old, putting in the new and reversing the process of dismantling.

The choice of 80 column card is important as it should facilitate most, if not all, of the Apple's screen output capabilities. Particularly important for a variety of packages such as Visicalc, Wordstar, Multiplan etc, is the ability to produce inverse characters.

## More on CHAIN

RE "Ways of overcoming problems with CHAIN", in the May Windfall, I am pleased to be able to add to the reply given by Max Parrot.

In a nutshell it IS possible for Aileen Beattie to:

● Store all variables above HGR2 and maintain their values when swapping programs.

● Store and exchange programs and parts of programs below HGR2 as often as required without clearing the variables, disturbing LOMEM, using CHAIN or interrupting program execution.

The way to achieve this is described in an article by David A. Lingwood entitled "Overlaying in Applesoft" and published in Call-APPLE (Nov/Dec 1980).

The solution is nothing short of brilliant in its simplicity and usefulness and I can promise that the expense of acquiring the back number will be repaid many times over, it can even enable you to reduce disc storage requirements on some occasions.

I will not infringe the copyright by describing the method but I will guarantee that it works and very well too, and the article is very clear.

The only time I use CHAIN instead of

the Linwood overlay method is when I want to move LOMEM.

The source for Call-APPLE is SBD with whom I have no connection other than as a satisfied customer. — **David Sutton, Gatley.**

## Files problem

I HAVE encountered a problem with writing text to files. Every time the record includes a comma or a colon the remainder of the text is chopped off and I get an EXTRA IGNORED message.

It is possible to overcome this to some extent by GETting each character separately and adding it to the preceding ones to make up a string which is finally written off to the file, but the same problem crops up when I try to retrieve the record.

Other people must have experienced this difficulty too, and if anyone can suggest a way round it I would be very grateful. — **Steve Jones, Galashiels.**

● There are two relatively simple ways of coping with this problem. The first complements your solution to the INPUT from the keyboard, namely using the GET statement and building the strings up before sending them to disc.

Where you were using GET instead of INPUT from the keyboard you can use GET instead of INPUT from the disc. Then with the routines you have already built in you can handle commas and colons.

The second method is to select another delimiter in place of a comma, say a double asterisk, send this to disc and when read back in they can be replaced by a comma by the appropriate lines of program.

## Apple on the air

AS amateur radio is my hobby I wonder whether you could help me to obtain details as to how I could use my Apple II to receive and transmit Morse Code and RTTY. — **Dr. Joseph R. Bartold, Valletta, Malta.**

● The British user group for radio amateurs is Radio Comms/Apple User Group, Keith Davenport, Hyde, Cheshire, England (Tel: 061-366 8024).

## Sideways scroll

I WOULD like to know if there is a command in Applesoft, or a machine code, which can be POKEd and or CALLED which will scroll the screen from right to left automatically. Can you help? — **Richard Bowles, Eastbourne.**

● A feature on this subject is planned for a future issue of Windfall.

## The value of games

I AM studying in the International School of Tanganyika, which uses an Apple II. I've just started to learn to programme and I find it extremely interesting and worthwhile. The school subscribes to Windfall. I've read many and I find it very helpful, for example the Crash course in Basic, from which I've learned quite a lot of things.

The program listings are also helpful, for example the listing of Ski-run in the issue of September 1981. I've copied the program onto a disc and it's a good game.

I am using this program to develop my own programming skills. I try to interpret each line and see what it does.

I am also interested in becoming an Applescribe and to join the exclusive Apple club. If it is possible I would like the Apple tie. — **Bhavin Radia, Dar-Es-Salaam, Tanzania.**

## Well received

JUST a note from "down under" to express my interest in your magazine, which is reflected by my attached order for back issues.

I hope that your past and future issues are as interesting and informative as the October '82 one which I have. Keep up the good work! — **Steven Spink, Stepten, South Australia.**

## Recruits sought

IF there is anyone in the Newcastle Upon Tyne Area interested in joining "The Apple People" — **Apple User Group (North-East)** I can be reached on Jarrow 898436 (day-time). — **J. Gatens.**



## The not so mysterious world of PEEKs, POKEs and CALLs

FOR most new computer users, the world of POKEs, PEEKs and CALLs must appear like some kind of no-man's-land. It is not, strictly speaking, a part of Basic (from which such operations are initiated) since it involves more direct communication with the memory than other basic commands. Yet at the same time it is not genuine machine code or assembly language (as executed via a "hex dump" or an assembler), though POKeIng is often used to set up small machine code routines from within a Basic program.

Confused? Don't be – the world of POKEs, PEEKs and CALLs is really rather fascinating, *fairly* easy to handle (especially if you're already familiar with the general pattern of operations of your machine), and can be the key to improved programming.

The first thing you really need to understand in order to use these instructions to the greatest effect is that the Basic language (or any other high level language, for that matter) is complete gobbledegook as far as the computer is concerned. Thus much of the ROM (Read Only Memory) in a computer is taken up by an interpreter, a program which translates Basic statements (or Fortran, Cobol, etc.) into hexadecimal numbers. Hexadecimal is simply a numbering system which uses base 16, instead of base 10 which we use in the decimal system.

*When is a number not a number?*

In the decimal numbering system each column of digits in any calculation (read from right to left) can be read as a multiple of 10:

10x10x10x10(=10 000)  
10x10x10(=1 000)  
10x10(=100)  
10x1(=10)  
10x0.1(=1)

and so on. In other words, the value of each column is the value of its right-hand neighbour times 10.

Because we are so familiar with decimal numbers we fail to realise that we actually carry out a piece of very rapid mental arithmetic every time we identify the value of a decimal number. For

---

By A.J. BRADBURY

---

example, the value of the decimal number 255 is actually:

2 x (10 x 10) = 200  
5 x (10 x 1) = 50  
5 x (1) = 5+  
= 255

Hexadecimal numbering works in exactly the same way, except that the value of each column (again from the right) is the value of its right-hand neighbour times 16. Thus:

16x16x16x16(=65536)  
16x16x16(=4096)  
16x16(=256)  
16x1(=16)  
16x0.0625(=1)

Of course there is one small problem attached to the use of hex numbers – the fact that the values to be indicated by a single digit lie in the range 0-15. To get round this the numbers 10-15 are represented by the letters A, B, C, D, E and F respectively (which is why an "address" – a single memory location – can make sense of "numbers" like DA and BF).

So far, so good, but why do we need to

use hex code in the first place?

Strictly speaking, you don't need to know anything about hexadecimal code at all so long as you stick to Basic. However there may well come a time when you want to use machine code (raw hex code), or assembly language (a sort of "Anglicised" machine code).

And even if you don't want to go that deeply into the business of programming there will almost certainly be times when you will want to POKE short machine code programs into memory (this is the only way to get music out of an Apple, for example). It always helps to have at least a vague idea of *WHY* you are doing *WHAT* you are doing.

OK. We've learnt a little about hex code, but what has this got to do with the computer – does it read hex code? At the risk of becoming totally obscure, the answer is yes and no. You see computers usually use hex code, but they only truly understand binary code. This works in the same way as decimal and hex, but uses base two. In other words, each column of a binary calculation carries one of two digits – 0 or 1.

The reason why the computer prefers binary numbers is because it is the equivalent of telling each "bit" (a minute switch) whether it should be ON (that is, set to 1) or OFF (cleared to 0). These bits are gathered together in groups of eight, and

# BEGINNERS PLEASE

each group is known as a "byte". Thus each address in the memory refers to one specific byte. (Just as a matter of interest, half a byte - four bits - is known as a "nibble". Binary numbers are often written in nibble form to make them easier to read.)

Assuming that you are still with me, you will quickly realise that the highest number that could possibly be stored in one byte is 255:

$$\begin{aligned} 2 \times 2 \times 2 \times 2 \times 2 \times 2 & (=128) \\ + 2 \times 2 \times 2 \times 2 \times 2 & (=64) \\ + 2 \times 2 \times 2 \times 2 & (=32) \\ + 2 \times 2 \times 2 & (=16) \\ + 2 \times 2 & (=8) \\ + 2 & (=4) \\ + 1 & (=2) \\ + 0.5 & (=1) \end{aligned}$$

So now you know why, if you've ever tried to do it, you can't POKE a value bigger than 255 into a single memory address. And you may also have noticed something else - that the highest value 8 bits can hold is also the maximum value of two hex digits:

**DECIMAL - 255**  
**HEX - \$FF**

*(the dollar sign is the standard prefix for a hex number)*

**BINARY - %1111 1111**

*(the % symbol is the standard prefix for a binary number)*

"Hold on", you cry, "if that were true then the Apple could never cope with a number greater than 255!"

That's true, in a way. But the Apple has more than a trick or two up its electronic sleeve. Let's take an address above 255 as an example.

Suppose I wanted to change the colour being used in a hi-res graphics routine without using the HCOLOR command. To do this I could POKE the value of my chosen colour into address 804, which is there for that particular purpose. What I tell the Apple is:

```
POKE 804,3 (to set the colour to white)
```

The first thing that the computer does with the number 804 is to convert it into hex code - \$0324. Then it splits the number into two parts, \$03 (the "high order byte") and \$24 (the "low order byte"). The high order byte tells the Apple which "page" of memory to turn to, and the low order byte tells it which address within that page it should handle. So the Apple turns to page 3, address 24, and switches on the first two bits only (binary 0000 0011 = 3).

So the total range of addresses that the Apple (or any other 8 bit computer) can deal with are in the range \$0000-\$FFFF, or 0-65535. This gives a total of 65536 addresses in practice, since address 0 actually exists and can be used.

(Note: Everything that I've said so far is related to Apple computers with 48k of memory. The use of larger amounts of memory, by means of RAM cards, etc., is achieved by leading the computer to think that the additional memory is within the

0-65535 range. A sort of electronic schizophrenia.)

Now that we've covered the background material we can consider the real meat of this article - PEEKs, POKEs and CALLs. Since the best way of learning anything in computing is by doing it, let's run through a few very short programs to see how these commands actually work.

At this point your computer's memory should be free of any other program - if necessary type NEW followed by RETURN to insure that this is the case.

The CALL command, which we will deal with first, is a means of using machine code/assembly language subroutines from Basic. The routine which you CALL may be one that is already in the Apple or one that you have created yourself. The number which follows the CALL instruction must be the address of the first step of the routine you wish to use, for example, type:

```
CALL -1184,RETURN
```

You should now have a clear screen with Apple II on the top line, and a square bracket and cursor at the left hand end of the second line. The reason for this is that you ordered the Apple to execute a built-in subroutine which it normally only uses during a "cold start," that is, when it is first turned on. The reason for the cursor lies in the fact that most built-in subroutines have RTS as their last instruction. In assembly language this is the mnemonic for Return from Subroutine (or Return To Sender), which in this case meant, in effect, "go back to Basic and wait for another instruction". Note that a CALL command uses information within the computer but does not change any of the values within the CALLED subroutine, which stays intact.

