A Database Publication

apple user

Vol. 4 No. 9 September 1984 £1

-the new name for Windfall

MACINTOSH

First field report . . . help for software developers . . . pioneering game reviewed

An Apple helps shot police hero walk again

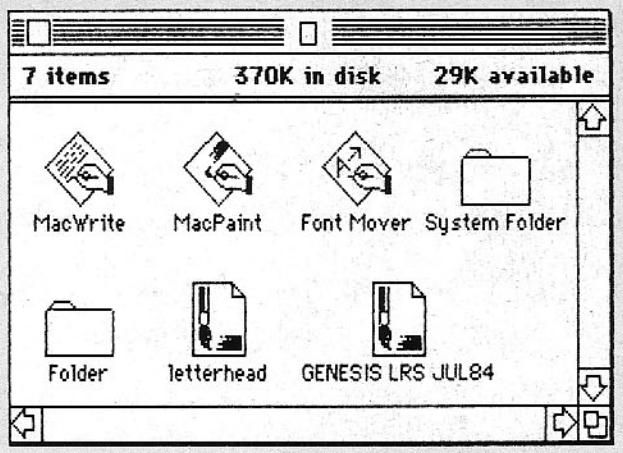
Teach Visicalc to think for itself

Mini-Pilot: full program listing inside

REVIEWS

Graphpak • Doublestuff Format-80 Enhanced

apple user
Volume 4
Number 9
September 1984



Apple User Managing Editor **Derek Meakin** *Features Editor* **Cliff McKnight** *Technical Editor* **Max Parrott**
Production Editor **Peter Glover** *Design* **Heather Sheldrick** *Advertisement Manager* **John Riding**
Sales **Peter Nowell** *Marketing Manager* **Susan Lipman** *Editor-in-Chief Database Publications* **Peter Brameld**

Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising) 061-480 0171 (Subscriptions) Telex: 667664 SHARET G. Prestel: 614568383.

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

Trade distribution in the UK and overseas: Contact Steve Fletcher, Circulation Manager of Database Publications at the above address or telephone him on 0293 26945.

Writing for Apple User: Articles and programs relating to the Apple are welcome. Articles should preferably be typed or computer-printed, using double spacing. Unsolicited manuscripts, discs etc, should be accompanied by a self addressed stamped envelope, otherwise their return cannot be guaranteed. Unless agreed, material is accepted on an all rights basis.

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)

Data on tap

ACCESS to more than 20 million articles, books, computer programs, conference papers and technical reports covering virtually every subject from accounting to zoology is now possible for Apple users.

For a once-only fee of £25, and $24 an hour thereafter, owners with a compatible modem and a telephone can access the 50 gigabytes of storage in the American mainframe of Oxford-based Knowledge Index.

A side-shoot and simpler version of the 12-year-old Dialogue information system, KI is available only at night – from 6pm to 5am – and from 2pm to 5am Saturday and Sunday.

Computer dealers are being offered free access to KI to demonstrate the communication capabilities of their micros.

● Details of *Apple User's* own communications package – which will allow you to access the Knowledge Index database – is on Page 49.

Now Apple sets its sights on Russia

THE signs are that Apple is poised to invade the Soviet Union following news that the Western nations have eased the embargo on selling micros to the Eastern bloc.

However the company is playing its cards close to the chest at the moment. No one at Apple's Californian HQ was prepared to be quoted on the company's plans.

Nor were they at Apple UK or at the Apple Europe office in Holland.

The export rules are laid down by a committee of the Nato allies. This is currently dominated by the Americans, who see every micro as potential war material.

As a result the committee still bans the export of micros over 128k and vetoes the sale of computers with networking capability behind the Iron Curtain. The decision is unpopular. Manufacturers are already scrutinising the new rules to see if there is a way round them.

And in June the American industry newsletter AdaData said the US may be better advised to encourage sales to ensure that Russian defence computers – known to be 10 years behind the times – do not accidentally trigger a nuclear exchange.

Meanwhile British manufacturers, led by Acorn and Sinclair, have already dipped a toe into the water.

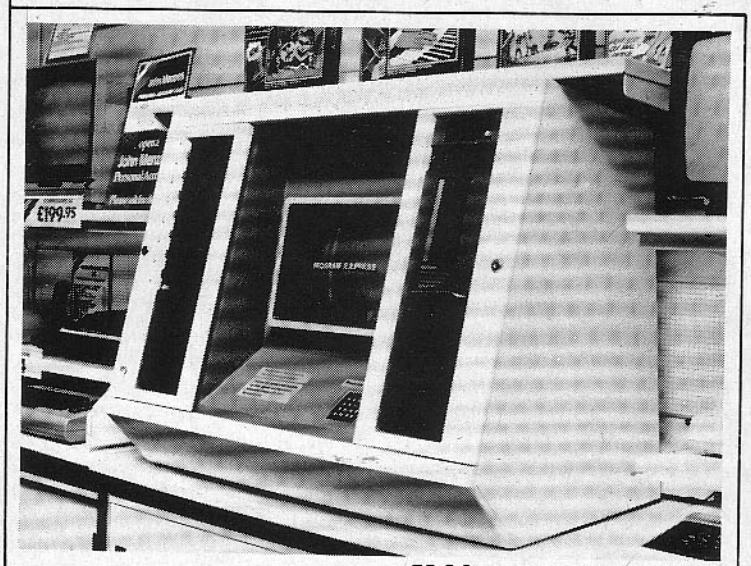
Senior managers from both companies recently visited the Eastern bloc for initial talks on supplying the lucrative 8 bit market, estimated at $10 million over the next five years.

And Acorn is about to sign up a company in mainland China for manufacture and distribution of their BBC Micro.

With these developments in the background, it is unlikely that the US can hold back the flood of would-be exporters much longer.

The pressure has redoubled with the recent Soviet decision to put micros into its schools – a natural Apple market.

So when the new regulations – now in the final draft stage – are published later this year, the chances are they may be already obsolete.



EDOS . . . software revolution

Software by phone is here

APPLE users will be able to take advantage of a revolutionary new development which could transform the face of software sales in the UK from next month.

It is claimed that Electronic Distribution of Software (EDOS) will herald the long-awaited dawn for commercial downloading of software over the telephone.

Developed by Inventory Transfer Systems of Palo Alto, California, EDOS involves in-store computer/programmer machines to allow the dealer to reproduce any program selected by the customers.

It is the first time anywhere in the world that the system has been launched.

All Apple users will have to do is take along their floppy discs and make their choice from the 1,000 programs contained on the machine's 40mbyte disc. These will be constantly updated by telephone link to a computer in Scotland.

However Apple owners may have some difficulty locating one of the new machines in the early days. For only 20 of them are to be made available to selected dealers across the UK in October.

But by the end of 1985 it is forecast that up to 400 machines will be on-site in Britain, the figure rising to 2,500 within three years of launch.

Meanwhile the distributors still have a few teething problems to sort out before the first machine is installed.

They have yet to reach agreement with the software houses – 90 per cent of which are claimed to be ready to throw in their lot with the new system – over the question of independent auditing.

Prices cut

RAPID Terminals, Visicorp's UK distributor, has cut its prices on business packages – some by 60 per cent.

Visischedule is cut from £245 to £125 for the Apple IIe version. Visifile, previously £212, is now £110.

But Visicalc, down to £96 in its IBM version, has not been cut for Apple machines.

Visiword Plus is down from £285 to £175, while Visitrend Plot – originally £245 – is now £85 for both Apple and IBM.

Sales were a record

APPLE has announced worldwide net sales of $442.1 million for the quarter ending June 29.

This broke all sales records at Apple, representing a 58 per cent increase over the same quarter of 1983, and a 41 per cent increase over the preceding quarter.

Net sales for the first nine months of the financial year exceeded $1,000 million, 46 per cent higher than last year and greater than all of 1983.

This marks the third consecutive quarter of profit increases for Apple. However net income was lower than the $24.2 million or $.40 per share earned during the third fiscal quarter of last year.

Apple president John Sculley said this showed the market's acceptance of Apple's two new mainstream products, Macintosh and the Apple IIc.

More changes planned

MAJOR changes are being planned in Apple UK's marketing strategy following the arrival of the new high-powered team headed by general manager David Hancock.

A major sales drive is being planned, with the emphasis on big-name high street outlets.

An Apple watcher said: "It seems that from now on there is going to be much more hard selling, rather than glossy marketing".

As part of the new look, a number of senior personnel have now left the company.

In the run-up to the pre-Christmas sales drive Apple are believed to be preparing for a massive increase in dealer support throughout the country.

Apple invades the

CORONATION Street will never be quite the same again as a result of Mike Baldwin introducing an Apple IIe into his factory in the series.

Rumour has it that the scriptwriters have now been locked in the Rover's Return and forced to drink Newton and Ridley's best bitter until they can come up with a story line to justify the computer's arrival.

The same dubious source has leaked to *Apple User* the news that they have already considered the following ideas:

Ivy Tilsley and her fellow workers are to call on Arthur Scargill to lead their protest against the new technology.

However it is thought the NUM are likely to object to this on the grounds that their leader will have difficulty making himself heard over the rantings of Vera Duckworth. (No wonder Jack is always in the pub.)

Meanwhile Emily Bishop, convinced that the Apple will make her redundant, has taken to drinking copious amounts of QC sherry in the office and then streaking down Rosamonde Street in the company of Mavis Riley.

Councillor Alf Roberts is sell software along with th beans in the corner shop whi Rita Fairclough has got her han caught in a jar of Uncle Joe Mint Balls in all the excitemen

Ken Barlow, dynamic edit of the Weatherfield Recorder, desperate to write a story abo the new technology but he still struggling to spe "teknowlogi".

Deirdre, on the other hand, concentrating on teachin Tracey (now presumably age about 55) how to use a kni

Removable hard disc

A REMOVABLE hard disc system launched by Intec is said to have solved all the problems which have plagued similar systems in recent years.

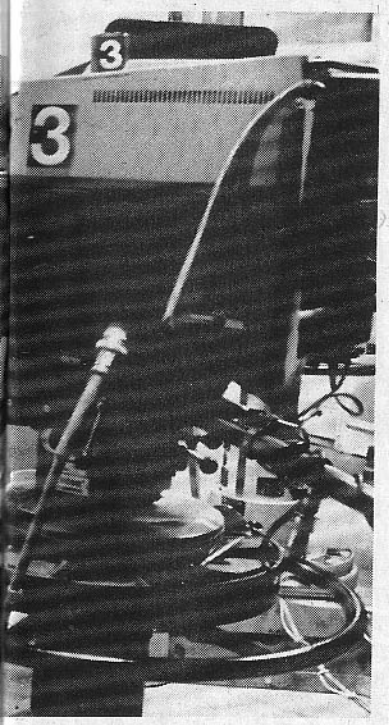
Intec's managing director John Groves said: "Previously it was found that the problems involved in interchanging discs from one drive to another compounded those already acknowledged to be a danger with fixed Winchesters".

However Intec is now convinced that it is onto a winner by combining its own drive with a new removable hard disc manufactured in the UK by Newbury Data under licence to US based DMA Systems.

Called the Intec 505, it incorporates several new software features to replace functions previously carried out by hardware.

A closed loop track followin system which refers constantl to actual track positions on th disc in guiding its placemen is said to prevent previou difficulties with disc positionin and locating information.

It overcomes environmenta contamination by tucking th delicate heads away into sealed chamber while the driv

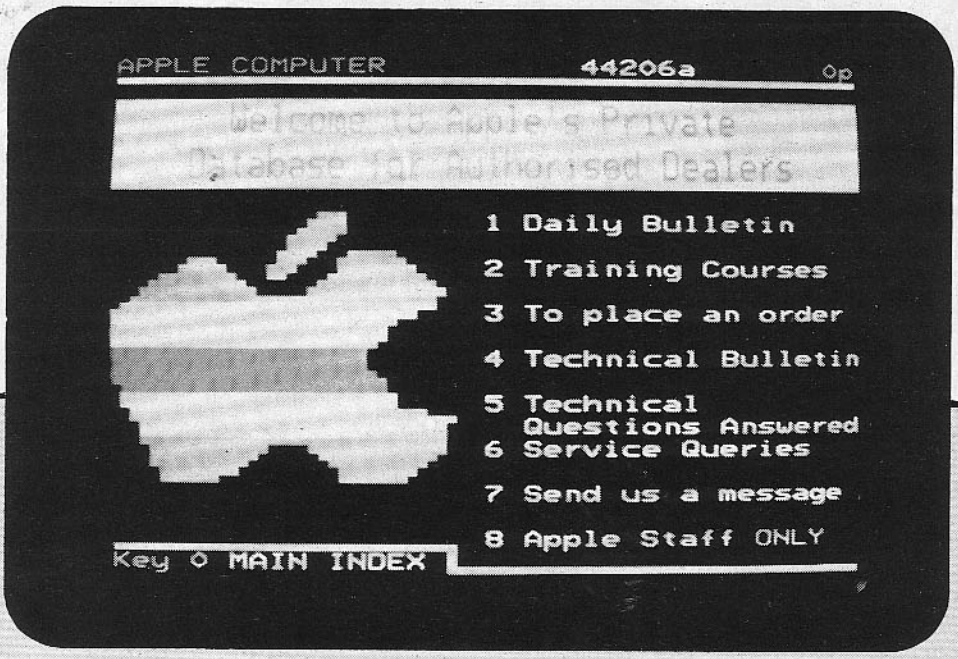


Street

and fork, while Billy Walker has been eyeing the Apple in the hope of flogging it to pay off his debts in Jersey.

All this notwithstanding, a meeting has allegedly been called by Bet Lynch during which she has been asked to burn her bra and turn the whole show into a disaster of Towering Inferno proportions.

Mind you, anything would be preferable to seeing Mike Baldwin – as really happened in front of the millions of viewers – using the Apple to play Snack-Attack . . .



Viewdata service for dealers

APPLE UK has introduced a private viewdata service on Prestel for authorised dealers.

The system will offer varied information, including a daily bulletin and technical data.

Designed to be interactive, dealers will be able to place orders and send enquiries to different departments within Apple, who can reply using the same system.

Apple chose Prestel for its flexibility and low costs – Prestel access is at local telephone call rate – and because most of their dealers already have the facility.

Using Owl software, Apple will edit the service daily.

Action over copyright

APPLE has started two separate actions in the UK against vendors of alleged Apple compatible computers – the Unitron 2200 and the Base 64A.

Apple alleges both these products are copied from the Apple II range and that they infringe Apple copyright in some of its operating programs and manuals, and in the case of the Base 64A their copyright case design also. Terms of settlement are being negotiated with both defendants, who will undertake not to sell these products and will deliver up allegedly infringing stock to Apple for destruction.

As regards one defendant who has not yet accepted terms of settlement, the proceedings are continuing.

Sirtel UK, exclusive UK distributor of the MPF 11 micro which, like the Unitron 2200 and the Base 64A, comes from the Far East, has now gone into liquidation.

Sirtel gave undertakings last year that they would not dispose of their stock of MPF 11 RMS pending a full hearing of the proceedings.

It is understood the micros have now been sold by the liquidator as scrap.

A solicitor's letter has also been sent to Copam Ltd, which is selling the Base 64A micro in the UK, saying that unless they stop selling the product and deliver up their stock to Apple proceedings will be started.

Meanwhile in Australia Apple has finally got its injunction against sales of the look-alike Wombat micro.

This was earlier refused on the grounds that computer programs were not literary works within the meaning of the Australian copyright laws.

ends problems

is not operating and while it powers down and up.

During powering up of the drive – before the heads are exposed – an air purge cycle is automatically completed to further avoid contamination risk.

Unlike other Winchester systems this one does not rely solely on disc rotation for continuous air circulation. Instead a separate full time circulation system operates to cut out risk from office or factory smoke.

The Intec 505 is already in use in several UK locations. A typical one is the chemical laboratory of a top UK soft drinks company.

It is being used there to store long term product analysis data for constant reference and comparison with other test results.

Mactips

TO help keep Macintosh owners and prospective owners up to date Thames Valley Systems is publishing a newsletter called MacNews – a mixture of hints, tips and product information.

Apple donate systems

APPLE Computer gave computer systems to four charities at a lunch held at Maxim's de Paris in London.

The Duke of Westminster and show business personality Frankie Vaughan were there to accept the donations on behalf of the St John Ambulance Brigade and Radio Link, a Derby hospital radio station.

The Rev Gordon Barritt, president of the Conference of the Methodist Church in Britain, accepted an Apple IIe computer system on behalf of the West London Day Centre, a Methodist group for helping homeless and rootless people.

The final donation was to Ridgeway School, a Bedfordshire school for disabled children. Apple donated a Macintosh, requested by Ridgeway for its ease of use for disabled people and the opportunities MacPaint presents for artistic pupils who cannot use normal drawing materials.

● Picture shows (left to right) Frankie Vaughan, Michael Spring, marketing services manager at Apple (UK) Ltd, the Duke of Westminster, Mr Derek Baker, deputy headmaster of Ridgeway School and the Rev Gordon Barritt.

One finger is the trigger

AT 21 years of age, Graham Thody is so severely handicapped that he can only use one finger to type.

But thanks to his recently acquired Apple IIe, he is able to hold down a worthwhile job helping run the office of the Disabled Housing Trust, a charity run by his father in Burgess Hill, Sussex.

With the aid of his Apple, Graham handles much of the Trust's correspondence. His machine has been specially adapted for the purpose by using a Mac Apple disc.

This has nothing to do with the Macintosh, but is a word processing disc written with the disabled in mind.

Explains Graham: "There are lots of ways of using a Mac Apple disc – it can be used by suck and puff or by using two switches.

"When using these inputs, half of the screen is like a grid with all the alphabet on it. So when you want to get a letter O you would look for it on the grid – you might go along four and up six to get the letter.

"The only input I use is the keyboard with five more keys. One of the five holds the shift down, another holds the Ctrl down, while yet another holds Rept down.