Now type:

```
POKE 50,127,RETURN  
PRINT "HELLO",RETURN
```

When the computer obeys your second command you will find that the word HELLO appears in flashing letters. This is because address 50 (\$32) is the location which carries the information which tells another built-in subroutine (known as COUT1 - 'character out') whether output to the screen should be NORMAL (white or black), INVERSE (black letters in white squares) or FLASHING (NORMAL/INVERSE/NORMAL/INVERSE/etc.).

We might call address 50 a "dedicated address," in that it has one specific function. Certainly it may be used as a temporary storage location so long as nothing is being output to the screen, but if COUT1 PEEKs at 50 in search of information then it will take whatever value it finds (it can't tell a "good" value from a "bad" value), and try to use it.

Whenever you POKE a value into an address the previous value held in that byte will be permanently erased. The "legal" values for any byte will depend upon the function that byte serves.

The values that address 50 may legally contain are 63 (\$3F) for inverse print, 127 (\$7F) for flashing print, and 255 (\$FF) for normal print. Any other number is "illegal"

and will interfere with the normal operations of the computer. For example, type:

```
POKE 50,189,RETURN  
PRINT "HELLO",RETURN
```

Once again you will see the instructions which you typed in printed in NORMAL characters. Yet the computer will seem to have ignored your print command. In fact it hasn't. The word HELLO is on the screen but in the same colour as the screen - black! 189 is an illegal quantity in address 50, and in this case it prevents bits being turned on in the positions necessary to allow the any word to appear visible on the CRT. Next type:

```
POKE 50,255,RETURN
```

Right, now that we're back with NORMAL print, let's try something else:

```
10 HGR  
20 FOR X = 8872 TO 8900  
30 POKE X,127  
40 NEXT X
```

When you RUN this program you will see a white line appear about halfway down the first hi-res graphics screen. Why? Because the addresses in the FOR statement (8872-8900) are part of a set of 40 consecutive locations which contain the information relating to line 104 of HGR. These addresses are "semi-dedicated" only, and POKing to them while using HGR can have rather unusual results.

If you are using HGR then the numbers held by the bytes relating to the HGR screen affect the number of bits turned ON or OFF. In fact only the first seven bits can literally appear on screen, which means that the highest legal number in one byte will be 127 (which will turn all seven bits on in each byte - hence the solid white line). However, each byte can actually hold a value up to 255 (as normal), but any value higher than 127 will also affect the next byte in the row:

Type:

```
10 HGR  
20 FOR C = 1 TO 255 STEP 5  
30 FOR X = 8872 TO 8900  
40 POKE X,C  
50 NEXT X  
60 PRINT C  
70 FOR T = 1 TO 900: NEXT T  
80 NEXT C
```

As this program RUNs you will see the effect of the "overflow" from one byte to the next, but since the program does not allow the value 127 for C you won't get a solid line. Normally, of course, the HGR screens are controlled by HPLLOT, DRAW and XDRAW commands, all of which check the value in address 804 to see which colour to use. In the two programs above we have by-passed the colour byte and affected each HGR bit directly. In this way we have been able to create patterns

which would need very long HPLLOT directions if we were using Basic (because each dot and dash would need to be individually plotted).

To return to my previous point, the HGR addresses (hi-res 1 goes from \$2000-\$3FFF, and hi-res 2 goes from \$4000-\$5FFF) are actually only "semi-dedicated". That is to say, these same addresses can also be used as storage, for machine code programs and by very long Basic programs - as long as you don't want to use hi-res graphics at the same time. In this case the overflow effect we saw in the last program no longer occurs and each byte is read by the computer as it stands.

Lastly we come to the subject of PEEKs. As most of you will already know, PEEKing a specific address can have a direct effect. I'm thinking particularly of the command X = PEEK (-16336) which causes a single click from the built-in speaker. In this example the value of X is irrelevant, it is the PEEKing which causes the click.

Both the Applesoft Tutorial Manual and the Reference Manual list several useful PEEK operations, and I shall not bother to repeat those lists here. What the manuals do not make clear is the fact that there are several blocks of memory which are not

generally used by the computer and which can, therefore, be used to store information between programs.

The ideal location for this trick is the lower part of "page" 3, which starts at 768 (\$300):

```
10 T = 0
20 FOR N = 768 TO 777
30 T = T + 1
40 POKE N, T
50 NEXT N
```

Typing NEW clears the memory you were using for your program. Next type:

```
10 FOR N = 768 TO 777
20 T = PEEK (N)
30 PRINT T; " ";
40 NEXT N
```

On RUNNING the last program you will see the numbers 1-10 appear across the screen. Why? because the locations 768 to 777 are outside the area of memory that is "cleared" by the commands RUN and NEW. Thus the information you POKEd in the first program was still available for collection by PEEKing in the second program. Note that PEEKing at the


value of a particular byte will not alter the value of that byte.

I hope that this article has given you some idea of how the POKE, POKE and CALL functions operate. If it has, then you may now want to start using them in your own programs. But how? Where can you find all of the addresses that can be used by these commands?

*Windfall* itself frequently prints useful addresses as Appletips, and going through back copies will enable you to build up a small but useful set of addresses. Ultimately, however, if you want to use these functions to any great extent then you will need to buy Prof. Lueber's excellent reference book *What's Where in the Apple?*

This gives the "label", the address (in both hex code and decimal), and the function(s) of *most* of the locations inside an Apple II computer - including the Apple IIe. The current price is around £10, and it is well worth the outlay.

For those of you who are interested in DOS addresses and functions *Beneath Apple DOS* is another essential purchase.

Admittedly both books are somewhat confusing at first, due to their fairly technical nature, but then, like everything else in computing, "hands on" experience is always the best way to learn. 

# WILD-WORD

The cheaper alternative in word processing, without sacrificing efficiency. **Wild-word** is a powerful 40 column word processor for the Apple II and //e.

What you see is what you get. Over 20 format and editing commands with three modes give total editing flexibility, and have immediate effect on the screen. **Wild-word** features true SHIFT key operation using a simple clip-on modification. Lower case is generated in software using the graphics screen. Any character, including printer control codes, can be inserted into the text.

**Wild-word** also features a built-in mailing list and merge feature, and can store over 1000 names and addresses on each disc. It is also core-resident, leaving all drives free for text and mailing list data.

Header with automatic page numbering is available for print-out.

**Wild-word** is not copy protected, and is supplied complete with keyboard modification and full reference manual, for **£35 + VAT**.

Contact your local dealer or

**ELITE SOFTWARE COMPANY,**  
**2 Almorah Road, Heston,**  
**Middlesex TW5 9AD.**  
**Telephone: 01-572 0453**

# A turnkey can make life much simpler

THE Apple II is often used in a business environment by people who need to run professional software but who do not want to get involved in knowing about the details of the hardware and the operating system.

In these circumstances it can be very useful to be able to set up a disc so that all the user has to do to run a program is to put the relevant disc in the boot drive and switch on the computer.

For software which runs under Apple DOS, this presents no difficulties. All that is required is to save the program in question under the name given to the "greeting" program when the disc was initialised.

It is a good deal more difficult to create a turnkey system for a program which runs under the Apple CP/M operating system. Nevertheless it can be done with the aid of a disc editor such as the Zap program which is part of the Bag of Tricks suite produced by Quality Software.

Consider the problem of setting up a turnkey system to run Micropro's word processing program Wordstar on an Apple II equipped with a 16k RAMcard and connected to an Epson MX-80 printer by means of an Epson interface card.

There are two basic difficulties to overcome. Firstly the CP/M BIOS (Basic In/Out System) needs a small modification to enable the computer to communicate with the printer.

Secondly there is no standard CP/M facility to allow a program to run automatically when the computer is switched on.

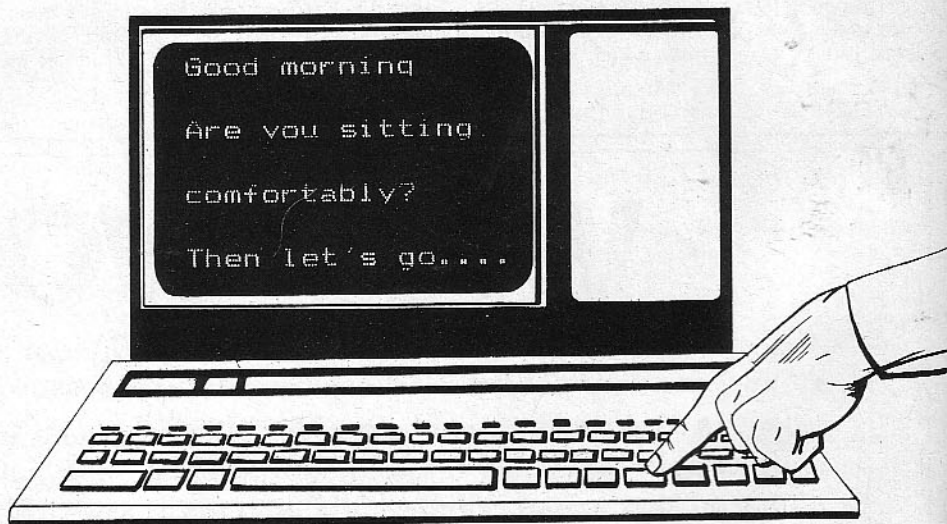
Normally the process of starting up Wordstar in this hardware environment involves the following steps:

- Insert program disc into drive A.
- Turn on power to computer and printer.
- Carry out the procedure set out on page 28 of Epson's interface card manual by typing:
  - DDT** [Return] Invoke the dynamic debugging tool
  - SDD2F** [Return] Use the DDT's 'Set' command
  - 31** [Return] Change 3E to 31
  - ^C** Leave the DDT
- Invoke Wordstar by typing:
  - WS** [Return]

Life would clearly be a great deal simpler if the last two steps could be eliminated.

To consider how this could be done involves understanding the process that CP/M goes through when a command is typed after the CCP (Console Command Processor) has displayed the normal 'A>' system prompt.

CP/M first calculates the length of the command and transfers the result of this calculation into the length byte in a command buffer. The command itself is then



transferred into the space available in the buffer immediately following the length byte.

If the command is one of those built in to the CP/M system, such as 'DIR', the CCP simply executes it. If the command is a transient program, with a filename ending in '.COM', the BDOS (Basic Disc Operating System) tries to find the file on the disc.

If it is successful a copy of the file is transferred into the TPA (Temporary Program Area) and is executed by the CCP. The length byte in the command buffer in either case is changed to zero.

When the Apple is first switched on the BIOS is loaded into memory and the CCP then examines the command buffer. If, as would normally be the case, it finds a zero in the length byte, the CCP will issue the 'A>' prompt.

However if it finds anything other than zero the CCP will treat whatever follows

that byte, up until the next zero, as a command to execute in the normal way.

The problem in our example can therefore be solved by modifying the BIOS on the disc itself, so that whenever a copy of it is loaded into memory the command buffer will contain a non-zero length byte followed by the letters 'WS' and a zero byte.

To effect this modification, a disc editor must be used to replace the zero in track 00 sector 0d byte 07 with any non-zero number. The contents of bytes 08-0A must then be replaced with 'D7 D3 00', being the Ascii values for 'WS' followed by a zero byte.

The next step is to eliminate the need for the third step of our example. To do this, the disc editor must be used to replace the contents of track 02 sector 08 byte 2F with '31'.

Having written these changes to the program disc the task is complete, and the

disc can be tested to check that it operates correctly. If it is now placed in drive A and the power switched on Wordstar should appear on the screen within a few seconds.

This general technique can be used to instruct CP/M to execute either a built-in command or a transient program.

If, for example, you wanted the disc directory to be displayed following system switch-on, you could place the command 'DIR' in the command buffer on the disc.

Alternatively you might want to have the computer 'autostart' a program written in MBasic by placing the words 'MBASIC BASPROG' in the buffer. This program 'BASPROG' might display a menu which allows the user to select one of a number of other programs simply by typing a number.

However, using a disc editor to modify each of a large number of different program discs is not always the best approach. If the user were to need a different program to run automatically on switch-on it would considerably simplify the procedure involved in modifying the disc if the changes could be effected with

the aid of an ordinary text editor.

One way of achieving such greater flexibility – setting up turnkey discs which can be modified without resorting to a disc editor – is to take advantage of the SUBMIT transient command file supplied on the CP/M system master disc.

The first step is to use PIP to transfer a copy of the SUBMIT.COM file on to every program disc. With the system master disc in drive A and the program disc in drive B use the command 'PIP B:=A:SUBMIT.COM'.

Next, use the disc editor to modify just one of the program discs by changing track 02 sector 08 byte 2F to '31'. Then alter track 00 sector 0D byte 07 to any non-zero number and replace bytes 08-16 inclusive with:

```
'D3 D5 C2 CD C9 D4 20 D4 D5 D2 CE  
CB C5 C9 00'
```

#### SUBMIT TURNKEY

Note that the Ascii code for the space between the two words must be '20' and not 'A0'.

Now leave the program disc in drive B and copy the modified CP/M to all the other program discs by invoking PIP

again, using the command 'PIP A:=B:/S'.

Finally, a standard text editor is required to create a file 'TURNKEY.SUB' on each disc. This file should contain the command you wish to be executed, and in the case of our Wordstar example it would simply consist of 'WS <cr>X'.

The procedure needed to set up a turnkey facility for CP/M programs is fairly straightforward, but it does seem a pity that a disc editor equipped to deal with CP/M discs is needed to put it into effect.

Perhaps Microsoft could be induced to put a simple utility program to do the same job in future revisions of the Soft-card package. Until they do, the technique described above will, I hope, be found helpful.

**THE disc editor referred to in this article was the subject of a review in the September 1982 issue of Windfall. Disc editors should be used with caution. Always carry out your modifications on a backup copy, never the original!**

# THE WILDCARD

Now available for the Apple //e

- ★ Copies 48K core resident programs.
- ★ No parameter list needed.
- ★ Copies most programs requiring RAM Card.
- ★ Wildcard copies may be copied with standard copy programs.
- ★ Copies even the bit copiers.
- ★ Programs copied are saved as standard binary files.
- ★ Works with any RAM card – plugs into any slot.

#### NEW FEATURES

- ★ Auto selects screens on //e.
- ★ Extracts foreign DOSes.
- ★ Patches to the Inspector.
- ★ Creates 'BRUN'able files.
- ★ Recovers listable, runnable Applesoft programs.

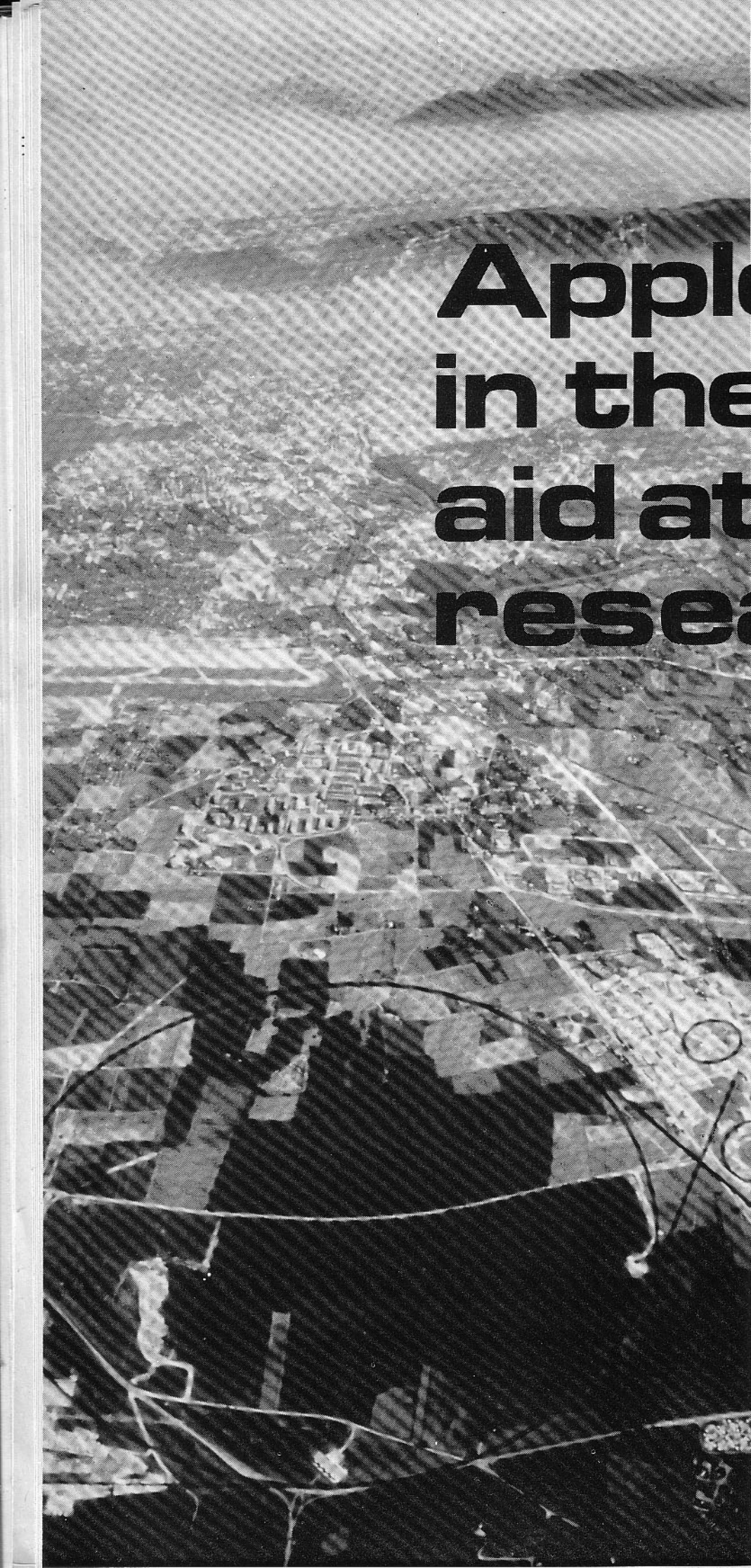
System requirements:

64K Apple + Disc Drive

**Price: £99.00 + VAT**

Available from your local dealer, or contact:

**ELITE SOFTWARE COMPANY,  
2 Almorah Road, Heston,  
Middlesex TW5 9AD.  
Telephone: 01-572 0453**



# Apples in the Alps aid atomic research

WITHIN sight of Mont Blanc and overlooked by the Jura mountains lies one of the most important nuclear research centres in the world. Known as CERN, it attracts some of the West's top physicists, including many Nobel prizewinners.

Encircling the site is a massive 7km tunnel, buried 40 metres underground, called the Super Proton Synchrotron, or SPS. This accelerates protons – the positively charged parts of an atom – and their highly elusive opposites, antiprotons, to energies up to 400,000 million electron-volts.

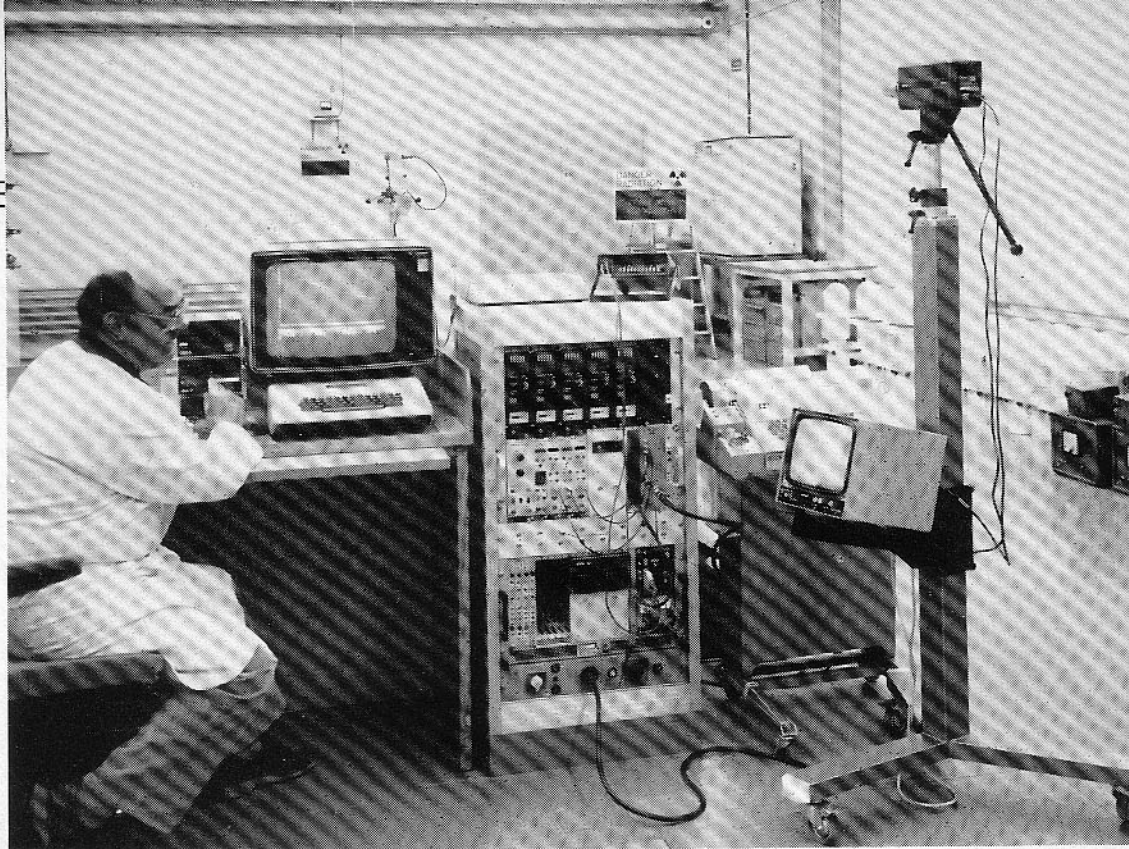
The SPS enables physicists to run nuclear particle experiments designed to unravel some of the secrets of inner space. Recently it helped them prove the existence of a subatomic particle so obscure that in the first experiment there were just five events that related to it out of a total of 1,000 million.