"Another key is for the wordlist, which means that when there's a long word I can call it up without putting every letter on the keyboard.

"The last key is the Mac command key which makes some of the other keys do things – S is for save, P is for printer, G is for file. This, like wordlist, only works on the Mac disc".

Icons in court...

THE word icon, made into a popular buzz word by Apple's mouse-driven marvels, Lisa and Macintosh, has surfaced in a legal dispute between two other companies.

A new local area network from Torus Systems was launched recently as the "Icon System".

Cambridge-based Eicon Research promptly objected to the use of the word, on the grounds that it would confuse customers for its sound-alike Eicon network.

But Torus declined to withdraw the name. So Eicon served a High Court writ on them.

Said John Hartley, Eicon's managing director: "They launched Icon as a trading style, and this conflicts with our application for a registered trade mark.

"That is different from using icon as a descriptive word".

He added that people had been ringing up Eicon under the impression they were talking to Torus.

Eicon – Engineering Innovation CONsultants – has been trading since 1979.

Torus has until September 9 to present their side of the story to the court.

Speedy MacPaint

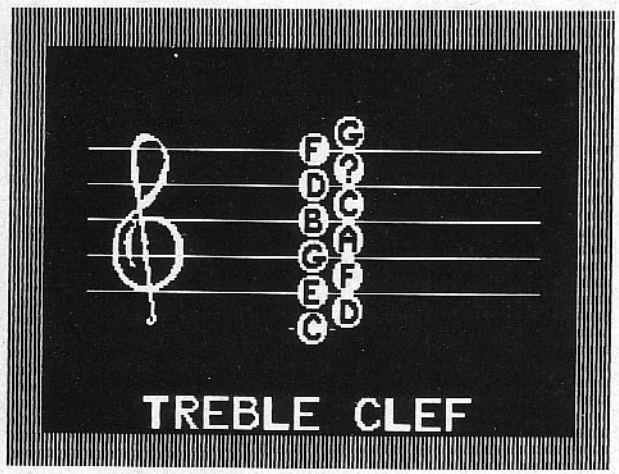
THE cover of this month's Apple User provides a remarkable example of the speed with which electronic drawing can be done on Macintosh using MacPaint.

The artist was Apple UK's newly-appointed regional manager, Roy Stringer. Under the watchful eyes of our editorial staff it took him less than five minutes to create his dramatic picture of a barren, crater-strewn asteroid hurtling through space.

Starting from a simple circle, Roy used the spray-can to add the initial texture. The entire background was poured in from the paint pot.

From a small patch of pencil dot stars, the copy option was used to spread the twinkling galaxy across the background.

Finally the craters were added using the pencil and zoom facility.



Early Games Music

... A TUTOR IN SHOW TIME

MAKING sounds on the Apple is not too difficult, and if you've got a package like Electric Duet you'll know that making music is within the machine's basic capabilities. In fact, if you've ever heard David Ellis perform, you'll know the heights to which you can aspire on the Apple.

It is not surprising, therefore, that the current boom in educational software has produced a music tutor for the Apple.

Early Games Music is written by John Paulson, an active composer, conductor and music teacher. He also happens to be president of Counterpoint Software which publishes the package.

According to the manual, it is designed for children aged four to 12, or the musical novice of any age. The price is about £25.

The package consists of four modules. In Perform mode the numeric keys of the Apple produce musical notes and the accompanying display can be a keyboard, treble clef or bass clef.

As a key is pressed the corresponding note is played and displayed on the keyboard or staff. In addition to performing, you c n also record your efforts and 'ay them back.

Guido's Quiz also allows a choice of keyboard, treble or bass clefs. The letters of the scale appear, then one by one they are replaced by a question mark.

Each time, you must identify the note by pressing the appropriate letter key. A correct response results in the removal of the letter from the display. When all the letters have been removed, the quiz continues with just the question mark displayed on the keyboard or staff.

The Melody Tutor is, by its own admission, little more than a Simon variant, using simple tunes. The keyboard format and numeric keys are employed, with the tune built up one note at a time in typical Simon fashion.

There are six tunes on the disc, like Jingle Bells and This Old Man, but you can also use tunes which you have recorded using the keyboard in the Perform mode.

Finally, Kaleidoscore allows you to perform, record or playback a tune, but adds a coloured shape to the screen each time a note is played.

All four modules are linked via a picture menu. Pressing ESC from any module recalls the picture menu, which cycles through all the options.

When you see the picture you want, pressing any key runs it. There is also a text menu which can be accessed using Ctrl-I while in the picture menu.

Obviously some thought has gone into this package. For example, in the sections where the learner is being tested, there is always direct correspondence between the display and the key-press required.

If numeric keys are being used for input, the keyboard notes are labelled with the numbers. If treble or bass clef is being used, the appropriate letter keys are used for input.

In this way the learner doesn't have to remember a translation code of which keys produce which notes.

However the package suffers from a basic problem – the minimum time between notes is a bit long. This, combined with a not unreasonable fixed note length, means all the tunes have a slow, chunky feel to them.

In Guido's Quiz this is not serious. The aim is to teach basic notation and the speed is reasonable. Furthermore, the easy section can be skipped using Ctrl-J.

In Perform mode the problem is more acute, and when using the Melody Tutor it is a real drawback.

Remembering the sequence of notes could be fun, but having to enter them at dirge speed puts the emphasis on memory, not music.

Graphically the package is quite strong, having been created with the aid of Penguin's Complete Graphics System II and Special Effects.

However the irony is that it is probably the time needed to produce decent graphics which gives the long inter-note interval.

Overall the quiz is a very useful teaching aid and the kaleidoscope has novelty value. However the children I got to try the package preferred performing, recording and playing back on a Casio VL1, complete with adjustable rhythm accompaniment which in total costs little more than this package.

If I had time, I'd work on interfacing the Casio to the package because the combination would be quite good. It's clear now why David Ellis needs so much *more* than just the Apple.

Title: Early Games Music
Author: John Paulson
Publisher: Conterpoint Software
Requirements: Non stated

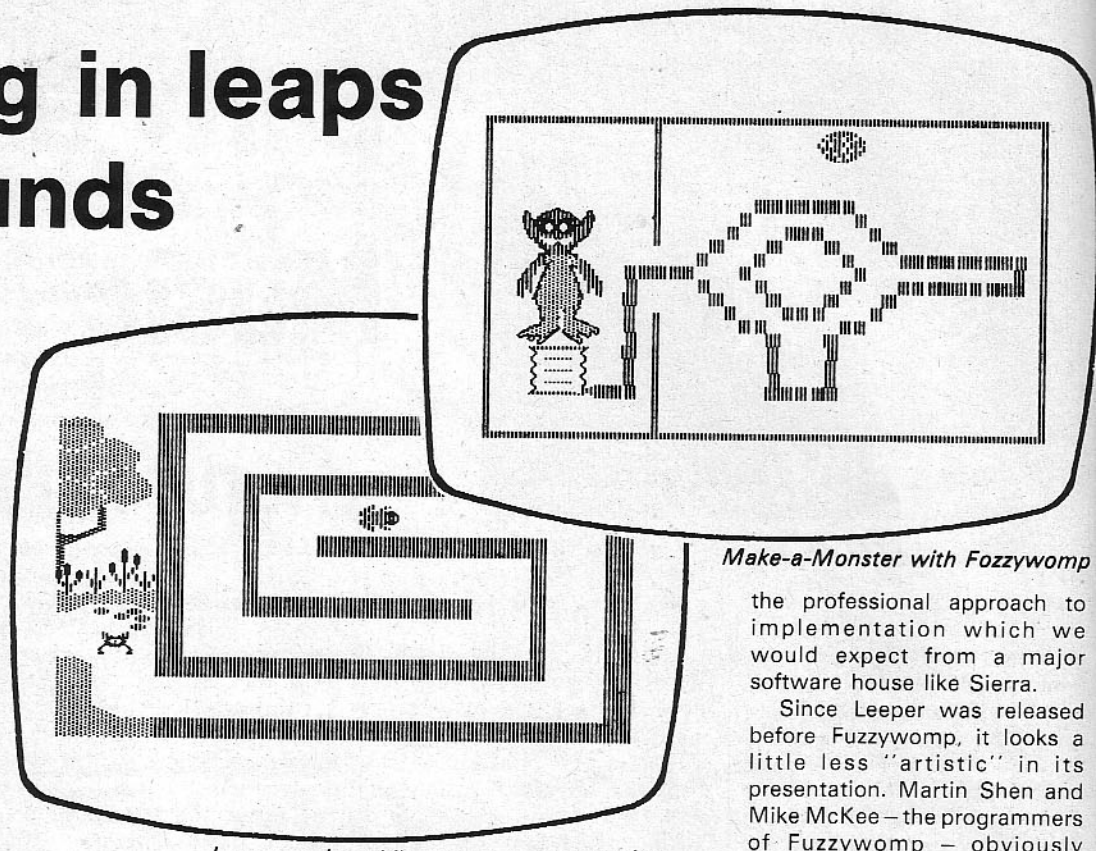
Learning in leaps and bounds

I CAN always tell when a game is popular with our children. It's not that I have uncanny sixth sense or anything like that – they wake me up at 6.30am to ask if they can play it.

Just recently Lorna, our 2½ year-old, has been disturbing my beauty sleep asking for Learning with Leeper.

Leeper is a frog who at least has the sense to avoid busy main roads and crocodile-infested rivers, so maybe it's worth trying to learn something from him.

Learning with Leeper contains four of what Sierra On-Line call "hi-res learning games". As the blurb puts it, educators built in the skill activities and games experts built in the fun.



Leeper waits while Froggy goes a-mazing

Make-a-Monster with Fozzywomp

The four areas of skill focused on are counting, shape matching, eye-hand coordination, and a painting game for "creative play". They are aimed at non-readers, so all control is achieved via the joystick.

The accompanying Parent Guide briefly explains each game. However, it points out that it's not necessary to explain anything to a child who can use a joystick.

The Guide does explain that two of the games contain "hidden" levels.

In the shape matching and eye-hand coordination games, a count is kept of the number of errors made by the child. Depending on how low this count is, higher levels may appear.

The rationale behind hiding levels is that in this way children won't encounter a challenge that is too difficult for them to handle.

However, the Guide also points out that if your child is advanced enough to play at the higher levels and doesn't want to go through the lower ones, the Esc key can be used to move straight there.

Sierra On-Line obviously thought the Leeper formula was a good one because they have also released Learning with Fuzzywomp. They certainly make "learning with my teacher" sound mundane, don't they?

Fuzzywomp is a green lump of fluff who "leads children through four preschool games that prepare them for reading, writing and math".

The four activities are pattern matching, counting, number sequencing and eye-hand coordination. This latter is developed through a make-a-monster game.

The other three are multi-level games. Higher levels are again directly accessible via the Esc key, but otherwise hidden.

Since the two packages seem so similar from what I've said, you might wonder why Sierra On-Line has released both of them. They are even aimed at the same age range – three to six years.

However, as with practically all software with any educational value, the ages on the packet are only rough guidelines at best.

My own feeling is that Leeper is more useful for the lower end of the range and even below, as Lorna demonstrates. Fuzzywomp is more useful at the upper end and even above, depending on the child involved.

Although the packages are "educational", they are clearly aimed at the home education market since both contain a Parent Guide. They both display the professional approach to implementation which we would expect from a major software house like Sierra.

Since Leeper was released before Fuzzywomp, it looks a little less "artistic" in its presentation. Martin Shen and Mike McKee – the programmers of Fuzzywomp – obviously polished some of their techniques after they'd worked on Leeper.

My only reservation about these packages is their cost. With the likely retail price being in excess of £20, they must have a very limited market in the UK.

I've tried them out on other

CLIFF'S COLUMN

DID you ever wonder if anyone managed to crack the code in Penguin's ***Spy's Demise?***

Well, I couldn't crack it, but I did wonder, so while I was on the phone to Illinois I asked about ***The Spy Strikes Back*** *too.*

The news is that so far, nobody from the UK has claimed the special T-shirt or software prizes. These were offered in each American state too, and some of those have been claimed.

Where are our dedicated cryptographers when we need them? Obviously, civil servants still turn to The Times crossword at the office.

Of course, having wondered about the two Penguin games, I naturally got to wondering about ISM's ***Prism*** *Storydisk.*

I mean, T-shirts and software are one thing – depending on how you count – but ***Prism*** *offered the chance to win three gold keys, each bearing a precious stone. One had a diamond, one a ruby and the third a topaz.*

As the blurb proclaimed "wealth awaits the clever" which explains why I never won. In fact, I'm not alone in my ignorance.

To date, they have had a few people claiming to have located one or more keys. However, they tell me that nobody has yet come within 20 miles of a true location.

They have two more

children and nobody had any trouble. Once the joystick has been mastered, the children were able to play unsupervised.

Incidentally, have you noticed how hard it is to find computer-naive children these days? Some of the little dears made comments like "Oh, what's *this* computer? It's not like the BBC one at school!"

Some of these also went straight for the Esc key, which jumps out of the program on the BBC machine unless it's trapped by the program.

So much for the attempt to hide various levels. It must be fun being a primary school teacher these days!

Title: Learning with Leeper
Designer: Nancy Anderton
Programmers: Bob Box, Carolyn Box, Robert Heitman, Mike McKee and Martin Shen
Publisher: Sierra On-Line
Requirements: DOS 3.3 and Joystick

Title: Learning with Fuzzywomp
Programmers: Mike McKee and Martin Shen
Publisher: Sierra On-Line
Requirements: Joystick

Storydisks on the way. These will be aimed at younger children and will be more interactive, but as far as I know there won't be any prizes involved.

Of course, if you can't win one of the prizes up for grabs, you can always buy one of the games which offer "extras".

My own favourite is the T-shirt transfer which accompanies ***Repton*** *from Sirius Software. However, the colouring book ("lavish documentation") which came with Krell's* ***Adventures in Flesh*** *is nice if you have a good set of crayons.*

The map from Sierra On-Line's ***Ultima II*** *bore a close resemblance to a tea-towel, but I can't imagine anyone actually drying the dishes with it.*

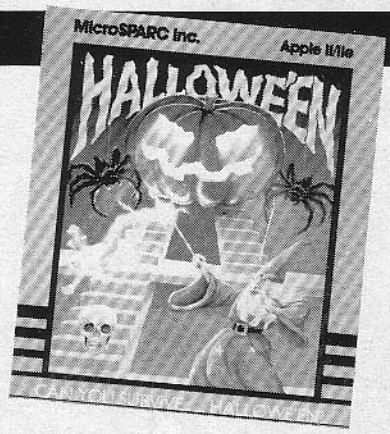
With Origin Systems' ***Ultima III,*** *the map had a texture which certainly wouldn't allow such a use!*

I can't wait to see what the marketing departments dream up for us next. If I wanted to be cynical, I could suggest they reduce the price of the games instead. What, me, cynical?

Incidentally, talking of prizes, you'll be pleased to hear that we recently received the winning entry for the competition I set in the April column. Well done, Harry from Australia.

Apple User's generosity knows no bounds, so by the time you read this you should have received the prize.

Skull-duggery in the house of horror . . .



WE'VE all heard of "Double, double toil and trouble; fire burn and cauldron bubble", and indeed what has been cooked up for you here is certainly very nasty.

After removing the pumpkin-coloured disc (well, it makes a change) from its sleeve you push on to find the idea behind Hallowe'en is really quite simple.

It's ladies' night and they have gone out on their broomsticks leaving you to clean up the haunted house. Household chores are not very pleasant, but usually don't present much of a problem you may think. Prepare for a surprise.

You must rid the house of a vicious lot of killer pumpkins, bouncing skulls, venomous spiders and chilling choppers.

Worse is yet to come. You find out that they are trying to rid the house of you!

You start, a lowly wizard, at the bottom of a five-floor screen with only a wand for protection. Movement between the floors is made possible by an assortment of ladders.

Your aim is to kill all the skulls and pumpkins on the screen before moving on to the next one.

To kill them you must first zap a hole in the ground with your wand, pulled mysteriously from the depths of your robe, and when a nasty falls into the hole you zap it until it falls to its death on the next floor.

Although it is not necessary to kill your enemies – except, that is, the pumpkins and skulls – doing so adds a fair amount to your survival score, as the authors call it.

There is a time limit on every screen and if you reach it you lose a man. You also lose a man when you run into one of the enemies.

After every third screen comes a challenging stage. This time you find letters moving around and you try to zap them in the correct order to spell a word.

If you manage this – and it is not easy – you collect a large bonus and you get onto a different challenging stage after the next three screens.

If you don't get all the letters in time, or if some of them disappear when you run into them, you must do the same challenging stage the next time. You don't lose a man if the time runs out on a challenging stage.

Altogether there are 10 different stages with increasingly difficult combinations of enemies to defeat. Once you have completed the last stage, it is repeated until you lose all your men.

Hallowe'en takes a lot of getting into, especially the set-up option. This allows you to change a whole range of game parameters (number of enemies, type of enemies, length of time for screen, etc).

It takes a lot of experimentation to work out what all the parameters change, as the instructions here are minimal. When using the set-up option on my copy the game just stopped.

Luckily there is a restore set-up function which sets all the parameters for you, but this also clears the high scores.

The controls take a bit of getting used to, as the paddle button is used for jumping, zapping and climbing ladders.

The instructions say the game can be played with either paddle or joystick, but it is impossible to use the joystick for the two-player game as both players are on the screen at once, each using one paddle.

A tremendous element of luck is brought in by the random placing of the figures on the screen, mainly on the higher stages. Screens can move from the relatively easy to the almost impossible depending on this.

Hallowe'en is the first 64k game I have seen and it does seem to have a bit more going for it than the usual arcade game.

However it did get very frustrating when it was impossible to go on, and it would surely have been more enjoyable had the set-up option worked.

Douglas Strathdee

Says Cliff McKnight: Micro Sparc sent two copies of the game so I sent one to Douglas and kept one myself. The set-up option works fine on my copy and we didn't have time to check if the fault was hardware or software. Sorry, Douglas!