Its theoretical existence had been known for 20 years, but the discovery of the W-Boson was a world first for CERN and an historic and extremely important leap forward in the field of high energy particles.

SPS accelerator runs are usually conducted over a number of weeks, since it takes some time to stabilise the beams of subatomic particles that are used. Only after a run has finished can engineers and technicians go down into the ring to check the highly complex equipment and do the necessary maintenance and repairs.

— And this is where two of the 30 or so

*The CERN site spans the Franco-Swiss frontier (the path of the underground tunnel housing the SPS accelerator is shown in red on the cover of this month's Windfall). In the background is the lake of Geneva and the Alps.*



*The calibration chamber, where checks are carried out into the sensitivity of monitors to various forms of radiation. An Apple now has pride of place alongside the sophisticated monitoring equipment.*

Apples in use at CERN come into their own.

Both are used by the centre's radiation protection group, which is responsible for continually checking radiation levels, not only on the site itself but in the surrounding countryside. While these levels tend to be very low, CERN realises that with residential communities less than a mile down the road they cannot be too careful.

Maintenance engineers need to know whether there are any radiation risks around the ring before they commence work. While there are fixed monitors located in selected areas, the task of covering all 7km calls for purpose-built mobile equipment – a trolley developed by two members of the group, Brian Moy and Tony Shave. They call it K9 – "because of its similarity to a dog sniffing out hazards."

Information collected by K9 during its travels round the ring is transferred to the Apple, which has three disc drives, a black and white monitor and a printer. Data stored on discs can be processed to produce graphical or tabular representations of the radiation present at certain points within the ring, or simply printed out in hard copy form.

After each experimental run a large graph is produced showing which areas have the highest levels of radiation. This is divided up into six divisions or sextants, and copies of each relevant sextant are sent to all the control buildings located around the ring.

Having received the data from K9 the Apple can then send it over the CERN communications network to the radiation protection group's main minicomputer, a Norsk Data Nord-10.

While the connection to K9 is basically just a means of obtaining long term data storage, plus additional graphics facilities,

the ability to link the Apple to the mini opens up more ways in which it can be used, and this is being exploited to the full by the radiation protection team.

They have developed their own software to give the Apple access to the large databases stored on the Nord-10. This makes the Apple's facilities compatible with those available on the group's intelligent colour console, which costs 10 times the price of the whole Apple system.

The group's second Apple is being developed to help in the calibration of the ring's 1,100 monitors. Much of this work is still done manually, but the head of the

---

By  
**MARTIN RUSSELL**  
in Geneva

---

monitor calibration section, Mr Rasmus Raffnsøe, hopes the Apple will change all that.

Initially he plans to set up a database that will inform the person doing the calibration about the peculiarities of the monitor being used, as well as presenting him with data relating to previous calibrations.

The operator will then be helped, stage by stage, to perform the calibration operation by following instructions given by the Apple, many of which will be unique to the instrument being calibrated.

Mr Raffnsøe sees the adoption of an Apple-based system as a means of

processing information at a much faster rate than is possible at present.

But he would like to develop its potential even further, and wants to use it to control the motor that drives the trolley, telling it where to stop when it reaches a certain point along the track.

The automation of the calibration process would make the whole operation even safer than at present and would help reduce the levels of radiation to which the operators are subjected.

These levels, however, are already extremely low. It has been calculated that during the whole of 1982 the maximum amount of radiation a calibration operator could have been subjected to was less than one-sixtieth of the recommended maximum yearly dose – or the equivalent of smoking six cigarettes.

All the software that is needed will almost certainly be developed at CERN itself. It will include facilities to present the data in a much more appealing way than at present, and will make the fullest possible use of the Apple's graphics.

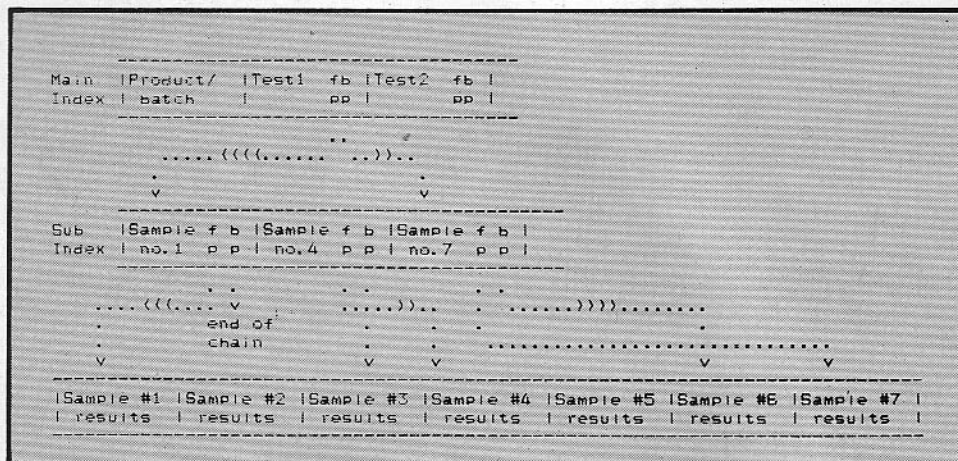
It seems certain that the Apples at CERN will be put to much greater use in the future, not only within the radiation protection group, but also in the six other divisions that employ them, in varying degrees, in their work.

Having talked to most of the CERN users of the machine, I discovered that many had future plans for their Apple-based system and most were very pleased with its peripheral support, although, to be fair, more than one wished that the graphics facilities were a little better.

However, for a micro to survive in a scientific organisation whose computing facilities are largely mini and mainframe based, and whose users are so specific in their demands, certainly takes some doing.







product + batch) is linked to another in a chain.

One end of the chain is linked to the primary key by means of a record number pointer so that the chain of results for a particular test can be followed by finding the product/batch entry in the index file and then following the pointers from entry to entry down the chain in the random test data file. See Example 1.

### Deletions

Deletions from the data file can be handled by following the chain until the record to be deleted is found. The pointer from this record (the one to be deleted) is then transferred to the previous record (the one next up the chain) and the chain is once again complete. See Example 2.

The record number of the deleted record is now added to a free space file.

### Additions

To add a new record to a chain, it is necessary only to consult the free space file for the record number of the first available spare record slot, to insert the new record at that place in the file and to remake the chain of pointers to include it in the chain. See Example 3.

The entry in the free space file is now deleted.

### Backward pointers

If the chain of test results is long and the most recent addition – the one at the

end of the chain – is wanted, then it may take a relatively long time to read down the chain. In this case, a chain of backward pointers will allow the chain to be read from the end back towards the beginning. See Example 4.

### Long chains

If the chains of test results are very long, then getting at a result somewhere near the middle of the chain can be rather slow, no matter whether the chain is read forward or backwards. If the test data records are in some defined order, then it becomes possible to predict approximately where in the chain the required record will be found.

The simplest way to do this is merely to change the index file key to include a sample number so that there are a number of index entries for each product/batch. Each index entry then points to a short chain of sample test results, but there are several index entries for each product/batch.

This method keeps the pointers relatively (!) straightforward but it means that the sample test results are not in one continuous chain. If this is a problem, then a subsidiary index file can be used.

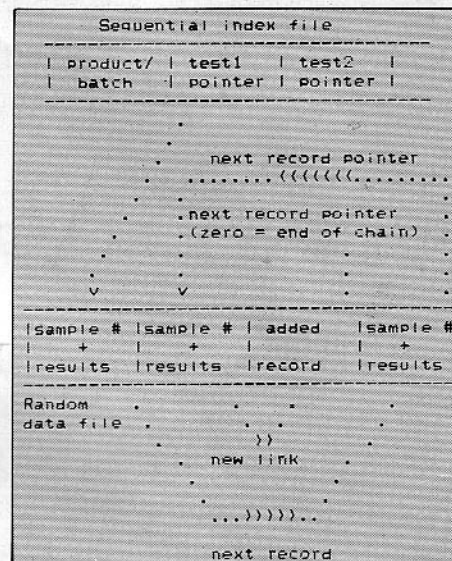
The main index file pointers point to the first and last records in part of a sub-index file. The results file must be organised in some defined sequence such as ascending sample order. See Example 5.

Thus, suppose we wish to access the results for sample No. 3 for test No. 1. First we read the main index until we reach the correct product/batch. The forward pointer then leads to a part of another file containing a chain of sub-index entries for the chain of sample results. The chain of sub-index entries is then read until the sample number we are interested in is reached or just passed. The backward pointers are then followed until the sample number record is found.

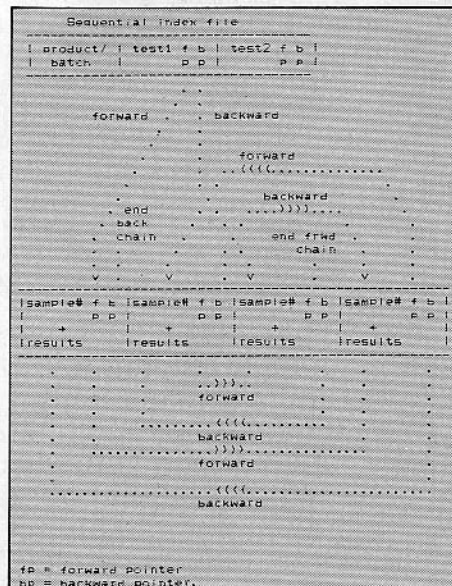
I've omitted the sample results chain pointers and the chain pointers within the sub-index file in the example for clarity, otherwise there would be pointers all over the diagram.

The sub-index entries need not refer to evenly spaced entries in the sample test results file. If part of this file is used more heavily than another, then the sub-index

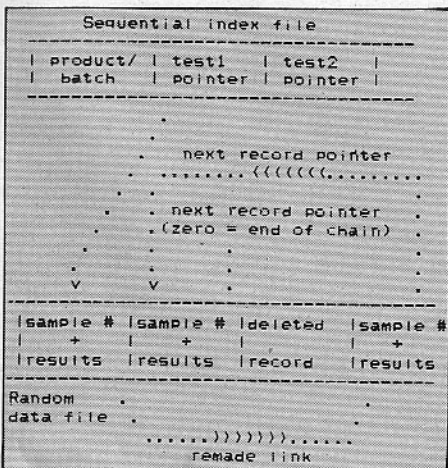
entries can refer to more closely spaced entries, and little used parts (old data for example) need have only a few widely



Example 3



Example 4



Example 2

# SPACE SAVER

spaced sub-index entries, and so on.

## The free-space file

When items are added to the sample test results file, empty slots in the body of the file should be used before items are added to the end of the file. When items are deleted, the empty slots must be kept track of. Both imply some sort of free-space file.

An ordinary sequential file could be used. Adding newly emptied slots to the file merely involves APPENDING them, but deleting slots from this free-space file is not so easy and involves rewriting the file each time. However if the free slot numbers are stored in a random access file, each record of which holds only one slot number, then a slot can be removed from the free-space file by writing a zero into the appropriate record.

Additions to the free-space file are

made by reading the file from its start until a record containing a zero is found. The newly freed slot number is then written to the file instead of this zero.

I think a sequential file using APPEND and POSITION could be used under Apple DOS instead. Maybe an expert could enlighten me.

## The updatable sequential file

A random access file, each record being only 1 byte long, can be used as an updatable sequential file. What's more, the record lengths can be as variable as we like. The snag, of course, is that we have to keep track of the length of each record ourselves.

However if the length of each record is included as the first field of the record, then the random file can be read sequentially until the record of interest is reached, a count of the number of bytes

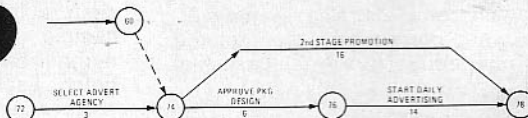
read being maintained as the file is read.

Having read the record, the file pointer is then repositioned to the beginning of that record and the record is rewritten. Of course, the new record must not be longer than the old record. So it is possible to have the space saving and variable length record virtues of a sequential file with the update-in-place virtue of a random access file.

This method has been named "Skip Sequential" by its inventors, Ecosoft of Indianapolis. I can't claim any credit for the idea (and I have doubts about its originality) but again I suspect that POSITION used with a sequential file might do just the same.

Disc space is precious and random files can be space gobblers if left to their own devices, so I hope that some of the above ideas will help use scarce space to good advantage. 🍏

# PERTMASTER



If your job is to plan, project and analyse the many interdependent activities contained in any successful project, why not take a closer look at PERTMASTER.

PERTMASTER is a software program designed to run on micro-computer systems. It is a proven planning tool, with over 3,000 copies sold to date, and is highly effective for the management of any project involving from 20 to 1,500 activities, precedence or arrow format.

PERTMASTER frees you from the constraints of mainframe computers, and puts control of your project at your fingertips. You can enter, update and analyse networks in minutes, and then run "What if" alternatives.

Most importantly PERTMASTER costs just £650 – a price you pay just once, unlike mainframe processing costs. A whole micro-computer system including PERTMASTER can be bought for under £4,000.

- ★ *Runs on IBM PC, ACT SIRIUS I, APPLE III or any CP/M computer.*
- ★ *User defined Calendars & Milestones*
- ★ *Produces time analysis and float calculations*
- ★ *Prints resource histograms, barcharts & S-curves*
- ★ *All commands are entered in plain English*

**ABTEX**  
Computer Systems Limited

58 Carden Place, Aberdeen, AB1 1UP  
Telephone (0224) - 647074

Contact us for the name of your nearest dealer.

## Keeping the competitive edge with an Apple

APPLICATIONS for Apple micro-computers in small businesses are rapidly becoming as numerous as the many different kinds of small businesses themselves.

Once the individual businessman has decided what it is he wants to try to do — and the prospect of computer use often helps sort out priorities and objectives even before installation — the versatility, ease of operation and inexpensiveness of the micro can provide him with a useful tool to do it.

Application of the micro can greatly simplify the working life of individuals while producing greater efficiency to cut costs or stimulate growth, and often enables the good new business idea to be put into practice.

Developing and maintaining a competitive edge must be a top priority for any small business, and Karl Emerson of Midland Commercial Insurance Services in Milton Keynes has done just that with the help of some progressive ideas and an Apple II.

Karl's application is of interest to potential Apple users not only in showing some of the things his system does but also in illustrating a businesslike approach to computerisation where the ideas come first and the Apple is used to implement them more efficiently.

This approach not only highlights advantages of the system, but also potential difficulties which all businesses would be wise to consider, perhaps with the help of a good Apple dealer or information technology centre, before the system is actually installed.

If this is done the full benefit of the Apple, or any other system for that matter, is more likely to be achieved. Like most business activities, proper planning and setting out of objectives based on sound knowledge is the key to successful computer application.

Insurance brokerage, a simple operation you might think, but not necessarily so at least in the case of the good insurance broker. Information quality, information volume and time (the basic parameters in computerisation, as it

---

By  
**BRIAN PEMBERTON**

---

happens) are the three all-important factors in the brokerage, all playing a major part in affecting the broker's decision on what to offer or recommend to a client.

The greater the amount of appropriate and accurate information the broker can use in the shortest time possible, the more effective will be his judgement and subsequent success.

Most of us have visited an insurance broker at some time or another in search of a quote for motor insurance and we know how long that can take and how varied can be the quotes from different sources.

Dealing with some 25 or 30 insurance companies, a broker will have to search through a full range of rating guides before he can come up with what he considers to be the best quote. This takes a lot of time, and with the large quantity of enquiries and many different types of insurance that the broker has to deal with a great amount of time-consuming paperwork is normally involved.

As Karl Emerson told us: "Efficiency is essential in any brokerage, and we decided to install a computer system at Midland Commercial to give us accurate useable information very quickly and also to maintain our customer records, a policy renewal system and our accounts — and very importantly, to fully integrate all of these functions."

Motor insurance is the area that Karl has looked at first, partly because of the amount of time-consuming work involved for the marginal return, and partly because a good service in this area was seen as a way of introducing Midland Commercial's expertise in the many other insurance areas to a large number of potential clients. As most new customers initially seek a motor insurance quotation, the speedy and accurate service makes a lasting impression.

Finding the right policy involves checking numerous variables. With the system introduced by Midland Commercial and

the software written by Karl himself, the practical application is simple and enables any member of staff who knows how to operate the keyboard and understands the various office codes especially developed to provide quickly a choice of quotes for any enquiry.

A driver's details and requirements are keyed in to the computer as the questions are asked and within seconds a printout offering a comprehensive list of deals available can be shown to the client. If the client wants to pay by deferred payments the computer will also provide information on the various facilities offered by each company.

The same program also enables service to continue after the client has his insurance. The system provides a renewal warning — often when the insurance company involved has failed to send the client a reminder — together with an updated quotation showing any increased bonuses available or better deals possible.

It all adds up to efficiency and a better service to customers, and although only covering a part of the envisaged fully integrated system, this program has been instrumental in bringing about a situation in which Midland Commercial is planning a massive expansion with the opening of further branch offices.

So how can other businesses make sure they get the most out of their new business tools?

As we have seen, the aim in this case has been to operate a fully integrated system covering several inter-related functions. Software has been of very great importance here — in addition to Visicalc, Visidex and Appewriter, Karl Emerson has written a special program for the motor insurance and renewal system, and his experience has led him to the opinion that one needs to think very carefully about software availability.

If the computer system is to do the job that you want it to do, rather than you do the job the computer wants to do or is capable of doing, then the right software must be available — either off the shelf, custom written or self-written.

It may prove impossible, for instance, to integrate off the shelf programs and if programming ability is limited or non-

\* Brian Pemberton is a PR and marketing consultant with Peregrine Atlantic.

# APPLICATIONS

existent, difficult to write the programs yourself or expensive to get a software house to do it. And then you may find that the hardware is not quite right.

The experience of Midland Commercial illustrates the need for a thorough understanding of the job you want the computer to do, and a good deal of user education so that the right combination of software and hardware is chosen right from the start, followed by the maximum and optimum usage of the equipment, perhaps through more easily understood programming training for the business user.

It is, of course, up to the computer industry to provide and promote "education" to help ensure full and proper use of their products. And this is happening. Milton Keynes, in fact, is a good example. The new Information Technology Exchange there is proving a welcome facility for potential users to try out various systems.

Anything that helps the supplier specify the best system for the job and enables the user to get more out of it must be good.

Computerisation is not without difficulties and pitfalls, but approached carefully and with thought it can certainly help the innovative business. Midland Commercial Insurance Services for one have come a long way with their Apple — let's hope you bump into them and not their clients!

*This printout shows comprehensive insurance availability (only the five best quotes having been requested from the computer) for a 23-year-old female driver of a six-year-old Austin Maxi 1750, based on owner driver only paying a £50 excess.*

*As can be seen, the cost variance between just five companies can be*

*significant. Figures on the left indicate the cost and the initials to the right are the broker's code for the companies concerned.*

*Below this are the deferred payments schemes offered by the top five companies with a selection of others offering a deferred payments plan.*

NOTE—PREMIUM INCLUDES LEGAL EXPENSES

SMITH

AGE-23:SEX-F:OCCUP CODE-XX:COVER-C:DRIVERS-1:CAR MAKE-S:CAR MODEL-S  
AGE OF CAR-6:NCD-S:EXCESS-50

197.81-GA

216.5-IT

229.57-PROV

233.44-CU

279.44-ES

GA HIRE CAR PLAN:-197.81

INSTALLMENT SCHEME-SMITH

GA-DEPOSIT 23 + 5 INSTS OF 35.16

GRE-DEPOSIT 70.62 + 4 INSTS OF 63.59

PROV-DEPOSIT 48.31 + 10 INSTS OF 19.21

SA-DEPOSIT 59.64 + 10 INSTS OF 28.32

NU-DEPOSIT 62.19 + 4 INSTS OF 59.19

CU-DEPOSIT 49.66 + 4 INSTS OF 46.66

L&G-DEPOSIT 71.27 + 4 INSTS OF 68.27

AVON-DEPOSIT 75 + 4 INSTS OF 73.44

PHX -DEPOSIT 41.04 + 9 INSTS OF 38.04

SENTRY-SIX MONTH POLICY AT 227.1

ES-DEPOSIT 58.28 + 4 INSTS OF 55.98

ALL-DEPOSIT 76.9 + 4 INSTS OF 72.9

ROYAL-DEPOSIT 84.51 + 4 INSTS OF 84.05

APPLE II  
COMPATIBLE

## APPLE USERS

# BACK UP VISICALC AND ALL YOUR VISIS QUICKLY AND SIMPLY

Copy II Plus is a versatile software back-up system, capable of backing up all visis, as well as most other protected software.