Title: Hallowe'en
Authors: Bernd Rass and Thomas Schumann
Publisher: MicroSparc
Requirements: 64k Apple

How to teach Visicalc to think for itself

By NICK LEVY

HAVE you noticed how easy it is to make mockery of Visicalc? Take for example the capacity planning model shown as Exhibit I.

Suppose you want to find out what would be the available capacity in week 28 (column F), assuming that on that week we had 8 working days in cell F8, or alternatively, –3 working days?

As Visicalc was not told that you cannot have more than seven days per week, nor can you have a negative number of working days, it will enter 778 in cell F17 in response to the first absurd "What if" assumption, and –292 in response to the second clownish supposition.

Similarly, if you enter 20 in cell F11 (number of operators on holiday) Visicalc may not always appreciate the fact that you cannot have more workers on holiday than the total number of 18 available operators as quantified in cell F10.

So being ignorant of such simple facts of life, it will respond by displaying in cell F17 a meaningless figure of available capacity of –61 hours.

Exhibit II is a listing of the key formulae used to develop the simple capacity planning model shown as Exhibit I.

Unfortunately the automatic computation of such absurd figures provide the anti-computer lobby, which probably exists in every organisation, with a unique opportunity to jeer and sneer at whoever has developed or demonstrates such Visicalc models.

But here is the good news – Visicalc can be made foolproof. It can be taught to think for itself and fight back anyone trying to take the mickey out of it.

Not only can it be made to halt the computing process when anyone attempts to enter, say, an eight day week, but it will also raise the alarm bells if for example anyone tries to contemplate a working shift which lasts more than 12 hours (row 9) or allowing for the productivity performance suddenly to jump unexpectedly to 150 per cent (row 16).

And here is the proof. Exhibit III shows how the capacity planning model can be formulated to reject such teasing entries as when:

- Eight working days per week are entered in cell F8.
- More operators are shown to be on holiday according to the entry in cell G11, than the available number of operators in cell G10.
- The productivity performance in week 30 suddenly jumps to 150 per cent (cell H16).

The formulae used in the capacity planning model shown as Exhibit III are listed in Exhibit IV. In addition to the above three examples, the formulae in this exhibit are also designed not to allow the following unacceptable circumstances to be passed unnoticed:

- ☐ Entering negative figures for the number of working days per week, or negative number of hours per shift, or negative number of operators on holiday or a negative percentage allowance for sickness/lateness.
- ☐ Entering productivity performances of less than 60 per cent or allowances for sickness/lateness higher than 25 per cent.

Incidentally, Exhibits II and IV were produced by a program called Docucalc, a formulae reader which has a facility for printing only the contents of Visicalc cells which contain formulae. In other words, it will not print the contents of cells containing labels or values.

Making foolproof Visicalc models requires first and foremost knowledge and ability to be able to manipulate Visicalc's logic functions, especially the @IF function.

It also requires experience in developing Visicalc formulae where one set of formulae is embedded in another. It follows that you must be prepared and able to develop formulae which are more than 100 characters long.

So having mastered the basics of Visicalc, why not venture into learning its logic

	A	B	C	D	E	F	G	H	I	J	K	L
1	Exercise in Capacity Planning:											
2	==========					Department:.....						
3												
4	Period: July 198..										Total	Av.
5	----------										For	Per
6	Week No.					28	29	30	31	32	Month	Week
7	----------											
8	Working Days					5	5	5	4	5	24	4.80
9	Hours per Shift					8	8	8	8	8	-	8.00
10	Total Number of Operators					18	18	19	19	20	-	18.80
11	No. of Operators on Holiday					2	2	1	1	0	-	1.20
12	Available Number of Operators					16	16	18	18	20	-	17.60
13	Available Working Hours					640	640	720	576	800	3376	675
14	% Allow.for Sickness/Latness					5	4	3	3	2	-	3.33
15	Expected Attendance Hours					608	614	698	559	784	3264	653
16	Est.Productivity Perform. (%)					80	86	86	86	90	-	85.84
17	Avail.Capacity in Hours					486	528	601	480	706	2802	560
18	==========											
19	Check acceptability of data:											
20	++++++++++											
21												
22												
23												
24												

Exhibit I

K8:	Format	Integer	@SUM(F8...J8)
L8:	Format	Two decimal	@AVERAGE(F8...J8)
L9:	Format	Two decimal	@AVERAGE(F9...J9)
L10:	Format	Two decimal	@AVERAGE(F10...J10)
L11:	Format	Two decimal	@AVERAGE(F11...J11)
F12:			+F10-F11
L12:	Format	Two decimal	@AVERAGE(F12...J12)
F13:			+F12*F8*F9
K13:	Format	Integer	@SUM(F13...J13)
L13:	Format	Integer	@AVERAGE(F13...J13)
L14:	Format	Two decimal	+K13-K15/K13*100
F15:	Format	Integer	100-F14*F13/100
K15:	Format	Integer	@SUM(F15...J15)
L15:	Format	Integer	@AVERAGE(F15...J15)
L16:	Format	Two decimal	+K17/K15*100
F17:	Format	Integer	+F15*F16/100
K17:	Format	Integer	@SUM(F17...J17)
L17:	Format	Integer	@AVERAGE(F17...J17)

Exhibit II

functions. The results can be most rewarding. Unfortunately the Visicalc manual is not the best source for learning how to use and apply these functions, mainly because it lacks practical examples.

The following must inevitably be only a brief explanation for the perplexed reader who is bewildered by all these functions: @IF, @NA, @MAX, @OR, @AND and @ISNA, which appear in Exhibit IV.

Note the formula for F12 in Exhibit II, +F10–F11. Why was it necessary to transform that formula to @IF (F11>=0, @MAX(0,F10-F11), @NA) in cell F12 in Exhibit IV?

Let's look first at the middle part of the formula which reads: MAX(0,F10–F11).

What the formula states, in plain words, is that we want to enter in cell F12 the larger of two numbers, zero or F10 less F11.

The purpose of this formula is to ensure that cell F12 cannot contain a negative figure. If F10 is 18 and F11 is 20, then the answer should be –2. But the availability of –2 operators does not make sense.

Fortunately however, –2 is less than zero so the zero will prevail in F12. This use of the MAX function provides a useful ploy for when it is desirable not to display negative figures in a working model of Visicalc.

The first and third parts of that formula (cell F12) state that the @MAX calculation (0,F10-F11) should only be performed if the content of cell F11 is larger or equal to zero.

If F11 happens to be less than zero then enter NA (not applicable) in cell F12.

Similar interpretations can be given to the formulas in cells F13, F15 and F17.

Note that each cell in row 19 contains a formula, yet the cells sometimes appear empty and sometimes display the NA notice.

How was that done? The secret lies in formatting these cells to bar chart (for example /F*). When all is well, the formulae in these cells should display '0', but as you cannot display zero number of asterisks the cell appears empty.

If on the other hand there are some irregularities in the figures in the model, the formula responds with NA, which stands out like a sore thumb.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Exercise in Capacity Planning:											
2	==========					Department:.....						
3												
4	Period: July 198..										Total	Av.
5											For	Per
6	Week No.					28	29	30	31	32	Month	Week
7												
8	Working Days					8	5	5	4	5	27	5.40
9	Hours per Shift					8	8	8	8	8	-	8.00
10	Total Number of Operators					18	18	19	19	20	-	18.80
11	No. of Operators on Holiday					2	20	1	1	0	-	4.80
12	Available Number of Operators					16	0	18	18	20	-	14.40
13	Available Working Hours					NA	0	720	576	800	NA	NA
14	% Allow.for Sickness/Latness					5	4	3	3	2	-	NA
15	Expected Attendance Hours					NA	0	698	559	784	NA	NA
16	Est.Productivity Perform. (%)					80	86	150	86	90	-	NA
17	Avail.Capacity in Hours					NA	0	1048	480	706	NA	NA
18	==========											
19	Check acceptability of data:							NA			NA	NA
20	++++++++++											

Exhibit III

```
K8:   Format      Integer       @SUM(F8...J8)
L8:   Format      Two decimal   @AVERAGE(F8...J8)
L9:   Format      Two decimal   @AVERAGE(F9...J9)
L10:  Format      Two decimal   @AVERAGE(F10...J10)
L11:  Format      Two decimal   @AVERAGE(F11...J11)
F12:                            @IF(F11>=0,@MAX(0,F10-F11),@NA)
L12:  Format      Two decimal   @AVERAGE(F12...J12)
F13:                            @IF(F8<=7,@MAX(0,F12*F8*F9),@NA)
K13:  Format      Integer       @SUM(F13...J13)
L13:  Format      Integer       @AVERAGE(F13...J13)
L14:  Format      Two decimal   +K13-K15/K13*100
F15:  Format      Integer       @IF(F14<0,@NA,100-F14*F13/100)
K15:  Format      Integer       @SUM(F15...J15)
L15:  Format      Integer       @AVERAGE(F15...J15)
L16:  Format      Two decimal   +K17/K15*100
F17:  Format      Integer       @IF(F16<0,@NA,F15*F16/100)
K17:  Format      Integer       @SUM(F17...J17)
L17:  Format      Integer       @AVERAGE(F17...J17)
F19:  Format      Bar chart     @IF(@AND(F9<=12,F14<=25,F16>60,F16<140),0,@NA)
K19:  Format      Bar chart     @IF(@OR(@ISNA(F19),@ISNA(G19),@ISNA(H19),@ISNA(I19),@IS
                                NA(J19)),@NA,0)
L19:  Format      Bar chart     +K19
```

Exhibit IV

Advanced Version harbours a bug

– reports PATRICK O'BEIRNE

IT'S rare to hear of a bug in Visicalc. It sells so many copies that problems are usually pretty thoroughly shaken down before release. Ordinary programmers can now take some comfort from the news that the biggies make mistakes just like the rest of us!

So, if you have an Apple III with Visicalc Advanced Version and are using DIF files extensively to pass information between worksheets, *be warned.* The problem is this:

Extra digits are added on to some cells when large DIF files are saved. It is reproducible for any given worksheet, but changing entries in the sheet change which cells are affected, and there seems (to me anyway) no obvious pattern.

The problem was first reported by a user in the Institute for Industrial Research and Standards. For example, in a worksheet where he saved from J23 to AM31 (nine rows by 30 columns), the following data was in row 23 (the row numbers start at 1, but column numbers shown start with A=0, B=1, etc.):

	AE (col. 30)	AF (col. 31)	... AM (col. 38)
Row 23	16	16	... 3

When the DIF file is saved, the integers above have extra digits added as follows:

If saved from J23 to AM31 – that is, with a lower right-hand corner of Row 31 and column 38 – the DIF file contains:

First Row	16.3031	16.3131	... 3.3831

Note that VisiCalc has added on four digits after the decimal point: two for the column number (30, 31, 38) and two for the bottom row number (31).

But if he saved from J23 to AM30, he got:

First Row	16	16	... 3.3830

Note that the 16 is now saved correctly, but the last cell is still wrong. And saving to AM29 gives:

First Row	16	16	... 3.3829

You should see the pattern now. When he saved only to row 26, the problem disappeared. So it looks as though DIF saves with as few as 150 cells in them can suffer from this problem. When this was reported to VisiCorp, they said that it was a known bug and that no fix was available. I guess that we will have to wait for the dispute with Software Arts (who actually wrote Visicalc) to be resolved first.

I think they should have warned users via the dealer network, but they didn't do that with the last Apple II bug either, back in 1981. With such a low bug rate compared to many other products, they need not hide them! Incidentally, there are a couple of – shall we say – inconsistencies in the way VAV III handles titles in the /Print command, but that's another story.

One important lesson here. What really caused the user worry at first was that he had been displaying all his figures as integers – so none of the above data corruption was visible. Only his good practice in building in check totals and calculations in subsequent worksheets saved him.

Sample the delights of programming with this mini-Pilot

A simple but effective mini-Pilot – ideally suited for education – from GORDON MILLS

CHUCK Carpenter published a version of a Pilot interpreter, written in Basic, which he had modified from a public domain Pilot program of N. Dealey.

Unfortunately, Chuck's version had several defects, the most serious being a propensity to crash after more than one EDIT.

In an attempt to remedy some of the deficiencies, the program has been completely revised and rewritten so that the ONERR command is not necessary. The original program continually jumped out of FOR . . . NEXT loops, eventually corrupting the stack.

When there is already available a perfectly good and comprehensive Apple Pilot, what is the justification for publishing a Mini-Pilot which is a very limited version with text only, no calculation facility and no subroutine capability? My reasons are:

☐ The main users of Pilot are likely to be educationalists. Many will baulk at the cost of the full version – about £80 – unless they are convinced that it will be of real value.

☐ Experience with running courses on micros in education indicates that teachers are very interested in trying out a simple user-friendly Pilot version which they can pick up very quickly.

☐ The commercial Apple Pilot is now at least as complex to program with as Basic. What has happened to the original aim of Pilot which was to provide a language to make it *easy* for practising teachers to write programs?

Gordon Mills is a lecturer in microcomputer applications in psychology at Bradford University.

☐ One advantage of a simple Pilot is that beginners in programming can be more easily initiated into the techniques of simple program structures.

Chuck Carpenter's version allowed the following main commands:

T: *Text line.*
A: *Accept user input.*
M: *Strings to check if there is a match with the last user input.*
J: *Jump to another line.*
C: *Clear the screen.*
R: *Remark or comment line.*
E: *(or END:) End the Pilot program.*

All the main commands can be modified by Y or N immediately following the command symbols which cause the command to operate only if the modifier is Y and the last match is positive or the modifier is N and the last match is negative.

Additional symbols are * to precede labels of lines to which the programmer wishes to jump, and $ to precede string labels in lines accepting user input – these can be used later in the program in text lines.

Examples are given in Program I and in the instructions for Program II.

When using the MATCH command, more than one string can be checked provided each is separated by a comma. Note also that the last symbol of a match line must always be a single quote (').

As there is a limitation of 33 characters on each line, if the number of matches to be checked is greater than can be accommodated on one line, the continuation match lines must start with MN: as shown in line 6 of Program I.

Apart from improving the integrity and structure of the original program, additional features include a comprehensive menu, improved editing by allowing the insertion and deletion of lines in a Pilot program and an option for the Pilot program to be listed on a printer.

There is also a paged screen listing of the Pilot program.

The maximum number of Pilot program lines is 400 – this could be enlarged further by increasing the value of MX in line 130. Alternatively, initialising/editing delays could be reduced by decreasing MX.

Instructions and commands, together with a sample Pilot program are included within the Basic program, as well as a score command, S: which can be followed by any number (positive or negative).

If the score number is less than 999, that number is added to the current score. If it is 999 or greater the score is reset to zero. At the end of the program the score is printed out.

One of the simplest uses for Pilot in education is for presenting simple knowledge tests. The example given in Program I, part of a Bible quiz, is intended to illustrate how program structures can be developed from the simple to the more complex, so providing an ideal teaching medium for beginners.

Question 1 allows one attempt only. Question 2 shows how a multiple choice question can be written. Questions 3 and 4 allow the user any number of attempts but it is important to include, as in Question 4, the option (?) to escape from what would be an endless loop if the answer is not automatically given after the user response.

Question 5 shows how to get two answers from one question. This involves a much more complex structure than the previous questions. The five types of questions demonstrated in Program I can serve as a model for similar questions in any discipline.

This Mini-Pilot interpreter, although written in Basic, is surprisingly fast.

The only delays – of several seconds – occur on initialising (including when LOADing from disc), and when inserting blank lines or deleting lines (under EDIT).

Also, the modular construction allows the expert programmer to readily experiment with additional commands if desired.

One other factor borne in mind during the rewriting was to allow easy transfer to many other low cost micros by just changing the INITIALIZATION section (lines 100 to 190) and all the subroutines from lines 5000 upwards.

Now is your chance to experiment with a simplified Pilot without parting with any money!