**Its fast** — only 45 seconds for Fast Copy, less than three minutes for Bit Copy.

**Its simple** — menu driven for ease of use with full instructions on backing up dozens of popular programs.

**Its comprehensive** — it includes all the file handling and DOS utilities you will ever need.

**Its priced right** — at around half the cost of similar competing products.

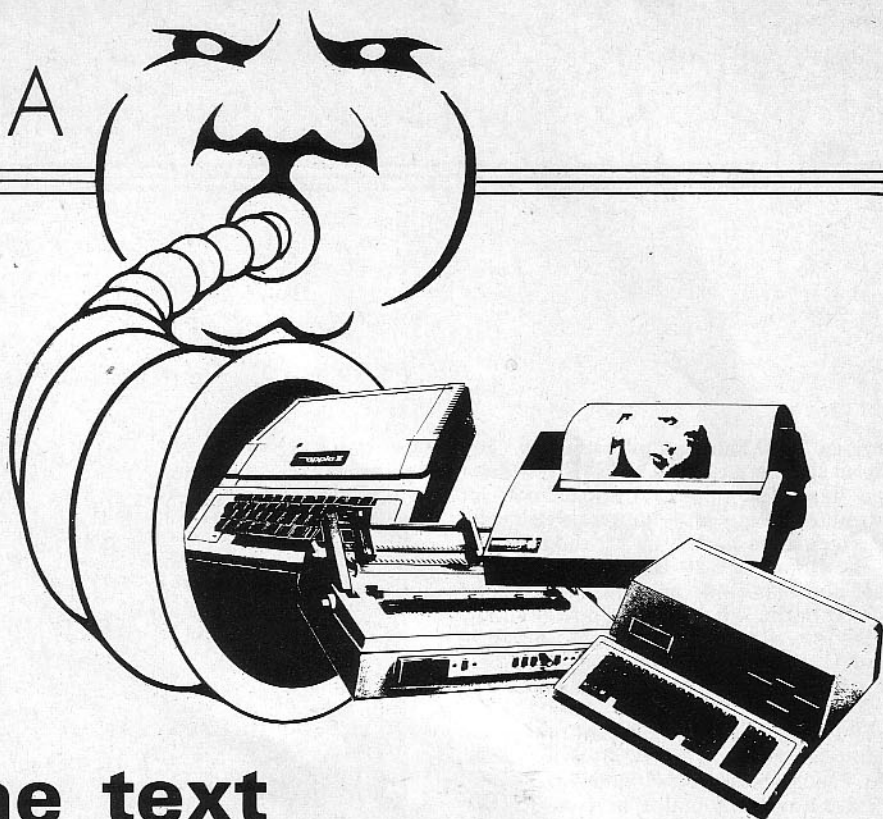
**Copy II Plus — only £35 + V.A.T.**

**COPY II PLUS  
COPY  
COPY  
COPY**

**ORCHARD  
SOFTWARE**

Send cash with order, or quote your Access or Diners card number to:

Orchard Software, 17, Wigmore Street, London W.1. Telephone 01-580-5816. Dealer enquiries welcome.



## Removing the text editor's blindfold

A SCREEN editor customised to exploit the features of the Apple IIe with Z80 card and standard 80 column text option is VuEditor/Apple.

Distributor Vuman claims the package eliminates "blindfold" text editing using ED.COM and lets a user see what is happening to the file being edited as it is keyed in. It can handle text files of any size.

The editor has been "hand-coded" to ensure fast screen display of updates and cursor movements so that the display keeps up with typing at all times.

Facilities include immediate mode operations with 19 lines around the current cursor position continuously displayed and updated, repeatable command macros including string search, insert and replace, in-memory text block move, copy and erase, line printer listings from within the edit, secondary input and output file support, and the ability to erase disc files and display disc directories from within the edit.

The package costs £49.95 to licensed users of Microsoft CP/M.

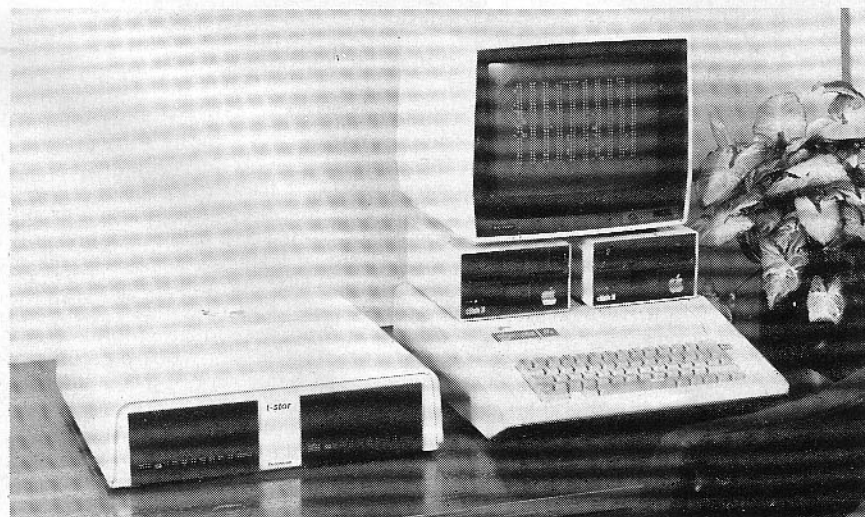
Tel: 061-273 5315.

## Sweetening the pill

FROM January next year pharmacists will be required to print the labels of all dispensations they make. Microskil, in conjunction with a group of pharmacists, has developed a program that meets these dispensing needs.

It is called Microskil-PDS (Pharmacy Dispensing System) and holds stock control details of 1,700 preparations with full reporting facilities.

The range of label instructions covers all requirements, claims distributor ABM Computers, and a label can be prepared and the pharmacist's stock record



**A DUAL Winchester drive subsystem for the Apple II, the T-stor, is a free-standing unit with a range of capacities from 5mbyte with a single drive to 30 mbyte with dual 15mbyte drives.**

**It is supplied with dual disc operating software to run under DOS 3.3 or Pascal. Distributor X-Data says CP/M will be available soon.**

**The T-stor uses drives from the IMI 5000H series, featuring shock absorbent mountings and hard-plated media, and uses the Xebec s1410 controller. Tel: 0753 72331.**

automatically updated, in less than 20 seconds.

Features include comprehensive error-trapping routines with informative screen messages, automatic disc diagnostics and copying, access to any record within two seconds and immediate notification of low stock.

Labels are compiled from a set of phrases which are accessed with only two key strokes, and several instruction codes are left unallocated to allow individual tailoring of the package.

The program costs £500 and runs on a II Plus or IIe with two disc drives and a printer (although two printers are recom-

mended — one for labels and one for reports.)

Tel: 021-707 1822.

## Kind to floppy drives

RECENTLY released in the UK is a 320k solid state disc emulator for the Apple II or III.

Manufacturer Axlon claims the

Ramdisc 320 boosts both memory and access speeds and is up to 50 times faster than hard disc drives. A spokesman for distributors Pete and Pam said two full discs could be sorted in 15 seconds.

The Ramdisc 320 is designed to save wear and tear on floppy drives and has no moving parts. It has its own power supply and a three hour battery back up. As a result, it draws no power from the Apple and retains data when the Apple is turned off.

The solid state memory add-on system is compatible with DOS 3.3, SOS, CP/M, and Pascal versions 1.1 and 4.0.

Its memory functions as two 35, two 40, or as one 80 track floppy drive. Up to four units can be connected to give more than 1.3mbytes memory.

The unit comes with a plug-in, slot-independent interface card and includes a specially designed operating system and software for diagnostics and fast-load copy routines. It costs £699.

Tel: 0706 227011.

## More mini Winnies

A RANGE of 3½in Winchester drives has been launched by ICE and is available from the company's world-wide dealer network.

They are supplied as three stand-alone units, incorporating a single 3½in Winchester, a single 3½in Winchester plus a 3½in floppy, and dual 3½in Winchesters.

Three different drives are available with formatted disc capacities of 5, 10 and 15 mbytes. Since the physical size of the drives is so small, the single Winchester can be housed in a unit measuring 4 x 5½ x 1½in — no bigger than two packets of cigarettes.

ICE claims the drives will completely change traditional micro systems based around the floppy disc.

Instead of a normal maximum of 1 mbyte on dual floppy drives, standard micro systems will now offer up to 15 mbytes on a single Winchester. The company says they will "emulate the cost of a floppy unit."

Tel: 07842 47271.

## Surveying information

A PROGRAM that cross reads information from one disc to another has been developed for use in the surveyors offices of building companies.

Surveyor uses six discs containing



ICE 3½in Winchester drive

different elements of data. They interlink during the running time and the program tells the user which disc contains the information he is looking for.

The system stores as separate files sub contractor names and addresses, sub contractor orders, specifications of works to be done, job section numbering and costing, exempted and tax liable sub contractors sorting, non-construction industry sub contractors sorting, job valuations of works completed and contra-charges against sub-contracts.

These files are sorted by the program which selects data from each section and tells you which disc contains the data required.

Tel: 0734 479314.

## The error trapper . . .

A RECENT arrival in the UK is Qbase, a database package that corrects errors before they happen by refusing to accept many types of erroneous entries.

It beeps its warning and then points out the error, stopping, for example, if important information such as a job number is omitted. Using sophisticated data checking features normally found on larger systems, Qbase allows check points to be built in to assure the accuracy of data entered.

It works in two ways. The Apple can be instructed to accept only certain data in certain fields, or a user can request that data entered is checked for possible errors before it is accepted.

Distributors Pete and Pam say the files and forms created are accurate, accessible and efficient and additions or

alterations can be made at will.

With Qbase a report can be prepared in a variety of ways. It is formatted in columns, allowing up to 200 spaces in a single line, data may be sorted three ways, subtotals can be produced for each group of items, and summary totals can be printed at the end.

Mailing labels can also be prepared in any format, and manufacturer Applied Software Technology says extra spacing is deleted and exact duplicates are automatically removed.

Qbase is described as 'upwardly compatible' with Versaform.

Tel: 0706 227011.

## Laboratory packages

SCIENTIFIC software packages to enhance Adalab, the laboratory Apple II interface card, have been released through Heyden Datasystems.

Vidisampler is a real-time data acquisition package which runs on a 48k Apple with at least one disc drive. It can be used to sample automatically up to four voltage inputs in a background interrupt driven mode, while data collection is going on.

Adalab's timer controls the sampling frequency, and the number of samples collected on any Vidisampler channel is limited only by available RAM memory.

Vidimemory gives extended memory options of 16k, 24k or 128k RAM for Vidi-chart, Adalab's high resolution graphics software, and Vidisampler.

Stripcharter is a four pen chart recorder used for displaying, recording and documenting large data sets such as chromatograms and spectra.

It plots from one to four curves as a continuous chart of any length on a monitor display or printer.

Also released is Lab Data Manager 1, a general purpose software package for acquisition, storage, display and output of lab data.

Tel: 01-203 5171.

## Complete CAD system

IT IS now possible to interface the Robocom Bit Stik interactive graphics system with the Strobe 100 Graphics Plotter.

Robocom has produced special software to do the job, and it is available from distributors Data Efficiency for £95.

Data Efficiency says the resulting combination is a complete CAD system for the first time user for less than £2,500.

incorporating an Apple IIe, two disc drives, software, interface, plotter and Bit Stik.

Tel: 0442 60155.

## Apple presents . . .

WITH Slideshow Programmer an Apple II or IIe becomes a slide presenter using high resolution graphics "pictures" rather than 35mm slides.

The slideshows are controlled manually or run automatically, and can be used for business and sales presentations, education and training courses and for automatic information screenings to the public.

Features include creation of text slides using a mixture of fonts and colours, different display times for different types of slides, use of "reveal" sequences, remote control using available devices

such as game paddles or the Robocom Bitstick, and the creation of self-contained slide show libraries.

The program uses a combination of Fast DOS and pre-loading of slides to provide speedy disc access and instant switching to the next slide. It can be integrated with business graphics. The system is available from Systems Consultation Services for £48.

Tel: 025 587-385.

## 128k RAM card

A 128k RAM card for the Apple III has been released in the US by Legend Industries. An SOS driver is available separately from dealers.

The card also works in the Apple II, II plus and IIe.

Tel: (0101) 313-674 0953.

## Schools package

A SCHOOLS administration and reporting package, Scholar III, has been developed by Deverill Computer Services in consultation with a public school, to meet the needs of a bursar's department in independent schools.

It handles up to 1,500 pupil records and 2,000 names and addresses.

The package can generate more than 300 reports, including fee notes, outstanding balances, grant/subsidy claims, fee analysis and pupil lists by form, tutor, house or any user-defined field (such as BUPA, squash, weekly boarder.)

DCS says it has designed the system to minimise the volume of information that needs to be entered.

"If all the Rifle Club members go on a

# ICE II IN CIRCUIT EMULATOR

Now you can build your own microprocessor based project using the same CPU as the Apple II/IIe. This card will enable you to test and develop 6502 based projects, locate hardware faults and develop software from the familiar environment of the Apple. After completion of the design cycle your program can be burned into EPROM to allow your microprocessor project to function independently.

The ICE II enables the powerful facilities of the Apple to be used in developing your system. The package includes supporting software for debugging and testing, an extended monitor program for working within the emulation memory, plus the added advantage of program trace facilities.

Spend your time developing your project ideas not trying to find hardware faults, let the ICE II do that.

- Examine generated interrupts and resets
- Test your specialist input and output devices from the Apple.
- Fault find existing 6502 based equipment from the Apple.
- Make full use of the screen, disks and Keyboard during the project development
- Printer spooler software provided to save time.
- Full memory test capability for the project RAM.

- Breakpoint on specified address or address range.
- Breakpoint on specific memory operation.
- Trace or single step over any area of code at variable speed; execute subroutines outside this area at full speed.
- Trigger an oscilloscope or logic analyser on any memory operation.
- Investigate professional and protected software. You know a particular subroutine is

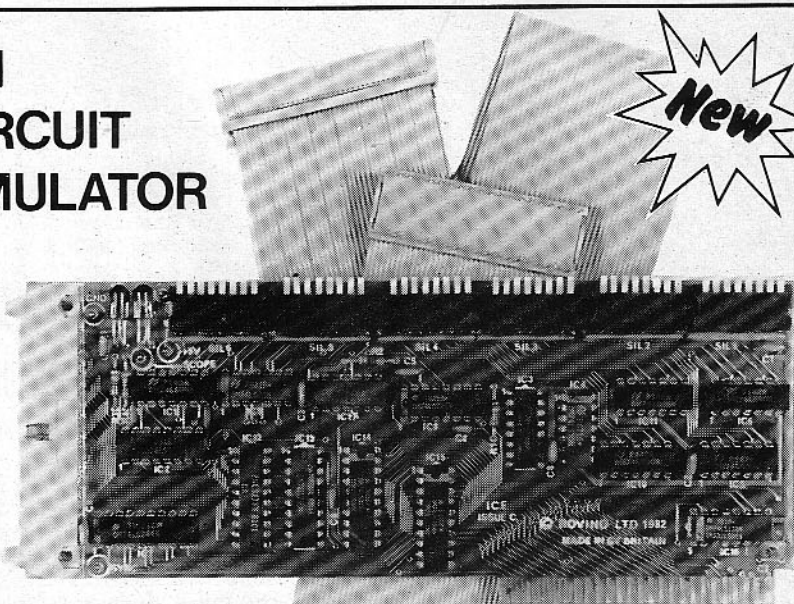
- called or a specific location written to, but from where? Find out with the ICE II.
- Handle interrupts in basic.
- Coming soon Prototype 6502 microprocessor boards for experimentation.

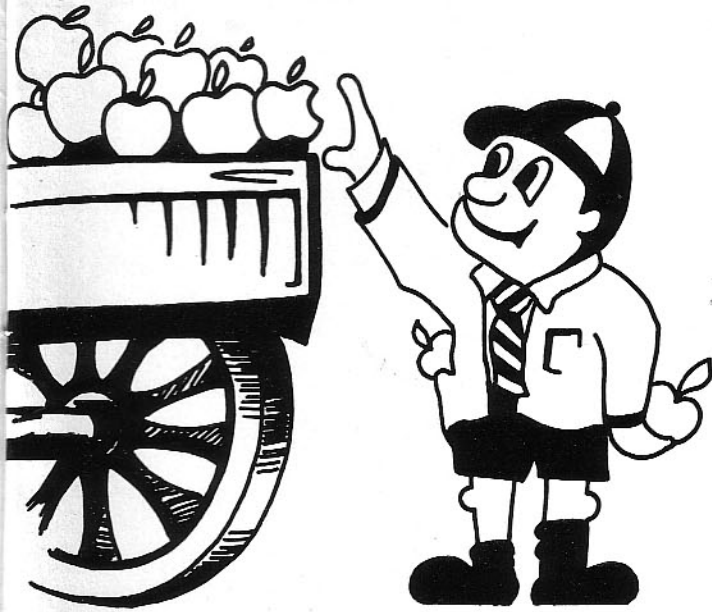
This professional in-circuit emulator, (already in use in industry) allows easy development of microprocessor circuits at a fraction of the cost of dedicated systems. Building a microprocessor with the ICE II is simple and exciting. It comes complete with its own menu driven supporting software.

PRICE:  
£195 + VAT  
Available  
direct from:

# rovino

35 James Street West  
Bath Avon BA1 2BT.  
Tel. 0225 310916





Monthly review of  
Apple in education

## Pilot Animation belies its glossy packaging

By  
**TONY  
COOK**

Department  
of Biology  
New University  
of Ulster

IN previous *Windfall* articles, various facilities in the Apple Pilot language have been discussed. It is a simple, flexible language which allows the production of teaching packages by people with little expertise in computing. A more sophisticated facility is provided by Pilot Animation, which has to be purchased as a separate disc, supplied on the understanding that it may be copied for "back-up use only".

If we are to stick to this rigidly then the program is useless for teaching since teaching programs are often widely disseminated on an informal basis. Indeed, since the Animator program itself (a relatively uncomplicated 100 lines) has to be copied on to all lesson discs where it is to be used, one could be accused of breaking the copyright in normal use.

The Animation instructions are so poorly written that it is worthwhile treating the topic here in some detail. In essence, the program enables you to store a file of characters and cursor commands. The animation is therefore of text, which can be made to look like pictures using the Character Set Editor.

If you want to move the character 'A' around the screen the Animator program is run from the Lesson Editor. It asks for the file name under which the animation sequences will be stored and then the name of the character set and the number of different sequences.

You might think that all that is now required is to give the character 'A' and where to move it. Not so. There is no erasing process built into the program, so that if you give it 'A' and 'move forward' then you get a line of As. To animate a single letter, therefore, it must be hedged about with the appropriate spaces to eliminate its predecessors.

This is something which is not made clear in the

handbook, and the more complex the shapes to be animated, and the more complex the movement across the screen relative to static characters, the more difficult it becomes to eliminate the previously printed characters without eliminating the background.

In most cases it would be simpler to move the characters around by using the cursor control commands available in Pilot itself rather than by using the Pilot Animation.

By way of illustration the programs in *Figures I* and *II* show the animation of a simple equation. The program in *Figure I* uses the Animator program from disc and that in *Figure II* is written from scratch - a DIY version. The steps in the animation are:

1. AMP + 2p<sub>i</sub> →
2. AMP + 2P<sub>i</sub> → ↓
3. AMP + 2P<sub>i</sub> → AMP
4. AMP → A■P → ADP
5. AMP + 2P<sub>i</sub> → ADP
6. ADP → A■P → ATP
7. Final display is AMP + 2P<sub>i</sub> → ATP

The unconventional characters are provided from an augmented Ascii character set called ATP with the % changed to -, the # to →, the ' to ; and the ! to ■.

Both programs consist of similar components. Part I is devoted to allocating storage for strings (d; dimension), Part II assigns values to these strings

*The first part of this article appeared in the June issue of Windfall.*



(c.; compute ), Part III draws the equation and moves the AMP to the right hand side (g:es, clear screen, tx.; load new character set) while Part IV moves Pi over to the right twice and changes AMP first to ADP and then to ATP (w.; wait). Part V asks if the animated portion of the program is to be repeated (as.; accept a single key response; jy.; jump if the previous match was made correctly). It is identical for the two programs and is, therefore, only given in Figure 1.

The starting point for all animation sequences is determined using the CHR(30) command (= g\$) which sends the text cursor to any position on the screen using a combination of symbols as coordinates. Both programs repeatedly print the same character sequences and the programming could be tidied up using the Pilot version of a FOR-NEXT loop (that is, \* start/th:\$a\$/Z=Z+1 /j(Z < 4): start). This makes for a tidy printout but for incredibly slow animation since Pilot takes a long time to perform the necessary evaluations. In both programs the types of changes taking place must be mapped out clearly from the outset.

In the Animator version (Figure 1) there are eight "sequences". The file (called "Trialfile") in which they are stored is created by the Animator program and once written is inaccessible from Pilot - so get it right the first time because that is the last time you can even look at it.

No sign of the file appears on any menu in the Pilot lesson or Author discs, and the whole disc must be copied in order to transfer the stored sequences. The sequences used are shown below. Space is represented by "." and end of line by "/".

- S1 - .AMP moved up (8 bytes)
- S2 - .AMP/... moved right (13 bytes)
- S3 - .../.AMP moved down (13 bytes)
- S4 - .Pi moved right (7 bytes)
- S5 - .../.Pi moved right (11 bytes)
- S6 - .Pi/... moved up (11 bytes)

Sequences S7 and S8 consist of three frames which change places with no movement.

- S7 - .AMP to .A■P to .ADP (24 bytes)
- S8 - .ADP to .A■P to .ATP (24 bytes)

The length of each sequence is given by the Animator program and is required when dimensioning the string variables into which they will be read.