Bible Quiz

```
DISCIPLES
0R:BIBLE QUIZ
1*1
2C:
3T:WHAT WAS MATTHEW'S PROFESSION?
4A:
5M:TAXMAN,TAX-GATHERER,'
6MN:TAX COLLECTOR,'
7SY:3
8TY:GOOD.
9T:HE WAS A TAX COLLECTOR
10T:NOW PRESS RETURN
11A:
12*2
13C:
14T:WHEN DID THE DISCIPLES SAY,
15T:'MASTER, CAREST THOU NOT
16T: THAT WE PERISH?'
17T:
18T: A - AT THE LAST SUPPER
19T: B - WHEN JESUS WAS ASLEEP
20T:     DURING THE STORM
21T:
22T:TYPE  A  OR  B
23A:
24M:B,'
25TN:ANSWER B WAS CORRECT
26TY:VERY GOOD
27SY:2
28SN:-1
29T:PRESS RETURN
30A:
31*3
32C:
33T:OF WHOM DID JESUS SAY,
34T:'BEHOLD AN ISRAELITE INDEED IN
35T: WHOM THERE IS NO GUILE?'
36A:
37M:NATHANAEL,NATHANIEL,'
38SN:-1
39TN:CORRECT IS NATHANAEL
40SY:3
41TN:NOW PRESS RETURN
42AN:
43JN:*3
44T:GOOD. NOW PRESS RETURN
45A:
46*4
47C:
48T:IF YOU REALLY DON'T KNOW THE
49T:ANSWER TO THE NEXT QUESTION,
50T:TYPE  ?
51*4A
52T:WHO WAS PETER'S BROTHER?
53A:
54M:ANDREW,'
55TY:GOOD. NOW PRESS RETURN
56AY:
57SY:3
58JY:*5
59M:?,'
60SY:-1
61SN:-2
62TN:TRY AGAIN. PRESS RETURN.
63AN:
64JN:*4
65TY:CORRECT IS ANDREW
66T:
67TY:PRESS RETURN
68A:
69C:
70JY:*4A
71*5
72R:A QUESTION WITH TWO ANSWERS
73C:
74T:NATHANAEL AND ANDREW WERE TWO
75T:OF FOUR DISCIPLES WHO FIRST
76T:FOLLOWED JOHN THE BAPTIST.
77T:NAME ONE OF THE OTHERS.
78A:
79M:PETER,'
80SY:1
81JN:*5A
82T:GOOD. WHO WAS THE FOURTH?
83A:
84M:PHILIP,'
85JN:*5B
86*5C
87SY:2
88T:EXCELLENT. PRESS RETURN
89A:
90EY:
91*5B
92T:CORRECT IS PHILIP
93*5D
94SN:-2
95T:PRESS RETURN
96A:
97J:*5
98*5A
99M:PHILIP,'
100SN:-2
101SY:1
102TN:TRY AGAIN
103JN:*5
104T:GOOD. WHO IS THE FOURTH?
105A:
106M:PETER,'
107JY:*5C
108T:CORRECT IS PETER
109J:*5D
```

Program I

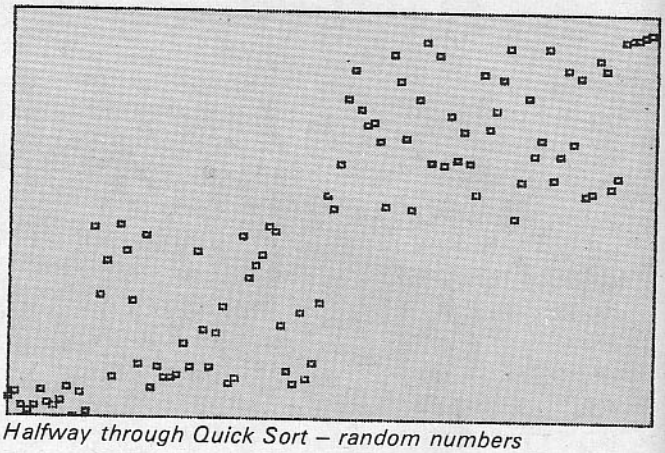
```
10  REM  MINI-PILOT
    INTERPRETER
20  REM  INITIAL VERSION BY
    N.DEALEY 1978
30  REM  MODIFIED BY CHUCK
    CARPENTER 1979
40  REM  REWRITTEN BY GORDON
    MILLS  1983
100  REM INITIALIZE
120 D0$ = ":":D1$ = ","
130 MX = 399:LL = 33:R = 19:LT
    = 30:NL = 20
140  DIM
    P$(1,MX),I$(1,R),S$(7)
170 D$ =  CHR$ (4)
180  PRINT
190  PRINT D$;"NOMON I,O,C":
    HOME
200  REM  DATA
210  FOR I = 1 TO 9: READ
    M$(I): NEXT
220  DATA  NEW
    PROGRAM,LIST,EDIT,RUN,REPLAC
    E,SAVE,LOAD,END,INSTRUCTIONS
230  FOR I = 1 TO 7: READ
    S$(I): NEXT
240  DATA A,T,M,J,C,E,S
499  REM  CONTROL SECTION
500  GOSUB 1000: GOTO 600
510  GOSUB 1100: GOTO 600
520  GOSUB 1200: GOTO 600
530  GOSUB 1300: GOTO 600
540  GOSUB 1400: GOTO 600
550  GOSUB 1500: GOTO 600
560  GOSUB 1600: GOTO 600
570  GOSUB 1700: GOTO 600
580  GOSUB 1800: GOTO 600
```

Listing continues on Page 73

Program II

More graphic guidance from PETER GORRY

Forever blowing bubbles...



Halfway through Quick Sort – random numbers

RECENTLY I was confronted with that perennial task – sorting a randomly ordered list into alphabetical order. I'm always tempted to use the good old faithful bubble sort for such tasks because it's simple to write and easy to understand.

Unfortunately for long lists it is very slow and, unless you've got a good book to read, it really isn't practical for serious applications.

I decided I'd need a faster sorting algorithm so I did a quick search of back issues of *Windfall* and *Apple User* for inspiration.

I settled on "Shell sort" by R.A. Mould *(Windfall* August 1982) and "Quicksort" by J.P. Lewis *(Windfall* January 1983) as fast efficient algorithms. I wanted to sort about 100 items on a regular basis so I programmed them both up to see which was best. In fact both algorithms were equally fast – about 10 times quicker than the bubble sort.

The shell sort uses no special tricks but the quicksort is a recursive routine – which normally can't be done in Basic.

However this version cleverly overcomes the problem by using a two dimensional array STACK(2,10) to keep track of the variables.

The size of the array needs to be increased to STACK(2,15) for 2000 items in the list.

All was fine apart from the fact that I don't like using things that I don't understand, and the more I looked at the code the more opaque it seemed to get.

Both algorithms rely on splitting the whole list into smaller sub-lists and then working on them, but in different ways.

Since I find pictures so much easier to follow than words (I never did get the hang of reading) it seemed that a visual presentation of the sorting might be more illuminating.

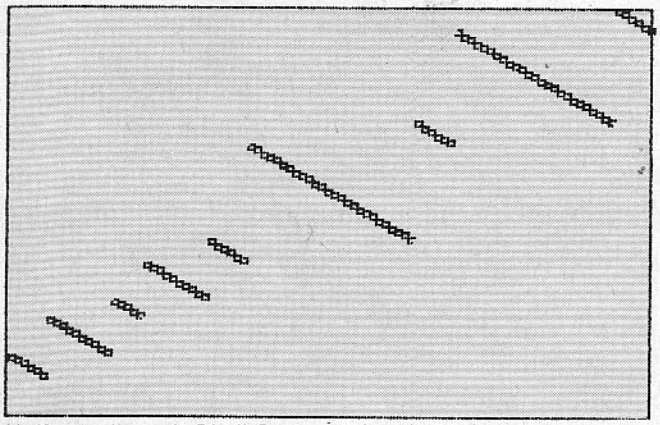
The listing given below provides just such a pictorial view of the sorting processes used by bubble, shell and quicksort routines.

The program sets up 100 numbers with values in the range 0-150 in an array X(100). The value of each element in the array is then plotted on the hi-res screen.

These are sorted into a *descending* order by each of the routines in turn. A descending order is chosen because, when plotted on the screen, it looks like an ascending order (remember 0 is at the *top* of the screen).

The program offers two choices for the initial numbers – a random order or numbers in ascending order, that is totally the wrong way round. The latter

```
100  REM  SORT PROG
110  DIM STACK(2,10),X(100)
120  TEXT : HOME :N = 100: REM  1
     00 NUMBERS
130  PRINT : PRINT "1  RANDOM SEQ
     UENCE": PRINT : PRINT "2  IN
     VERSE SEQUENCE"
140  PRINT : INPUT "WHICH ? ";NT
150  IF NT < 1 OR NT > 2 THEN  CALL
      - 198: GOTO 120
160  GOSUB 4000: REM  SET UP NUMB
     ERS
170  HTAB 10: VTAB 22: PRINT "BUB
     BLE SORT"
180  GOSUB 1000: REM  BUBBLE SORT

190  GOSUB 5000: REM  SPACE BAR
200  GOSUB 4000: REM  SET UP NUMB
     ERS
210  HTAB 10: VTAB 22: PRINT "SHE
     LL SORT"
220  GOSUB 2000: REM  SHELL SORT
230  GOSUB 5000: REM  SPACE BAR
240  GOSUB 4000: REM  SET UP NUMB
     ERS
250  HTAB 10: VTAB 22: PRINT "QUI
     CK SORT"
260  GOSUB 3000: REM  QUICKSORT
270  END :

1000  REM

BUBBLE SORT

1010 K = 1
1020 SWAPS = 0
1030  FOR J = 1 TO N - K
1040  IF X(J) > = X(J + 1) THEN
      1110
1050  REM  SWAP NUMBERS
1060  HCOLOR= 4: HPLOT J * 2,X(J)
      : HPLOT J * 2 + 2,X(J + 1)
1070 W = X(J):X(J) = X(J + 1)
1080 X(J + 1) = W
1090  HCOLOR= 3: HPLOT J * 2,X(J)
      : HPLOT J * 2 + 2,X(J + 1)
1100 SWAPS = SWAPS + 1
1110  NEXT
1120  IF SWAPS > 0 THEN K = K + 1
      : GOTO 1020
1130  RETURN :

2000  REM

SHELL SORT

2010 I = 1
2020 I = 2 * I
2030  IF I <  = N THEN 2020
2040 Q = I - 1
2050 Q = (Q - 1) / 2
2060  IF Q = 0 THEN 2190
2070  FOR J = 1 TO N - Q
2080  FOR I = J TO 1 STEP  - Q
2090 L = I + Q
2100  IF X(L) <  = X(I) THEN 2170

2110  HCOLOR= 4: HPLOT L * 2,X(L)
      : HPLOT I * 2,X(I)
```



Halfway through Shell Sort – ordered numbers

is very good at revealing the different strategies adopted.

A couple of modifications can be made to increase the clarity even more.

One is to highlight the two elements being used in each comparison (XDRAW a small circle round them, say). The other puts a delay loop after each swap to give you more time to follow the changes.

The sorting routines can be incorporated into a program for normal use simply by taking out the plotting sections, for example lines 2110 and 2150 for shell sort.

If you want an ascending order you just have to swap the < and > symbols over in the tests – lines 1040, 2100, 3050 and 3060.

With my new-found understanding of sorting techniques bubble sorts are a thing of the past, at least until I need a routine in a hurry and . . . well . . . they are so easy to write and, in this hyper-efficient computer world, I find their inefficiency rather comforting.

Now where did I put "War and Peace"?

No trouble with double hi-res

I WAS very excited when I got my first Apple IIe with extended 80 column card (extra 64k memory) because I had heard that it would allow me to do double hi-res graphics. That is 560 x 192 points on the hi-res screen.

My hopes were soon dashed when I discovered just how the IIe performed this miracle. Briefly, it uses the normal hi-res page 1 for half the picture and an alternative "hi-res page 1" (on the 80 column card) for the other half.

Unfortunately it doesn't do this in half page chunks but in little narrow strips.

If we draw a horizontal line from 0-599 the first seven pixels must be on the alternative page 1, the next seven must be on normal page 1, then back to the alternative page for the next seven – and so on.

This interleaving of the pages makes trying to use double resolution a nightmare and, of course, the normal Applesoft graphical commands are totally useless.

It was clear that a lot of work would be required to perform all the page swapping and point plotting, so I went back to my world of single res.

Fortunately a new product written by Louis Bonfiglio and Peter Joselow called Doublestuff has come to the rescue.

It replaces all the usual Applesoft commands by their double resolution equivalents – so there's almost nothing new to learn in order to use it.

The system works by putting a modified version of Applesoft

```
 2120 W = X(I)
 2130 X(I) = X(L)
 2140 X(L) = W
 2150  HCOLOR= 3: HPLOT L * 2,X(L)
     : HPLOT I * 2,X(I)
 2160  NEXT I
 2170  NEXT J
 2180  GOTO 2050
 2190  RETURN :

 3000  REM

QUICKSORT

 3010 B = 1:LOW = 1:HIGH = N
 3020 JLOW = LOW:KHIGH = HIGH
 3030 PIVOT = X((JLOW + KHIGH) / 2
     )
 3040  IF JLOW > KHIGH THEN 3090
 3050  IF X(KHIGH) < PIVOT THEN KH
     IGH = KHIGH - 1: GOTO 3050
 3060  IF X(JLOW) > PIVOT THEN JLO
     W = JLOW + 1: GOTO 3060
 3070  IF JLOW <  = KHIGH THEN  HCOLOR=
     4: HPLOT 2 * JLOW,X(JLOW): HPLOT
     2 * KHIGH,X(KHIGH):W = X(JLO
     W):X(JLOW) = X(KHIGH):X(KHIG
     H) = W: HCOLOR= 3: HPLOT 2 *
     JLOW,X(JLOW): HPLOT 2 * KHIG
     H,X(KHIGH):JLOW = JLOW + 1:K
     HIGH = KHIGH - 1
 3080  GOTO 3040
 3090 B = B + 1:STACK(1,B) = HIGH:
     STACK(2,B) = JLOW
 3100 HIGH = KHIGH: IF LOW < HIGH THEN
      GOSUB 3020
 3110 LOW = STACK(2,B):HIGH = STAC
     K(1,B):B = B - 1: IF LOW < H
     IGH THEN  GOSUB 3020
 3120  RETURN :
 4000  REM  SET UP NUMBERS
 4010  IF NT <  > 1 THEN 4040: REM
     STRAIGHT LINE
 4020 Z =  RND ( - 1): REM  REPEAT
     ABLE SEQUENCE
 4030  FOR I = 1 TO N:X(I) =  RND
     (1) * 150: NEXT : GOTO 4050
 4040  FOR I = 1 TO N:X(I) = I: NEXT

 4050  HGR : HCOLOR= 3: SCALE= 1: ROT=
     0
 4060  FOR I = 1 TO N: HPLOT I * 2
     ,X(I): NEXT
 4070  RETURN :
 5000  REM
 5010  REM  GET SPACE BAR ROUTINE
 5020  REM
 5030  REM  PUT QUESTION AT HTAB 8
     :VTAB22
 5040  HTAB 8: VTAB 22: CALL  - 86
     8: REM  CLEAR TO END OF LINE
 5050  PRINT "PRESS <SPACE> TO CON
     TINUE";
 5060  NORMAL : GET ZZ$: IF ZZ$ <
      > " " THEN  CALL  - 198: GOTO
     5040
 5070  HTAB 8: VTAB 22: CALL  - 86
     8
 5080  RETURN :
```

on the language card of the IIe. This contains the double resolution graphics routines, but everything else is the same as ordinary Applesoft – the only visible difference is that the prompt cursor is) not the normal].

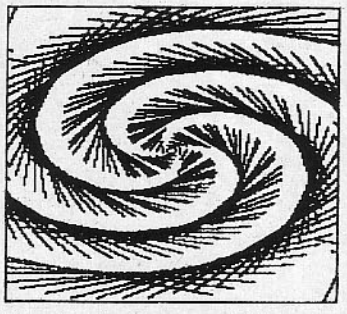
This means that all your normal programs will run immediately, although not exactly as you expect, as we shall see.

You can switch between Doublestuff and Applesoft by using DFP and FP rather as you would when switching between Integer and Applesoft with INT and FP. There is even a tip on how to preserve the program during such switches if you so wish.

Doublestuff comes on an unprotected disc, with a slim instruction manual and a handy reference card. In fact I found the manual a little too slim.

It wasn't that I couldn't use Doublestuff straight away, but rather that after a little use I had several unanswered questions.

Such as: How do I save the final pictures to disc? How do I load them back in again? Can I convert pictures between single and double-res easily? Will the DOS toolkit hi-res character generator still work in double-res? Can I compile the Doublestuff programs like I can Applesoft . . .?



The illustrations on this page were produced with Doublestuff, using an ordinary dot matrix printer.

The disc comes with a host of programs to demonstrate the double resolution in both lo-res and hi-res graphics. These consist of colour charts, patterns and pictures of various types. They are selected from a menu and give a good feel for the things Doublestuff has to offer.

One of the most immediately striking features is that Doublestuff offers 16 colours on the hi-res screen instead of the usual 6.

These have the same numbers as the 16 lo-res colours – those available on the lo-res screen in normal Applesoft.

Unfortunately Applesoft uses different numbers for the same colour in lo-res and hi-res (for example, white is 15 in lo-res but 3 or 7 in hi-res). As a result an unmodified Applesoft program using hi-res will come out with different colours in Doublestuff.

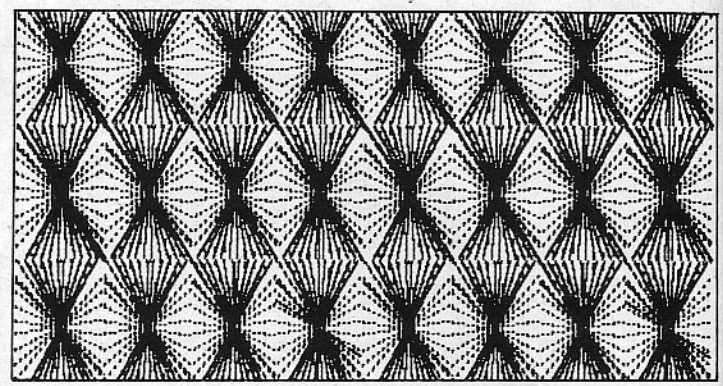
I soon found that although all my existing Applesoft programs would run under Doublestuff, they all needed slight modifications to get the desired effect.

This is due to the slightly different way Doublestuff handles the graphics commands. The major differences in Doublestuff hi-res are:

HGR sets the (double-res) mode to text+graphics and clears the screen to the last designated HCOLOR (not black as in Applesoft).

HGR2 sets the mode to graphics only and clears to the last HCOLOR.

HCOLOR sets the hi-res colour in the range 0-15 (as in lo-res).



HPLOT, DRAW, XDRAW all work over the range 0-559, 0-191.

Any of you who have used colour on the Apple will be well aware of the problem that plotting vertical lines presents in hi-res – for instance white lines come out purple or green depending on the position. One has to plot in pairs to obtain a reasonable white line.

The same limitations apply to Doublestuff, but sometimes even more so. I would have liked a little better description of what to expect in the manual.

The only hint one gets is: "Remember, especially in double hi-res that if you choose a colour and then plot a point on the graphics field, you will not get that solid colour until you plot the correct number of pixels that compose that colour (up to four pixels for any colour)". That's all, apart from encouragement to experiment!

In fact I have one of the colour cards that allows one to define a two colour mode for use in hi-res. In this mode only background and foreground are distinguishable colours.

This does away with all the colour anomolies and the full double resolution can be achieved without the eye-strain produced by the rainbow version. I found this much better for serious applications.

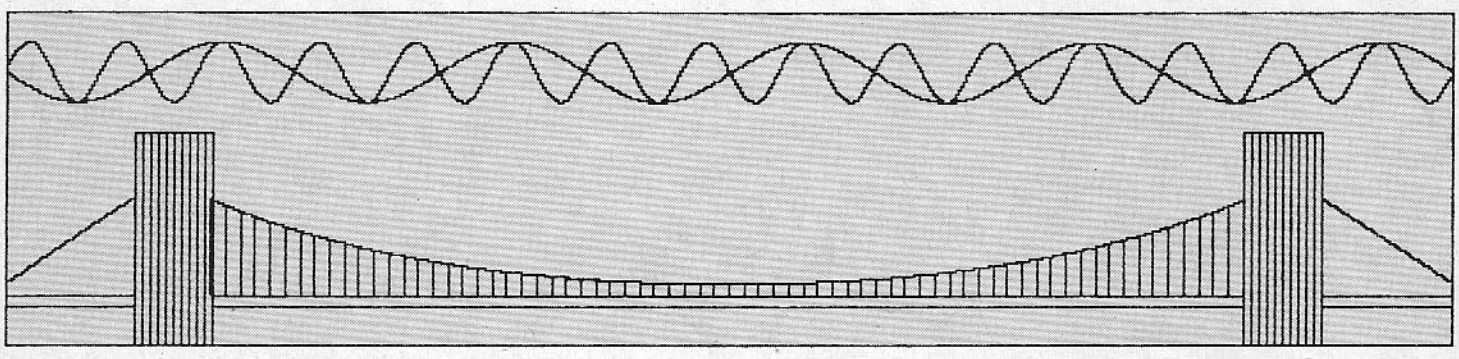
To return to some of the questions I raised earlier – yes you can use the DOS Toolkit character generator – but it comes out with a space between each letter.

No you can't compile the programs (with TASC at least). Saving and loading to disc is not quite straightforward and the best thing to do is inspect the demo programs to see how they do it. This last point really should have been dealt with in the manual.

To summarise then, Doublestuff works very well and is easy to use. It is well thought out – I never managed to crash it unexpectedly. It provides double resolution graphics with the minimum of fuss and with an extended range of colours.

I certainly enjoyed using it and it kept me at the keyboard for many an hour.

Unfortunately, like so much software, it is marred by poor documentation – a shame, for what is otherwise an excellent product.



Now let's see how functions function

Part VIII of our teach-yourself series by GORDON FINDLAY

IN the last part of this series we discussed procedures, subprograms which may be written and, most importantly, tested separately.

Many programmers find it useful to build up a collection of useful procedures as text files on disc, so that they may be included in other programs using the editor C(opy) from F(ile) option.

A more sophisticated and much faster way is to build up a library file, such as SYSTEM.LIBRARY, and include in it the units – collections of procedures – which you find useful.

The details of how to do this are all in the Pascal Language Manual, but might be a bit overwhelming at first. In the meantime, storing the text on disc might be a viable alternative.

There is another sort of 'subprogram' in Pascal, the function. These differ from procedures in that they return a value through the function identifier itself, rather than through a parameter.

You will already have used many functions, such as SQR or SCREENBIT.

A function must of course be declared, and is declared with the type of the value to be calculated:

```
FUNCTION CUBE(X:REAL):REAL;
```

declares a function which has a real parameter, and returns a real value. The function may be used in exactly the same way that any of the built in ('intrinsic') functions are:

```
X := CUBE(1.2);
IF CUBE(Y) > 10 THEN ....
Y := CUBE(SQR(X) + SQR(Z) +
  CUBE(2));
```

(Obviously the identifiers will have been declared as real.)

The body of a function will look exactly like the body of a procedure, with the addition of an assignment to the function identifier near the end.

Here is a complete function which merely cubes its parameter:

```
FUNCTION CUBE(X:REAL):REAL
  VAR Y:REAL;
  BEGIN
    Y := SQR(X)*X;
    CUBE := Y
  END
```

The local variable Y isn't really needed here, but it is good practice to use a local variable and make one final assignment to the function identifier. This will help to keep errors down when alterations are needed.

Functions differ in one important respect from the more general procedures. A function must return just one value to the "calling" program. A procedure can return none, or as many as you like, through the parameters.

Boolean functions are useful. Here is an example which reads a character from the keyboard, and returns TRUE if the character is Y for yes, whether in upper or lower case:

```
FUNCTION YES:BOOLEAN;
 VAR CH:CHAR;
 BEGIN
  READ(CH);
  YES := (CH='Y') OR (CH='y')
 END;
```

This function has no parameters. It may be used like this:

```
WRITE('Any more transactions? (Y/N)');
IF YES THEN....
```

As you can see, a function can perform input or output. It may in fact do almost anything.

As a final example, here is a function which goes some way to meeting the need for an exponent operation, normally missing from Pascal:

```
FUNCTION POWER(X:REAL; I:INTEGER):REAL;
VAR PW : REAL;
    J : INTEGER;
BEGIN
   PW := 1;
   J := ABS(I);
   IF I<>0
      THEN FOR COUNT := 1 TO J DO PW := PW*X;
   IF I<0 THEN PW := 1.0/PW;
   POWER := PW
END;
```

The statement WRITELN (POWER(2,–3)) ought to output 0.125.

Now, as the saying is, for something completely different. Arrays are probably familiar to you from Basic, and of course they are just as useful in Pascal.

There may be some who haven't encountered arrays before, so as well as describing the particularly Pascal features, I will mention the general use of arrays.

An array is a collection of related variables, in the form of a list. Pictorially it might look something like Figure I on the next page.

This array is a collection of integers. The individual elements are referenced by number, or 'subscript', to introduce the jargon. In the array A[4] has the value 12.

The Apple II Plus has no square bracket keys. Use Shift–M for the right bracket (]) and Ctrl–K for the left bracket ([).

An analogy might help here. Some houses, rather special ones, have their own name as an address such as Buckingham Palace and Westminster Cathedral.

Other houses are referred to as members of a street – such as 37 High Street.

Here High Street is an array, and the particular house is selected by number, or subscript.

Simple variables are like the special houses, array elements like the more humble dwellings.

Arrays are used for storing a lot of related values, such as the prices of many items, the price of one item over many years, the locations of all the men in a game of space invaders or whatever.

A payroll program might well store each employee's name in an array of strings. Arrays may be of virtually any type – arrays of Booleans seem to occur frequently.

Before using an array it is necessary to tell the micro – the compiler really – that an identifier names an array, what the possible subscripts are, and

what type each element in the array is. This is done in the declaration part like this:

```
VAR AR : ARRAY[10..60] OF REAL;
```

The subscripts must, for now anyway, be integers. The range is indicated by giving the lowest and highest values permitted. An attempt to use AR[3] for example, would lead to an execution error.

It is very rare that an array can be operated on as a whole, by using just the array name. The usefulness of arrays arises from the possibility of writing a loop to pass through the array using a variable subscript.

Here is how to clear an array to zeros:

```
VAR AR[1..300] OF INTEGER;
    I : INTEGER;
BEGIN
  FOR I := 1 TO 300 DO AR[I] := 0;
      : : : : : :
```

This is much shorter than writing 300 separate variable names:

```
AR1 := 0; AR2 := 0;...AR300 := 0;
```

Finding the sum of all the members of an array is equally easy:

```
TOTAL := 0;
FOR I := -20 TO 30 DO TOTAL := TOTAL + X[I];
```

An array may have more than one dimension. Figure II is a picture of a four by three arrray.

This array would be declared, if it was to contain integers, as:

```
VAR A : ARRAY[1..4,1..3] OF INTEGER;
```

The first subscript, giving the row number, can vary between one and four; the second, giving the column number, between one and three.

It usually doesn't matter which you think of as the row or which as the column subscript so long as you are consistent in a program.

The main uses of two dimensional arrays are to represent mathematical matrices and tables of information.

There is a major restriction in the use of arrays which you ought to know about, particularly if you have ever programmed in Basic. The size

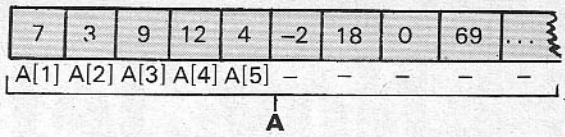


Figure I: An array A. showing some of its elements.

of an array is fixed at compile time, not during execution.

This piece of Basic has no Pascal equivalent:

```
INPUT N
DIM AR(N)
```

Pascal has no ability even to use a variable dimension in a procedure call, such as Fortran does.

The cure of course is simple – dimension the array at a maximum value, and when the actual size is known (by reading it for example) use this value as the limits in all the loops.

Arrays are one of the most powerful data structures – so powerful that almost every programming language and all useful programming languages support arrays even if they have no other form of data structure.

Some of the very exotic chips now being produced – the "micromainframes" – even support array operations directly in hardware, so you can see how important they are.

Pascal has many other data structures, some of which we will meet next month, but arrays are the most vital.

A[1,1]	A[1,2]	A[1,3]
A[2,1]	A[2,2]	A[2,3]
A[3,1]	A[3,2]	A[3,3]
A[4,1]	A[4,2]	A[3,4]

Figure II: A two-dimensional array A

appletip

I THOUGHT you might be interested in a little tip for spreadsheet users I have been using for some time.

Unless you have an expensive utility program, it is not possible to see and print the formulae except in a vertical column. Well there is a simple way.

Save your spreadsheet and exit the spreadsheet program. Boot up your word processor such as Applewriter II or IIe and load the spreadsheet file.

You will see a vertical list of the cells like this:

```
>A 1 : /FI 12345
>A 2 : /F$(A2 * A3)
>A 3 : 1234
>A 4 : /--
>A 5 : /@SUM(A1...A3)
etc. . .
```

Then, using the global search and replace (Ctrl F in Applewriter), type **Ctrl F /:/:"/a**

This will place a double quote in front of every formula. It only remains to delete the additional quote in the label cells by another **Ctrl F /""/"/** then save the altered file as Filename. FORMULA – to differentiate it from the working file.

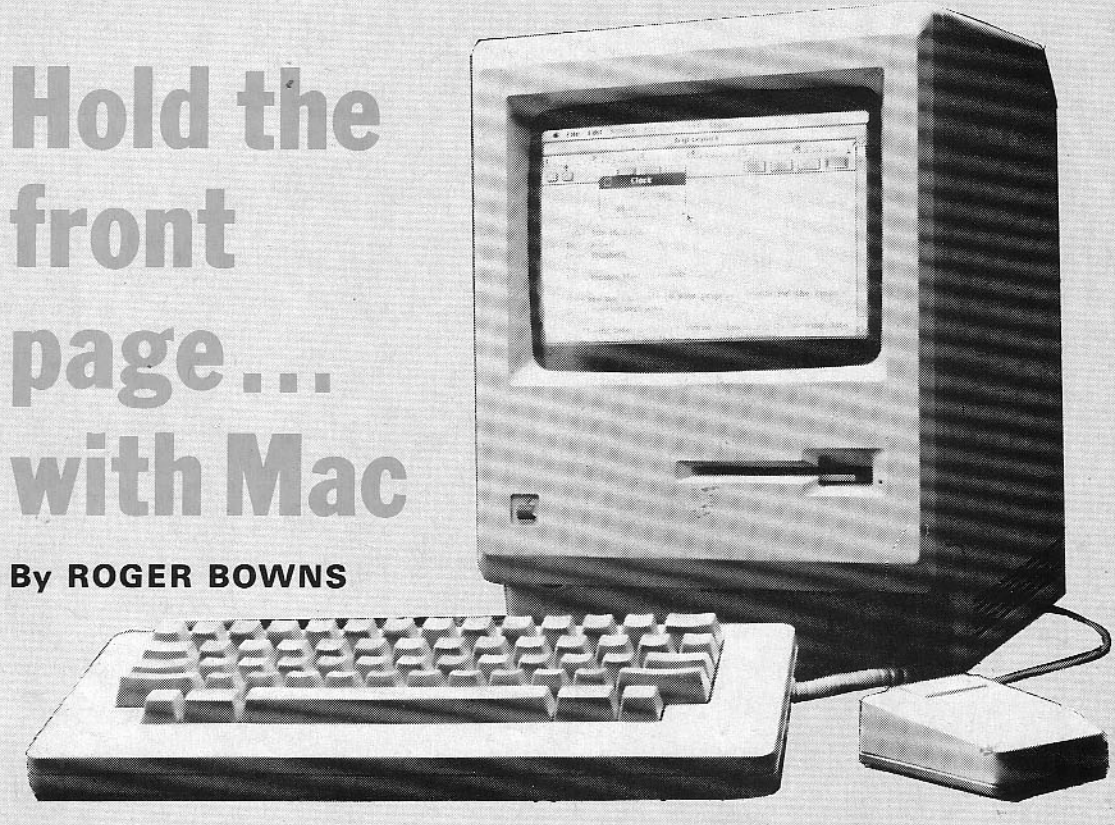
Exit Applewriter and boot the spreadsheet, load the Filename. FORMULA and reset the columns so that they are wide enough to see the whole formula.

Owners of the Magicalc spreadsheet have an advantage here over users of Visicalc. You can then print the sheet and see all the formulae in their exact locations.

Peter Trinder

MacWrite

Hold the front page... with Mac



By ROGER BOWNS

MUCH has been said and written about the powers and uses of word processing to the working journalist, but I fear that too much of it may have come from salesmen and people who don't really know scribes.

So as a working journalist myself with more than 30 years experience, I thought it might be useful for all of you other computer users to try to understand why many of the people who could make the best use of a word processing package are still shying away.

There are two major stumbling blocks which I suspect computer salesmen have not yet spotted when they try to twist the arm of the Press.

For a start, the everyday reporter is still attached to his own portable typewriter. Like myself, most of them bought a sturdy portable during their training days and it has become as comfortable as a pair of old gloves.

It would be quite unthinkable to desert such a friend which has accompanied us to many parts of the world with international airline lables still clinging to the casing.

Second, most journalists are not the slightest bit technically inclined. This may not be true of the *Apple User* reporting staff, but the general newsman does not in the main give a hoot about subroutines, return keys and floppies. The thing is, you see, he's lived with the telephone or behind his trusty typing machine.

But conversion can be achieved. It happened to me – the hard way I admit. And within the last few weeks I have discovered that if anyone has the key to a journalist's attention then it has to be Apple through the unique Macintosh.

Let me tell you my story because it will answer a lot of the reasons why reporters are not being easily won over to the micro.

I began to see reason, albeit slowly, when I was lucky enough to be spending a month in California about three years ago. It made me quite worried, though, when I saw American kids talking to Tandy computers in the Radio Shack shops with such contemptible ease.

So I was more than a little interested when I went to stay with a good friend just outside of San Francisco.

Now David lives in the Portola Valley which is just an apple's throw from Palo Alto, so it was hardly surprising that he had taken delivery of an Apple II to handle his home business affairs which were and still are quite considerable.

He was patient with me and launched me gently enough into Apple to send me back home thinking and wondering.

Several months later I had an Apple II Europlus together with twin disc drives, a monitor and an NEC printer.

Armed with Applewriter II, I set about undertaking the biggest challenge I've ever faced. Fortunately I was also advised to include an 80 column card and upper and lower case facilities in my fascinating new system.

It was no easy hurdle, but now as far as word processing is concerned I am the master of machine and I'm quite ready to try to convince other writers that word processing and a good database file are essential ingredients for putting some order into the rag-bag mind of people who collect information like a hedgehog picks up fleas.

It was a terrible battle. I don't suppose it was helped by the fact that my home is in Guernsey in the Channel Islands where computer advice wasn't readily available without pestering the few people who were trying to pioneer the introduction of the micro age into a small island.

Today, I'm thankful to say, we have a superb, if not the best Apple presence possible in Dynatech who give instant advice on all matters.

My thanks go out to John Chubb, who convinced me that Apple and I had a future, and to Caroline Craig and Martin Thwaite, who both quite rightly lured me away from my Apple II in favour of Apple Mac.

But in the beginning it became so worrying that I even went to a computer workshop at our local College of Further Education. That didn't help me at all.

In fact it only confused my muddled brain even more because my interest was always word processing. Loops and strings simply didn't fit into any of my plans.

Yet it slowly came to me. Armed with badly written manuals (have you ever seen the one which comes with the NEC 8023 printer?) and seeking help over the telephone from as far away as Pace in Bradford, I managed to format my text to my wishes and eventually convert the whole lot to the printer.

Finding the commands to make the printer work was a living nightmare and even look-

MacWrite

ing back today I'm still not convinced I had it all right by a long way!

I cannot honestly say that it's any quicker to write a story or article on a word processor, but the end product is so much cleaner and more professional. And the storage system of using discs rather than dozens of folders is absolutely unbeatable.

It's a funny thing, you might think, but despite my technical inability I do sometimes contribute computer features to our company television programme journal.

You would also be quite right in thinking that it must be almost totally non-technical, but I do try to give a fair appraisal of whatever I'm trying out and then fit it into its place of usefulness.

For instance, I have had some systems for review which are no earthly good to someone like me but invaluable to an accountant.

And I hope that by using my simple descriptions rather than the salesman's highly technical approach I may have encouraged some new users to the fold.

It was through these features that I met the Apple IIc recently. I had the machine home for a weekend and was instantly impressed, though disappointed that Applewriter II and some other programmes would not run as expected on the new unit.

However I felt that this portable computer was just the sort of thing a journalist should be looking for. And then I discovered Macintosh.

Now it may not be too portable, and it may take entirely different sized discs, but its ease of use puts it into a class way out on its own.

I am convinced there is nothing finer for word processing, largely because of the typewriter clarity of MacWrite and thanks to the mighty mouse getting everything done is so easy and quick without all those cumbersome commands to worry about. It is to be presumed that eventually everyone else will have the sense to make computers this easy to operate.

I've only had my Mac for a few weeks but already it has become a true friend. It has also been packed up in its bag and taken to the office where I can have it set up and running within minutes.