The Animator version (Figure 1) first allocates space to the strings to hold these sequences (part I), reads the file (f\$ = trialfile) and assigns the sequences to the strings (part II). The various parts are then moved around by typing these sequences using th: (Type & Hang).

The procedure is fairly simple provided you stick to your original intentions. It is not possible to build in modifications which require different sequences. The movements proceed fairly rapidly and since one is trying to convey information rather than to impress, there will often be merit in slowing the process down.

In the Animator version no changes can be made within a sequence though it is easy enough to build in pauses between sequences using the Wait (w) command.

The second, DIY, version of the animation may look longer but in fact is shorter since all the sequence construction is built into the program. The sequences are built up in part IIB out of the characters and cursor controls (Up, Down & Back) in part IIA. The sequences are then positioned on the screen in a similar way to that used for the Animator version.

The basic information required for DIY animation is the same as for the Animator version and the result is much the same, though the Animator version seems to have a smoother quality about it. There are, however, enormous advantages to DIY animation.

The first is flexibility. All the information is in the program and available to be played with if your first attempt is imperfect. The second is in running time. The Animator version has to load all the files, the character set and the program from disc before the first character is displayed.

For this program there is a blank screen for 36 seconds while the disc is read. This would be a boring pause in any teaching program. The DIY version on the other hand starts after 18 seconds of disc whirring representing a 50 per cent reduction in the time taken to load this part of the program.

Thirdly, the DIY version gives greater control of the speed with which movements occur. It takes longer to evaluate two strings than one so that although (\$a\$ \$b\$) is the same as (\$a3\$) in the program in Figure 11, the first will run slightly slower than the second.

Finally, the greatest advantage is in cost. DIY animation is free and does not really take more time than using the expensive Pilot Animation disc.

In conclusion let me say that the special effects in Apple Pilot are brilliant - easy to use, flexible and, most importantly, appropriate for computer-aided-learning in a wide variety of subjects.

As for Pilot Animation, frankly it is not worth the posh packaging and the write protected, Special Delivery Software disc that it comes on.

Sorry Apple - but it is easier, cheaper and more fun to do it yourself.

## PRINT OF LESSON TRIAL

```
r:ATP Animator animation using file
:'trialfile'.
d:f$(9)
c:f$="trialfile"
d:s1$(8)
d:s2$(13)
d:s3$(13)
d:s4$(7)
d:s5$(11)
d:s6$(11)
d:s7$(24)
d:s8$(24)
d:g$(1)
```

Part I

```
fix:8,f$
fi:1,s1$
fi:2,s2$
fi:3,s3$
fi:4,s4$
fi:5,s5$
fi:6,s6$
fi:7,s7$
fi:8,s8$
c:g$=chr(30)
```

Part II

```
tx:atp
*anim
g:es
c:z=0
t:$g$ $+AMP + 2P' %#
th:$g$ "*" $s1$
th:$s2$ $s2$ $s2$ $s2$ $s2$ $s2$ $s2$ $s2$
:$s2$ $s2$ $s2$ $s2$ $s2$ $s2$ $s2$ $s2$
:$s2$ $s2$ $s2$ $s3$ $s3$
w:2
*fos
th:$g$ +,$s4$ $s5$ $s5$ $s5$ $s5$ $s5$ $s5$
:$s5$ $s5$ $s5$ $s6$ $s6$
th(z=0):$g$ 3+ $s7$
th(z=1):$g$ 3+ $s8$
c:z=z+1
j(z=1):fos
```

Part III

Part IV

```
t:$g$ 'Would you like to see it again?'
t: (Y or N)
as:
m:y!Y
jy:anim
e:
```

Part V

## PRINT OF LESSON TEST

```
r:ATP D.I.Y. Animation
d:g$(1)
d:b$(1)
d:b3$(3)
d:b4$(4)
d:d$(1)
d:u$(1)
d:a$(4)
d:c$(4)
d:p$(4)
d:x$(4)
d:y$(4)
d:z$(4)
d:a1$(13)
d:a2$(13)
d:a3$(7)
d:p1$(13)
d:p2$(13)
d:p3$(7)
```

Part I

```
c:g$=chr(30)
c:b$=chr(8)
c:b3$=b$!!b$!!b$
c:b4$=b$!!b$!!b$!!b$
c:d$=chr(10)
c:u$=chr(31)
c:a$=" AMP"
c:c$=" "
c:p$=" P' "
c:x$=" A!P"
c:y$=" ADP"
c:z$=" ATP"
c:a1$=c$!!b4$!!u$!!a$
c:a2$=c$!!d$!!b4$!!a$
c:a3$=a$!!b3$
c:p1$=c$!!d$!!b4$!!p$
c:p2$=c$!!u$!!b4$!!p$
c:p3$=p$!!b3$
```

Part IIA

Part IIB

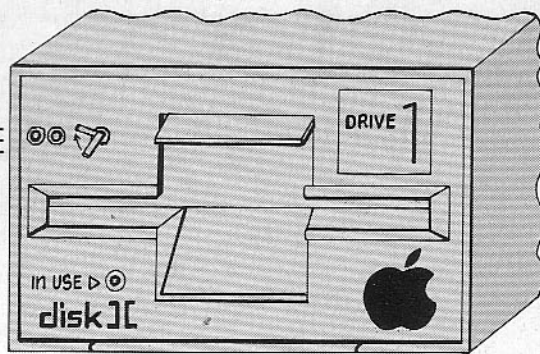
```
tx:atp
*anim
g:es
c:z=0
t:$g$ $+AMP + 2P' %#
th:$g$ "*" $a$ $b4$ $a1$ $b3$
th:$a3$ $a3$ $a3$ $a3$ $a3$ $a3$ $a3$
:$a3$ $a3$ $a3$ $a3$ $a3$ $a3$ $a3$
th:$a3$ $a$ $b4$ $a2$ $b4$ $a2$
w:2
*fos
th:$g$ ), $p$ $b4$ $p1$ $b3$
th:$p3$ $p3$ $p3$ $p3$ $p3$ $p3$ $p3$
:$p3$ $p3$ $p$ $b4$ $p2$ $b4$ $p2$ $b4$
:$c$
th:$g$ 2+ $x$
th(z=0):$b4$ $y$
th(z=1):$b4$ $z$
c:z=z+1
j(z=1):fos
e:
```

Part III

Part IV

Figure I

Figure II




---

## By P.F. WILSON

---

UNDER normal factory fresh conditions, the Apple II disc drives are one sided drives which utilise the underside of the disc to record data. To cut costs of data storage, some time ago I removed the write-protect switch on my No. 2 drive to enable me to use the reverse (which is really the top) of my discs, thereby cutting my costs in half.

This I later refined a little by fitting a switch to my No. 1 drive to enable the selective use of the reverse side of my discs by being able to write protect the reverse of discs and/or the fronts if I used the normal method of using the sticky tabs.

The obvious problems of constantly applying and removing the sticky tabs need not be reiterated, and after wiping two discs full of information clean by mistake I decided there must be a better method.

Disregarding the correct advice on not using the top side on which to record data, the following is a vast improvement on the above mentioned method giving total control over the read/write head. It was developed in conjunction with Glynn Davis, of Chorley.

Using the method described you will be able to:

- Write to either side of the disc without having to resort to cutting a slot in both sides (in fact with this method you can write to a disc with no slots in it).
- Write-protect either side of a disc, totally, even if there is a slot in the disc.
- Be able to read at any time either side of the disc (obviously only the side which has the read/write head).
- Know at a glance which mode you are in – either total write protect or total write enable.

**WARNING:** This modification to your drive will almost inevitably nullify your warranty, so it is not recommended if your disc drive is under one year old.

To carry out the modification you will need for each disc drive:

- 2 LEDs (Light Emitting Diodes), one red, one green (preferably the ones with

long leads and a hood on the front for easy mounting).

- 1 miniature double pole double throw switch.
- 2 220 ohm resistors.
- A soldering iron.
- Solder.
- An assortment of drills to make three holes in the front cover of the drive.
- A Phillips screwdriver.
- About half an hour or so.

Procedure: Remove the four screws from the bottom of the disc drive, and remove the cover by sliding it to the rear.

Identify the following components on the disc drive:

(a) The pin in the rear right corner of the PCB marked TP3 (Test Point 3).

(b) The 470 uf 6.3v capacitor in the front left. On the PCB (Printed Circuit Board) itself you will notice that one end is marked with a + sign and the other end a - sign.

(c) The write protect/enable microswitch on the front left hand side just below where the disc is inserted.

If you cannot positively identify the above positions go no further until you can.

Having properly identified all the above points, remove the small Phillips screws from the write protect/enable microswitch and unsolder the brown and black wires.

It is a good idea to replace the microswitch in place as it will help to seat the discs correctly (don't forget to re-adjust it for correct seating of the discs).

Solder a six inch length of wire onto both the brown and black wires you have just removed from the microswitch and cover the join using insulating sleeving or tape.

Carefully route the wires up to the

upper half of the disc drive, taking care not to impede the drive door entrance by going round the drive entrance, and locking the wires behind the metal chassis.

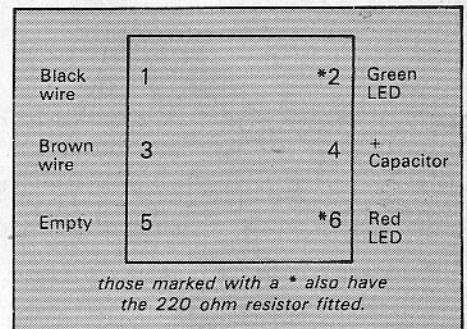
Taking great care on the positioning, drill the three holes in the front panel in the upper left hand side allowing enough room for the washers and nuts on the rear of the LEDs and switch.

You might consider removing the front plastic cover to avoid getting any swarf from the drilling into the works.

Looking from the rear, solder the two 220 ohm resistors onto the upper right and lower right tags of the switch (positions 2 and 6 on the diagram).

Also solder the brown wire to the centre tag (position 3) and the black wire to the upper left tag (position 1). Shorten the wires as necessary to leave a small neat loop.

Insert the two LEDs in position and solder the anodes (normally the red wire)



View from rear of switch

onto the switch as follows: From the red LED to the lower right (position 6). From the green LED to the upper right (position 2).

Next solder a short two inch wire from the positive side of the 470 uf capacitor to the centre right tag (position 4).

Take both the cathode leads from the LEDs (normally black) and solder them onto point TP3.

Carefully fit the switch onto the front panel and tidy up any long or loose lengths of wire that you may have left around. Also check that you can insert a disc into the drive door without any hindrance.

One final check that you have connected up as per instruction, replace the drive cover from the rear, boot DOS and try out the system.

With the red light showing, the discs will be write protected, and with the green light you will be able to write to either side of a disc.

Incidentally, the lights will not come on until you boot DOS.

If you have two drives, proceed as above with the second drive, and Bob's your uncle – suddenly you have total control.



# A switch in time saves two... or how to use both sides of your discs