But there is one thing I wish I could do with Mac as long as it remains a 22lb mains operated system.

I'd like, one day, to be able to buy a small lap portable which could travel with me so that I could file and print when I get home.

It may be that the IIc is getting close to that goal, but it is still not the answer, especially with its separate power pack.

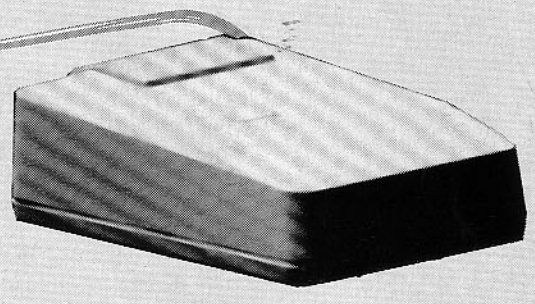
Apple could win themselves a lot of friends if and when they ever got round to this.

Each month more and more newsrooms up and down the country are changing over to computers, and as that happens I think it's a fair bet that the journalist will buy for himself to maintain his independence.

But I am sure that the accent will be on true portability and extreme ease of use. And from my investigations that sort of deal is still pretty thin on the ground.

I have tried one or two of the battery-driven portables, most of which are admirable. But they do lack compatibility. And I doubt whether the IIc with its separate battery pack and clip-on flat screen will really be the answer.

Someone like NEC almost has the answer with a machine that comes out for less than £500, but in my case it lacks compatibility with an Apple and that's no good at all.



However I remain full of praise for my Mac because I knew what I was buying at the time and it answers all my needs.

Finally, as a wordsmith, may I congratulate all those responsible for computerese, that new language which is quietly slipping into our daily vocabulary.

How descriptive "user friendly" is, but I've never really grasped what "state of the art" is supposed to mean.

But having said that, let me also add that as a staunch defender of the English language I cannot condone the spelling of disc with a "k" or leaving "me" off the end of programme. And colour has a "u" in it.

I have accepted the change to computer ownership with considerable patience and grace, but I cannot possibly allow myself to come to terms with incorrect spelling, even if the reason for this is obvious.

Oh well, never mind! I must get back to my Macintosh. Come to think of it, didn't that used to have a "k" in it once upon a time?

***ROGER BOUNDS** is station manager and senior reporter at Channel Television's satellite station on the island of Guernsey. He and his team are responsible for supplying 40 per cent of Channel TV's news output.*

MacSoftware

Lisa helps in Mac software development

AT PRESENT anyone wishing to develop fully functional applications software for Macintosh in Pascal has to use Lisa.

With 1mbyte of RAM and 5 or 10mbytes of disc, Lisa provides a very powerful development environment which is similar in appearance and operation to any other UCSD development system.

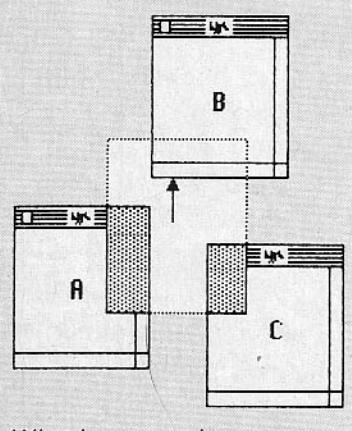
One key feature is the window-based mouse editor, which allows multiple programs to be resident and visible at any time, each in its own window. An author can cut/copy/paste from window to window.

The resulting programs are compiled and linked on Lisa and then saved on Macintosh disc placed in the Lisa drive, or sent serially down a communications line at 9600 baud to a waiting Macintosh.

Writing for Macintosh does not require learning any new Pascal extensions. Standard Pascal control structures, procedural and functional calls allow full use of icons, pull-downs, controls, multiple overlapping windows and graphics.

"Events" are fundamental to the way Macintosh programs run. They come from a number of sources and may be split into two categories – user events and system events.

Keyboard and mouse-button activities constitute user events; system events can be best explained with the diagram below. Consider the situation where one window, overlapping two others, is moved by the user.



What happens when you move an overlapping window

The result is two holes in the windows that were underneath, and the window manager tells the application that something has happened.

The sequence of 'messages' would be as follows:

- Mouseclick in window B – user event.
- Drag window B – user event.
- Update event for window A – system event.
- Update event for window C – system event

Hence, the structure of your application must handle this type of situation. This is known as event driven software as opposed to the more usual linear flow variety.

A program will typically consist of a main loop looking for events which are then processed accordingly when identified.

Application program files are given a type and a signature, a four character, unique identity. For example, MacPaint's type is APPL (application), and it's signature is MPNT.

When MacPaint creates files it marks them with a type PNTG, and signs them MPNT. By signing a file with the creator's signature the user can open a document on the desktop and the desktop manager will launch the application program that created that document and then tell the application which document was activated.

This means that if there exists a MacPaint document, say FloorPlan, the user can open that document directly without having to run the application and then load the document required.

An application may also hold specifications of icons that are to be used when the destop manager displays files of a given type. An application can use file types to search for and display to the user a list or choice of, let's say, MacPaint documents to choose from.

On Macintosh a file is a file is a file. All files – or documents – are identical in structure and are called resource files. Externally they appear as a single entity. Internally however they have two portions, a data fork and a resource fork.

If a file is accessed using standard Pascal techniques, the data fork will be accessed. If, however, the resource manager routines are used the resource fork is accessed. A resource can be almost anything from a data structure, menu definitions, window definitions, text strings, control definitions, pieces of application code etc.

The advantages of this are several fold. Firstly since language-dependent items such as control titles and strings can be factored out from the code, translation is much simpler.

Development time is reduced since after making changes the programmer does have to go through the entire process of re-compilation and linking, only the final phase of bringing the code and resource definitions together.

Apple supply the tools for changing resources actually on a Macintosh without the need to access the source code. Thus programmers spend less time making windows and controls etc. look pretty. They can do that when the program is finished.

Since even an application is a resource file, if the programmer is writing a suite of programs, all of which use a common menu say, one program can open another program's resource fork and use its resources.

So the programmer can build central libraries of, not just pieces of code, but other common objects too. Resources can behave as purgeable relocateable objects. Using this to advantage, Macintosh can manipulate many large data objects at one time with minimal supervision from the application.

Among the list of tools that Apple supply are utilities for designing icons and fonts, moving fonts and other resources from file to file, document manipulation utilities for

EVERY day we are deluged with more lists of Apple II programs converted for the Macintosh. Many of the lists mention games programs, and I'm delighted to report that I finally got my sticky hands on one.

Regular readers will know that Penguin are one of my favourite software houses – even though they have put their prices back up. They beat all the others to the punch recently (marketing managers please note) by sending me the Mac version of Transylvania.

I reviewed the Apple II version in the January 1983 issue of *Windfall* (vol. 2, no. 7) and since the game is the same I won't bother going into detail again.

If you don't have that issue (shame on you!), suffice to say that the scenario is 'Hammer Horror Film' country and my review was positive.

As you can see from the screen dump, the Mac screen is split into four distinct sections. The upper left-hand quadrant contains the picture.

These are the same images

hiding and protecting files and setting file types.

The following programming examples indicate how, from standard Pascal, the programmer can manipulate complex Macintosh objects:

Drawing a rectangle:

myRect:RECT;

Simply use the procedure: setRect(myRect, 10, 10, 100, 100); and pass it the name of the rectangle to set up with coordinates for the left, top, right and bottom then frame it . . .

frameRect (myRect);

Opening a window:

myWindow: WindowPtr;

A function exists to create a window. Parameters are passed to indicate where the window record is to be stored, a rectangle to define the boundary of the window, a title, whether or not the window is visible or invisible, the type of window, and whether it is to appear in front of or behind all other existing windows:

myWindow:= newWindow (wRec, myRect, title, visible, type, orientation,...);

Closing a window:

To dispose of a window, the programmer simply calls the procedure closeWindow...

closeWindow(myWindow);

Creating a control such as a scroll bar, the process is almost identical to creating a window

myControl: =newControl (the Window, myRect, title, visible, value, type,...);

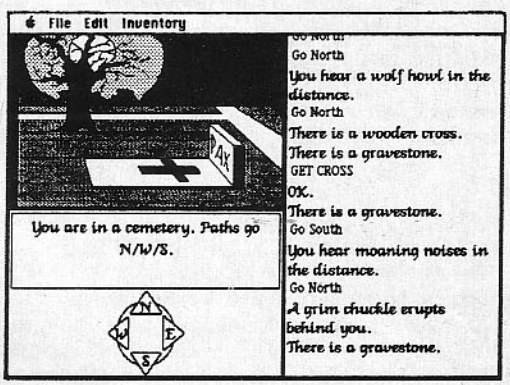
The similarity between procedures and functions to manipulate objects of different types is not an accident. Although it obviously helps the programmer, there is still a lot to learn. The technical programming material for Macintosh now amounts to over 900 pages of detailed information.

It is not within the scope of this article to discuss the reasons why Apple chose Pascal as the prime development language for Macintosh and Lisa.

Suffice to say that it is reasonably well suited to handling complex data structures, it is easy to learn, robust, and since it compiles all the way down to object code, is very fast in execution.

Nigel Hearne
Macintosh Software Development Manager, Apple UK.

MacAdventure



as the II version but they are now limited to black-and-white, and are smoother because of the increased resolution.

The right-hand half of the screen is a scrolling text window in which is printed the commands and additional descriptions. Underneath the picture is a rectangle containing the frame title and the possible exit directions.

So what's that funny thing underneath the frame title? That's where the mouse comes in.

You can move around Mac Transylvania by moving the mouse arrow onto the appropriate compass point and clicking. It's as simple as that.

As you click on 'N', the words "Go North" appear in the text window on the right and the next picture appears. Of course, if you want to issue anything other than a direction command you must use the keyboard.

You can see my manually-entered 'GET CROSS' command in upper case half-way down the window.

Actually, that's not quite true. The eagle-eyed among you will have noticed the word "Inventory" at the top of the screen. This is a mouse-controlled pull-down which displays all your worldly goods.

The commands to initiate a save or restore are also issued via the mouse using the 'File' pull-down. The 'Edit' menu offers the usual cut/copy/paste options, but I didn't manage to do anything useful with them.

So what do I think of Mac Transylvania? Novelty value aside, I still think the adventure is a good one.

I'm not sure that an adventure game is the ideal thing to be ported to the Mac because most of the input is textual. Unless you want to do little more than wander around, the mouse could be dispensed with.

It's useful having a bigger text window – the reduction in the size of the graphics window wasn't really noticeable. However, I'm looking forward to the arrival of some games specifically tailored to the Mac.

Cliff McKnight

Title: Transylvania
Author: Antonio Antiochia/ Mac version by Robert Hardy
Publisher: Penguin Software

AT last, linking Lisa to an ICL mainframe and capturing of data in a saved form available for instant rapid processing by its own multi-tasking software has become a reality.

CHARLES FAIRFAX and his assistant TONY PARSONS describe the latest advance in the development of micro technology in the North Western Regional Health Authority Emergency Planning Department.

Lisa talks to ICL

SINCE interest has become focussed on the development of micros for the work of our department, many and varied questions have come our way.

The enquiry made most frequently relates to the possibility of our Lisa II at H.Q. in the centre of Manchester and our ICL 2966 mainframe at Prestwich talking to one another.

Hardly had we whispered our intentions in this respect than there began an onslaught of gratuitous advice.

Heads shook sideways – the task was considered too difficult. We, mere ignorant amateurs in computer work, could not expect to achieve what had up till now defeated the professional brains.

The solution was still a long way off. To attempt this using our Apple micros was a mistake, far better throw them away and use our DRS micros designed by the same manufacturer to work with their own mainframes.

This was the sort of discouraging advice we received in such an incredible torrent one almost began to question its motivation. We thanked our advisers politely but continued to take our own view of the problem.

We regarded it as being quite straightforward, certainly theoretically soluble. Underterred by the "wiser heads" we determined to convert our theory to practice. This we have now done.

Our starting point, being unfettered by any commercial considerations and free to choose from all available sources, was to examine the possibility of using only ICL equipment. This did seem to have a simple basic logical appeal.

Our eyes were soon to be opened. The DRS micros that may have appeared to be reasonable on superficial enquiry were found to be no use for our type of working on the slightest depth of probing. They were by our standards very user unfriendly.

A long time passed before they were in fact linked to their own mainframe. Their system of data capture and the processing of data received seemed too under developed for ease of use. One felt they were far too much programmer rather than user orientated in their design concept.

This meant that much was still done by old outdated methods. The tables and graphs, for example, continued to be hand drawn. Multitasking, as had become our normal practice using Lisa integrated software, was beyond the limited capabilities of the available system.

The DRS was hard work – it did not lend itself to ease of imaginative manipulation.

After trying our best to induce ICL to come up with the appropriate software we decided to go for the best of both worlds by hybridisation, Apple/ICL.

The system we devised used the Saturn enhanced Apple II described in an earlier *Apple User* article for accessing the mainframe, thence networking to Lisa. This operated in fully transparent mode with Apple II type software.

We could as readily have chosen an Apple IIe, which has similar software available. Likewise in place of Lisa we might have chosen any of a wide range of possible micros, not forgetting the Macintosh that was due to arrive on the scene at any day.

Our approach made the best of the large data storage capacity of the mainframe and the brilliance of captured data handling that we had now come to associate with Lisa.

We were told at the outset all we might hope for was to be able to view a screenful of data from the mainframe. The use of different protocols would prevent any usable data capture. We would not be able to process the data even if we could print it.

We put it to the test and found this to be quite untrue.

With the cooperation of Apple UK, ICL, Ultracomp and Style we set up a system at the Prestwich Computer Centre where the mainframe is located at the kind invitation of the computer manager and his staff.

The hardware of the Saturn Enhanced Apple II Europlus was adapted by the insertion of a Multicom Card to drive the synchronous mainframe link, and a California Computer Systems 7710-A Asynchronous Serial Interface to drive the asynchronous Lisa link.

The software comprised an Apple compatible floppy with automatically loading programs and data files. These came in a comprehensive Ultracom 3 communications package.

Our Saturn already had a Videx card, without which we would have needed to fit an 80 column card.

The slot selection for the extra cards does not matter. They work equally in any slots, although it is best to keep to Slot 1 for the printer interface card.

We used a single Lisa by choice, but equally we could have used three Lisas or their alternative equivalents in other micros which would need to support the necessary subset of the DEC VT 100 protocol.

If it has not already been done, LisaTerminal requires

Mac arrives

SINCE this article was written the Macintosh has arrived at NWRHA and will be used increasingly in the work there in conjunction with Lisa.

Ease of movement between the two systems will be discussed in a future article.

installation. The communications link is controlled from Lisa by LisaTerminal and physically uses one of the two serial interfaces which has to be selected from device connections in the preferences window.

Lisa needs to be configured compatibly with the particular physical connection in use and with Ultracom 3.

The first setup is done by opening a sheet of LisaTerminal stationery, pulling down the setup menu and selecting computer compatibility. Once the appropriate settings have been entered, this does not require repeating.

For convenience, a LisaTerminal stationery pad can be created for ease of making a series of new documents.

Our spacing of equipment was compact, but we could have separated our micros by up to 50 feet or so, simply by cable. Had we wished, say, to use a telephone link back to our headquarters in the centre of Manchester, a modem would have been required.

The Ultracom 3 emulates all the normal ICL terminal features, and many ways of using them are possible.

We concentrated for the purpose of the exercise on producing a screendump off Lisa and the capture of data for processing using the Lisa capability in the same way as if the data had been fed direct into the micro in the first place.

Our example is typical of a quite common service planning application. Figure I shows part of a table of the data relating to the admissions of children to hospitals in two of the districts in the region.

This is first displayed on a document created from a LisaWrite Toolpad. Anxieties have been expressed over the supposed difficulty of transferring this data into other usable systems. The cut and paste sequence does this admirably.

The table shown could of course have been incorporated directly into a written report or letter on LisaWrite.

It is often necessary to construct and edit graphs, adding and subtracting information and displaying the results in various styles and formats. This is accomplished by using cut and paste to and from LisaGraph and LisaDraw.

Producing parallel documents on a split screen as we described in a previous article in *Apple User* is a great deal of help.

It is important to observe a simple rule to paste into a similar area as the cut. This need not be piecemeal. Once in graph mode, the whole table or graph can be selected for transfer.

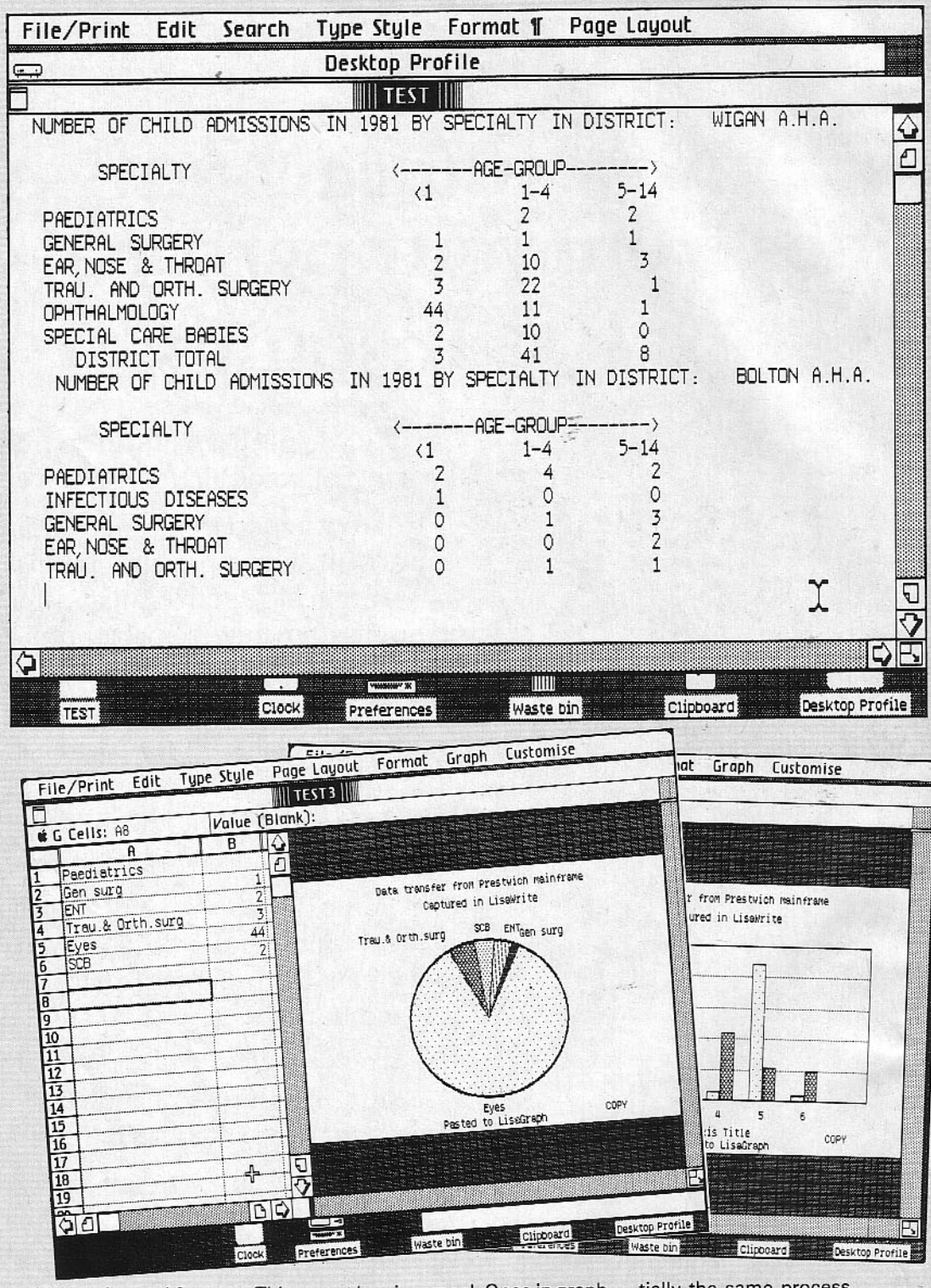
The use of copy in place of cut allows previous information to be retained in its old form as well as in the new, should both be required at some future date.

For our experiment we did not bother with LisaCalc. But interchange with this is just as easy following what is essentially the same process.

ICL use a 25 instead of a 24 line display. Window control takes care of that, moving a 25 line display within a 24 line window.

We concentrated on the use of a single window, as this is the normal way we use our micros.

For some purposes file to file transfer is desirable. This will follow.



ROBIN HUDSON says: Forget the past, get on to the right number and . . .

Serialise, modulate, communicate!

COMMUNICATIONS is, perhaps, the most interesting area of microcomputing that has yet to be explored by the majority of home computer users.

For many years now the public telephone network has been used to transfer data between computers which are often many hundreds of miles apart.

For companies using computers in this way the advantages are obvious. Large volumes of data, including programs, can be transferred between various establishments very rapidly and without the necessity to physically transport discs, tape or any other storage media.

It is only recently however that the additional equipment – both hardware and software – that was needed to take advantage of the telephone network in this way has become sufficiently inexpensive to be used widely in the home.

So let's examine the possibility of using the Apple to communicate via the telephone system.

There are many advantages in doing so.

One major application is the ability to access data which is held on other, often very large, computer systems. The best known example of this is British Telecom's Prestel service, although many other public database systems exist.

This is an area which is currently arousing great interest, and which will no doubt become one of the more significant aspects of computing in the future.

The transfer of files to other computers, in the same way as is done between mainframe installations, is another area that has great potential in the home.

You've just written a new utility program which you would like a friend, who happens to live at the other end of the country, to test for you.

Instead of sending a fragile disc, which could take days to arrive, you could simply send a copy of the file over the telephone in the space of a phone call.

Another type of system, the "bulletin board" is peculiar to micros. Bulletin boards were first developed in America where micro communications has progressed to a more advanced state than in Europe.

It is simply a micro which accepts telephone calls from other micro users and allows them to access its files, download programs, and, as with electronic mail systems, leave messages for other callers.

It is quite common to find that such systems operate on a worldwide basis with users calling from countries as far apart as Australia, Britain and America.

These are just a few of the possible applications. But what about the equipment required to use the telephone network in this way?

Fortunately the rapid development of large and very large scale integration technology in recent years has made it possible to produce suitable equipment at prices which are no longer prohibitive.

The obvious starting point is some form of interface, in the same way that a disc interface is required before data can be stored onto and retrieved from a disc drive.

In the case of the Apple, this takes the form of a serial interface card.

The reason for using a serial card, as opposed to a parallel card such as that used with Centronics-type printers, is fundamental to the whole area of long distance transmission of data.

Since the telephone network uses only two, three or four wire circuits, data must be transferred in serial form, one bit at a time, in sequence.

However data within a computer is transferred between different components in parallel. This means that in an 8 bit machine such as the Apple, eight individual wires, known collectively as the data bus, are used to move information around a byte at a time. This is termed parallel transfer because eight bits are transferred simultaneously.

We have noted that the primary reason for serialising data for transmission over the telephone network is simply that there are insufficient wires to transmit eight bits at a time.

Indeed, it would be extremely expensive to lay eight cables instead of one, and even if this were possible there are further complications with respect to parallel data transfer.

These revolve around the fact that the individual bits of data tend to travel at different speeds within the wires and introduce what is known as data skew. The result is that the data becomes garbled.

This effect, illustrated in Figure I, is more noticeable over

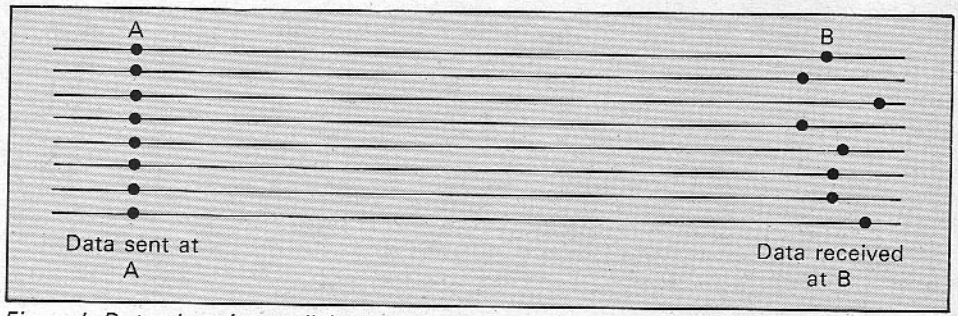


Figure I: Data skew in parallel transfer

greater distances. It is one of the reasons that data cables used with parallel printers, or disc drives, are rarely longer than about a metre.

We see then that the function of a serial interface is to convert data from a computer into serial form so that it may be transmitted via a circuit consisting of as little as two wires.

The parallel transfer of n data bits requires n+1 individual wires, n wires for the data and one ground line.

In addition to the obvious economies of using fewer wires, serial data may be transmitted over longer distances than is possible with a parallel interface due to the absence of the skew effect in serial circuits.

What are the other components of a communications system?

Consider the type of signals that a computer generates. These are digital in nature. The 1s and 0s are represented by two separate and distinct voltage or current levels, generally the former.

The public telephone network however was developed for a different application – the transmission of the human voice which is analogue, as opposed to digital, in nature.

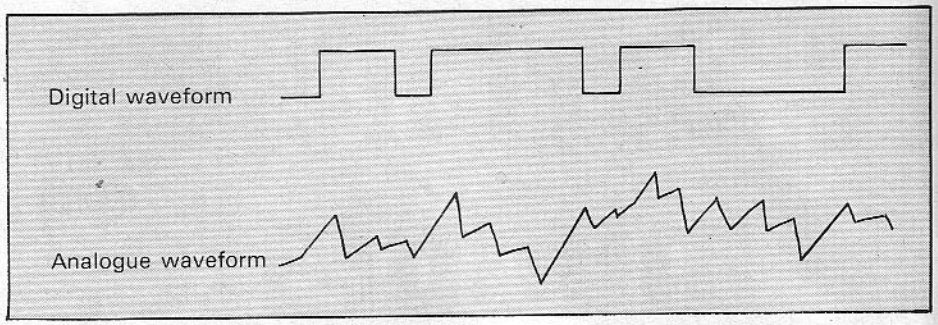


Figure II: Analogue and digital waveforms

The difference between the two types of signal, analogue and digital, can be seen in Figure II.

The point is that the square waveform produced by digital computers will not pass through the telephone network because of the nature of the network itself and various filters and switching equipment used in telephone exchanges.

To overcome this problem a further piece of equipment is required that will convert digital data into analogue form so that it can then be transmitted via the telephone network.

This task, termed modulation, is the function of a modem.

In practice a modem carries out two jobs. Digital data for transmission is modulated into analogue form before being passed to the telephone system. Conversely, received analogue data is demodulated into digital form before being given to the computer.

Hence a modem is used to both *MO*dulate and *DEM*odulate electrical signals.

The final element involved is the software.

The facilities offered by communications software may vary greatly, but essentially it provides the means by which the serial interfaces, and the data passed between them, can be controlled.

Just as a disc filing or disc management system is used to store and retrieve information from disc, communications software may be regarded as the communications management system.

Thus, although the hardware provides the physical means by which data and files can be transferred, some form of protocol is required in order that each computer can understand and use the signals it receives from the other.

These protocols may be regarded as the equivalent of word, sentence and paragraph structure in a spoken language.

In the case of written text punctuation is used in order to ensure that it makes sense to the reader. Similarly a protocol structure of some form is required to allow two computers to communicate sensibly with each other.

It is the software that provides this structure, along with the facilities for storing and manipulating received data or data to be transmitted.

So we can see that three basic elements, apart from the computer, must be present in a communications system using the telephone network – a serial interface, a modem and the appropriate software.

On a smaller scale, within a particular room for example, it is perfectly possible to link the two computers together without using the telephone system.

In this case a direct wire link between the two serial interfaces is sufficient to allow communication to take place and no modulation is required. Provided that the distance between the computers is not too great there should be no problems.

Using a hard-wired link in this way, however, reveals a further advantage of using the telephone system.

When a hard-wired connection is used only the two computers involved can communicate with each other. The telephone system, however, offers far greater flexibility in that any two computers with access to a phone may be linked.

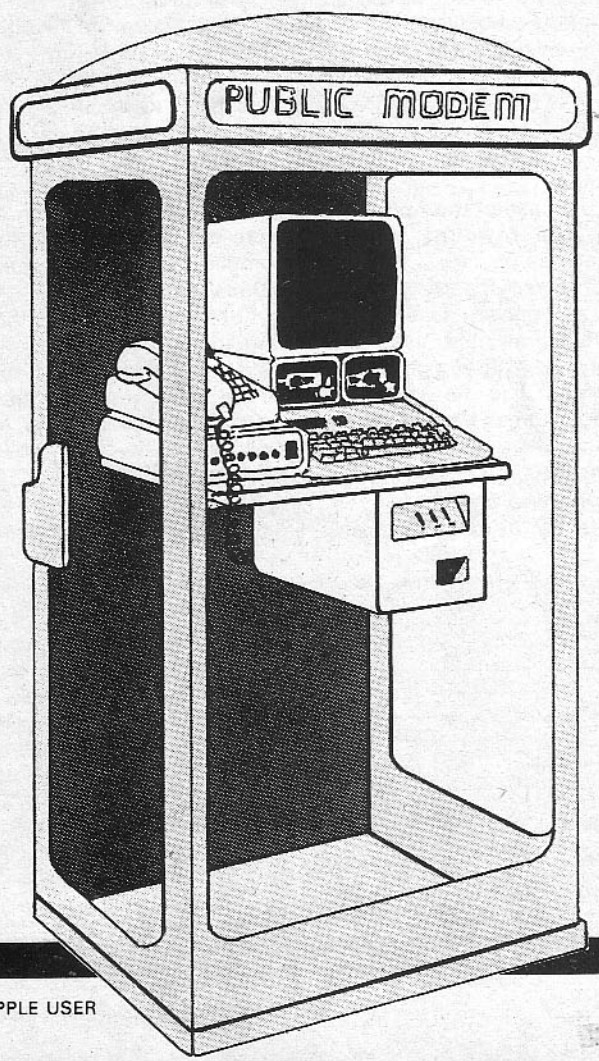
As we have said, the Apple needs a serial interface card installing, and there are several of these available.

Modems themselves may be obtained with various levels of sophistication, ranging in price from as little as £70 to several hundred pounds.

Finally, there are a number of software packages available to suit the needs of almost any application.

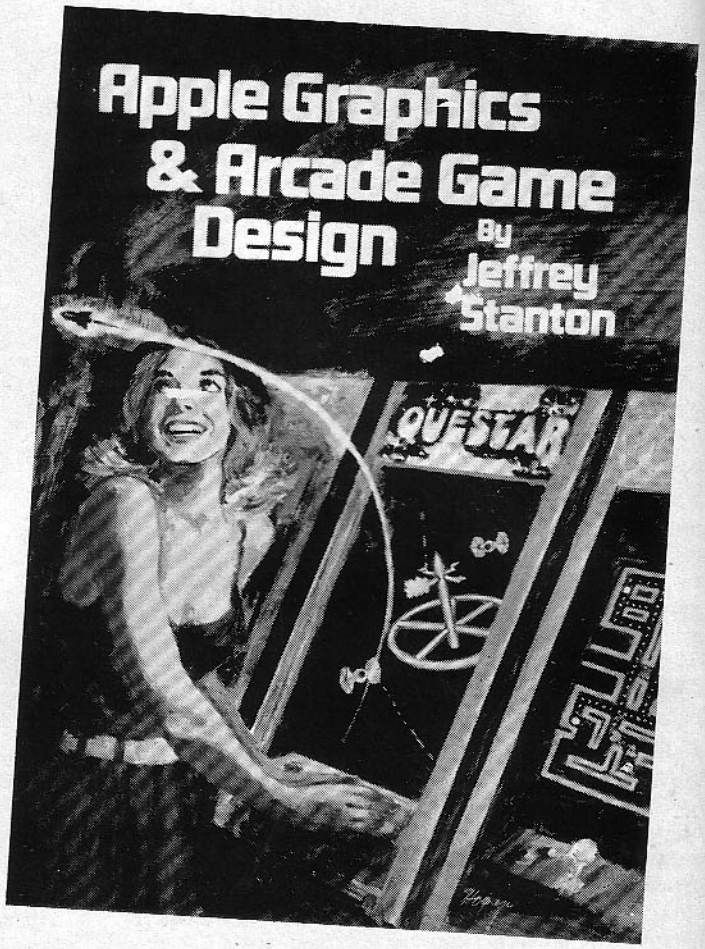
Communication between your Apple and almost any other type of mainframe, mini or micro is possible, so give it a try.

Serialise, modulate and communicate!



Shaping up with a better routine

By JASON W. SMITH



ALTHOUGH Jeffrey Stanton's book *Apple Graphics and Arcade Game Design* is excellent, there are a number of errors.

The worst is on Page 142 where the routine printed is supposed to draw a shape while saving the background and ensuring that this doesn't interfere with the shape, but surrounds it.

The method Mr Stanton describes on Page 140 simply recaps the algorithm on Page 132. However, this method isn't actually used in the listing given.

The listing doesn't give very good results, as you will find if you place a green background on the screen before running the routine.

The changes required to make the routine work properly are given here.

Only three new variables are needed:

New variables

MSHPL EQU $F8
Lo-byte of mask shape.

MSHPH EQU $F9
Hi-byte of mask shape.

MASKSHP EQU $8000
Location where shape starts.

MASKSHP can be any location you desire – it is where the actual mask of the shape is stored. For the rocket used, the mask shape is:

```
01  00  00
03  00  00
07  00  00
0F  00  00
7F  7F  00
7F  1F  07
7F  7F  1F
78  7F  7F
```

The new listing given here starts from SDRAW2. ">" means it is a new addition.

```
SDRAW2  LDA (HIRESL),Y   ;Load byte from screen.
        STA (BACKL,X)    ;Store in background table.
      > LDA (MSHPL,X)    ;Get byte from mask shape.
      > EOR #$FF         ;Exclusive – OR with $FF.
      > AND (HIRESL),Y   ;AND it with screen.
      > STA (HIRESL),Y   ;Store it on screen to leave
                          hole.
        LDA (SSHPL,X)    ;Get byte from shape.
        ORA (HIRESL),Y   ;OR with screen.
        STA (HIRESL),Y   ;Store on screen.
        INC BACKL        ;Next byte in background
                          table.
        INC SSHPL        ;Next byte in shape.
      > INC MSHPL        ;Next byte in mask shape.
        INY              ;Next screen pos. in row.
        DEC SLNGH        ;Decrease width.
        BNE SDRAW2       ;Finished with row?
        INC TVERT        ;If so, INC. to next line.
        DEC DEPTH        ;Decrease depth.
        BNE SDRAW        ;Finished all rows?
        RTS              ;Yes, done!
```

The SSETUP subroutine must also be modified as follows:

```
SSETUP LDA#<SHIP
       STA SSHPL
     > LDA#<MASKSHP
     > STA MSHPL
       LDA#>SHIP
       STA SSHPH
     > LDA#>MASKSHP
     > STA MSHPH
       LDA#<BACKGRD
       STA BACKL
       LDA#>BACKGRD
       STA BACKH
       LDA#$08
       STA DEPTH
       LDA#$09
       STA HORIZ
       LDA#$03
       STA SLNGH
       STA TEMP
       RTS
```

Graphpak – just what the engineer and scientist ordered

MANY programs are available for the Apple which allow the business user to produce graphics quickly – for example bar charts, pie charts and line graphs – on the Apple screen and then on a graphics printer or, in a few cases, on a digital plotter for higher quality output.

They are often ideal for such users, and enable attractive, meaningful business graphics to be produced, modified and presented far quicker than by hand.

Some, like the Visitrend/Visiplot combination, and AIDA, (reviewed in *Windfall* in September 1983), also offer statistical analysis, or trend analysis, to help in analysing previous figures and predicting future trends.

Business users are not alone in requiring such programs, however.

In the scientific and engineering field, graphs are used probably more than any other method for presenting large quantities of experimental data in an easily appreciated form.

Statistical analysis is also commonly required, to analyse experimental data and derive suitable equations to model test results.

Many scientists and engineers have tried to use business graphics programs for such purposes. These attempts are rarely successful, owing to the different nature of the data and differing requirements for data analysis.

Briefly, these differences are: business data is almost always time-dependent – that is one variable is always time – whereas scientific and engineering data is rarely of this type.

Statistical analysis of business data usually involves very simple trend analysis (often linear), combined with sophisticated facilities for correcting or seasonally adjusting data.

For scientific or engineering purposes, correction of data is usually unimportant, but rather more sophisticated trend analysis is necessary.

These requirements are met remarkably successfully by the set of graph plotting and data analysis programs from Interactive Microware, distributed in the UK by Heyden Datasystems. They consist of Scientific Plotter, Curve Fitter and Vidichart.

This review concentrates on the first two because I have used them more, and also because they fit naturally together, performing complementary functions as already outlined.

First, let me deal with the drawbacks to the programs.

The graphics requirements of business, scientific and engineering users differ considerably. Here KEITH WILLIAMSON explains why this is so and examines Graphpak, a suite of three programs from Interactive Microware that go beyond the commonly available business graphics packages to cater for more specialised requirements. His report covers Scientific Plotter, Curve Fitter and Vidichart and considers their use with the Heyden flat-bed plotter.

Scientific Plotter and Curve Fitter are both written in Basic, have a rather idiosyncratic style and terminology, and can at times be very slow.

On the plus side, they certainly work, and can perform very sophisticated graph plotting and data analysis in the hands of an experienced user.

Scientific Plotter is a program for plotting two co-ordinate graphs in one, two or four sector format on the Apple screen, from where they may be dumped to a graphics printer.

A modified version of the program also allows the graphs to be drawn on a Heyden Digital Plotter for improved quality.

The general function of the program is as follows. First the graph axes are defined, then the data to be plotted is input, scaled and plotted. Further data may then be plotted on the same axes if required.

No menus are used by any of the IMI programs, each graph parameter is input on a separate line, and where alternatives are presented the choice is made by typing either the initial letter of the selection, or the complete word.

Each input line includes a default value, which is either an earlier input made by the user, or an entry in a Format file – more about this later.

This provision of defaults is a definite plus point, allowing similar graphs to be produced by altering only those parameters which are different.

The graph axes are defined by specifying their ranges (minimum and maximum values), and their positions on the screen.

This second part can be a little tricky if done directly in screen co-ordinates (I still have difficulty with the "upside-down" Y axis on the screen), but is made much easier by the ability to use the games paddles to move a flashing cursor on the graphics screen to define the position required.

This facility is not clearly defined in the manual, but it is well worth using.

Two and four-quadrant graphs are produced by specifying the X and Y axes without common corners. Either the whole screen, or part of it, may be used for each graph, so it is possible to have more than one graph at a time on the screen.

The axes may be graduated as required, with major and minor tick marks and labels at each major tick. A neat point is that the ticks need not correspond exactly with the start and end of the axis.

The labels may be printed in fixed decimal point notation, and they are automatically staggered or rotated to prevent overlap if this would otherwise occur.

The axes may then be framed, and grid dots drawn at each minor tick intersection.

The next stage is to feed in the required data, either from the keyboard (a fairly rudimentary editing facility is provided), from disc, or, for standard functions, by a Basic subroutine.

The data may then be scaled, offset and/or converted to logarithmic values before plotting.

Five different plotting symbols, each in four sizes, are provided, and most may be either left open or filled in with solid colour, thus giving more variety.

The symbols can be joined with straight lines, and error bars included.

Once the data has been plotted, up to five labels may be defined. These may be used as axis labels, in any of four perpendicular orientations, or as keys to identify individual data, or as graph titles.

The method for label definition is probably the most difficult aspect of Scientific Plotter to come to grips with at first, since it is necessary to specify the colour, the orientation and the position of each label using special embedded characters.

As an example, a label

reading "EXAMPLE GRAPH" to be printed horizontally at screen co-ordinates 100,50 in colour number 3 would be defined as:

@100,50#3&0EXAMPLE GRAPH

I am not ashamed to admit that this complication, along with the bug in the original program which caused a crash if a null label was defined (this has been corrected in the Mark II version) made my early attempts at label definition absolutely disastrous.

Once the technique has been learned, however, it becomes relatively straightforward.

This process completes the plotting of a simple graph. More complicated plots may be produced by overlaying extra data and/or labels without erasing the first graph.

The next stage is to save the data, the format and the picture on disc in a standard format which may then be read by other IMI programs. The complete graph may be stored as a Picture file, which is a simple binary screen image.

The most interesting and useful storage facility is the Format file, which contains all the parameters used to plot the graph. This file may then be read back into the program to set all the default entries, as described earlier.

Any graph produced may be dumped to a graphics printer. Since the program is written in Applesoft Basic and the relevant lines are well documented, it is quite simple to modify it for any common printer and interface.

A typical hard-copy graph, produced on an Epson FX80 printer, is shown as Figure I.

This is where the main drawback of the screen-oriented type of graph plotting program on the Apple becomes apparent, for the final product is certainly not of presentation quality.

This is not, of course, a fault of the program, but a limitation of the Apple graphics display. Really top class graphs can only be produced by a digital plotter, and now that several excellent machines are available for under £1,000, such a device would seem to be the ideal hard copy unit for serious graph plotting.

With this in mind, I am pleased to see that Heyden is also marketing a version of Scientific Plotter to drive its own six-pen digital plotter.

Since this is probably the first mention of the Heyden plotter in UK literature, I shall describe it briefly.

It is a fairly inexpensive (£980 for the six-pen version with parallel interface) A3-size flat bed plotter, made by Iwatsu in Japan.

I tried it briefly and found it reasonably quick and very convenient to program. It has very powerful commands for a plotter in this price range and the only fault I could find was the rather strange-looking character set.

All the usual accessories such as fibretip ballpoint and drafting pens, different grades of paper and OHP (overhead projection) film are available.

The modified program is supplied, as a separate modification which, when EXECed, alters the standard Scientific Plotter program to allow for output either to the screen or to the plotter. If the screen is used, the program runs exactly as normal.

However, if plotter output is selected, the screen is not used at all, and all output is sent to the plotter.

The program runs almost identically, the only main difference being the different orientation of the Y axis (0 at the bottom), and the order of magnitude or so difference in the co-ordinates used.

Figure II shows an example graph produced by the Heyden plotter, and the vastly improved quality is self-evident. The modified program contains some very impressive details, such as a reproduction of the "graphics cursor control" method used for defining co-ordinates on the screen.

This allows the plotter pen to be moved, under the control of the games paddles, to the required position on the plotter surface for any axis or label, with both coarse (covering the whole plotter surface) and fine (covering about +–25mm) control.

Any required co-ordinate can thus be defined to the maximum accuracy of the plotter, something which normally requires a digitisation facility which is standard only on higher-priced plotters.

Using this method, graphs can even be plotted exactly on the graduations of drawn graph paper, which is very impressive for demonstration purposes.

So far I have referred only to the Mark I version of Scientific Plotter, which has been available in this country for some

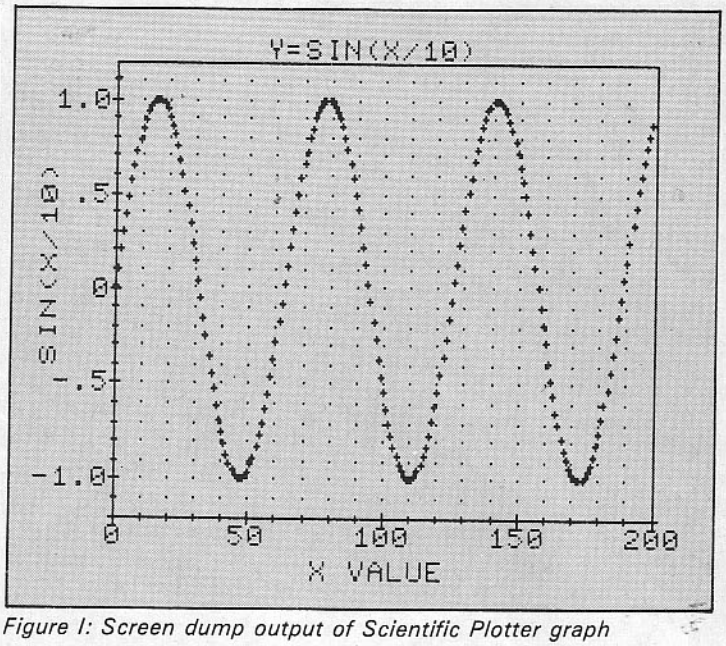


Figure I: Screen dump output of Scientific Plotter graph

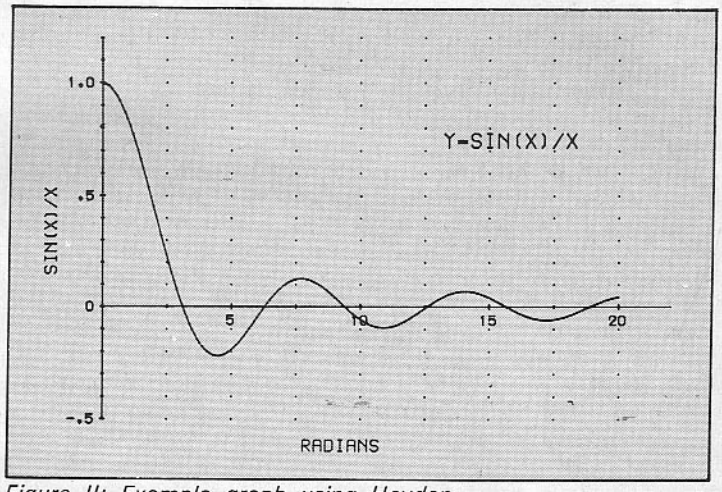


Figure II: Example graph using Heyden digital plotter (Mark I program)

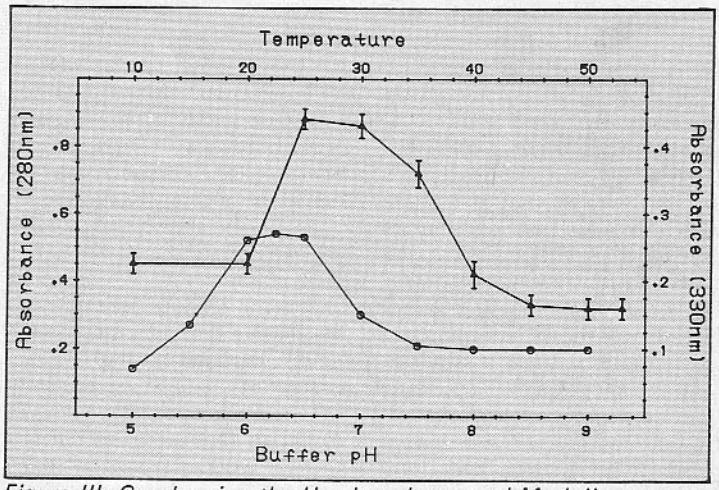


Figure III: Graph using the Heyden plotter and Mark II program

time. The main differences between this and the recently-released Mark II version are in the screen layout, which has been "prettified" with screen headings and subheadings, and the provision of lowercase letters in labels.

There is also now a Screen Editor which allows labels to be written directly onto the graphics screen (I prefer the old version, but this is probably due only to familiarity), and most of the Mark I bugs have been corrected.

I have also tried the Heyden plotter version of the Mark II program, which again is the same as the standard program except in the quality of the finished graph (Figure III). Similar IMI-produced modified programs are available to drive the Hewlett-Packard 7470A and Houston Instruments Hiplot plotters.

These are not quite as convenient or versatile as the Heyden plotter version, since they first require a Format file to be set up using screen display, and then simply reproduce this in one of four fixed sizes on the plotter.

Certain parameters must be altered or added before plotting (for instance, the graph origin and the colour of the axes), so the whole process seems a little long-winded. However, for owners of these plotters the modified programs must still be highly recommended.

I have used Curve Fitter extensively over 18 months, and have at times felt like giving it up in favour of something faster, like pencil and paper!

The root cause of this is the sorting routine, which must be the slowest ever invented (even slower than a routine which sorted by comparing every number with every other number!), taking *10 minutes* to sort 200 pairs of numbers.

It sometimes even sorts already sorted numbers, and still takes almost as long as for random data. The routine deserves some sort of prize for inefficient programming.

Now for the strong points of the program. The main one is that it works. It will perform linear and non-linear curve fitting (regression analysis) and interpolations on data covering almost any range, using

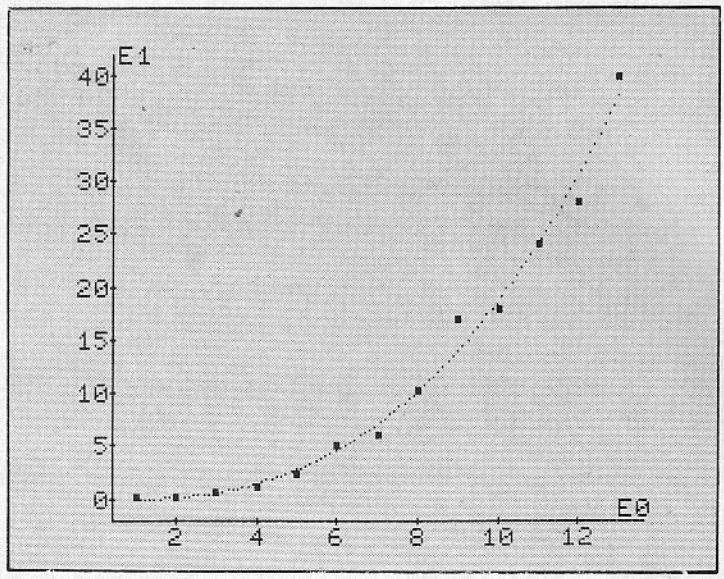


Figure IV: A quadratic equation fitted using Curve Fitter

exponential, geometric and polynomial methods for curve fitting and Stinemann, cubic spline and polynominal for interpolation.

Now I am by no means an expert statistician, but to me this seems a fairly standard set of statistical functions. However, I have been unable to find any other readily-available program running on anything less than a mainframe that has the same range of applications.

Despite my reservations about its speed therefore, I have used Curve Fitter regularly and shall continue to do so until something better appears.

The program is closely integrated with Scientific Plotter, to the extent that the same data files may be used, and fitted curves from Curve Fitter may be plotted using Scientific Plotter.

The layout of the program is similar, with Format files being used to save the parameters used, and default entries provided for all input requests.

The general procedure is that the data used is entered, and scaled and/or offset if required, and then optionally listed and plotted, using a simple auto-scaled layout for the graph. Smoothing and/or averaging may then be used to take out random variations.

The actual curve fit or interpolation is then performed, and known data may be input to determine the unknown parameter. Finally, the format, the picture, the data, the unknown data and the fitted curve may be stored on disc.

By combining several of these functions, it is possible to fit data to quite complex types of equation. For instance, I have used a linear analysis on logarithmic data, with a fixed offset applied after taking logs, to fit data to an equation of the form:

$$y=x^{-a}+b$$

The parameters returned by the program after a regression analysis are the coefficients of the best-fit equation, and also the coefficients of correlation and determination and the standard error.

The fitted or interpolated curve is plotted superimposed on the data points to help the operator to decide whether the equation is satisfactory. If not, the interpolation or curve fit may be repeated as often as required.

Figure IV shows a quadratic curve as fitted to a typical set of data points.

Vidichart is one of the most impressive graphics programs I have ever seen on the Apple. I have not used it extensively however, because I have not yet found an application for it in my field of work.

Basically, the program allows you to enter up to four separate sets of data into the Apple memory, and then to display any or all of these concurrently.

Each curve may be scaled, in both directions, and scrolled in four directions. Then you may perform numerical operations on individual sets of data (for example, add a constant, multiply by a constant, integrate or differentiate), or combine two sets in various ways to create a third set.

Other functions are provided which have particular application to chromatographic and spectrophotometric data, such as baseline definition and normalisation.

Vidichart uses the same type of standard disc file as the other two programs, so data may easily be transferred between the three.

Data may also be input from the keyboard or from an external device, such as an analogue-to-digital converter. Each set of data may contain up to 1,024 points.

I hope this brief description gives you some idea of the power of the program, since I shall not attempt to describe it in detail. The most impressive thing about Vidichart is the sheer speed of its graphics. My reservations about the slowness of the other IMI programs are certainly not relevant here.

My conclusions are based mainly on my use of Scientific Plotter and Curve Fitter. They are extremely useful graph plotting and statistical analysis tools for any scientist or engineer who habitually needs to produce accurate graphs and/or perform regression analyses or interpolations on experimental data.

Main faults are the slowness of some routines and the general lack of user-friendliness.

The graph-plotting routines are immensely flexible, if difficult to use at first. To produce a quality of output commensurate with the accuracy of the program, the digital plotter option (preferably the Heyden unit) is strongly recommended.

The curve-fitting and interpolation routines are again very flexible, if sometimes slow. Very complex equations can be fitted to data.

The programs are very cheap for what they do. They are not copy-protected, so backing up is no problem. In my experience, no other similar Apple software is available with comparable usefulness to scientists and engineers.

From a tragedy to a miracle

MIKE COWLEY reports on how an Apple is fighting for a better life for disabled people

Pictures: Daily Mail

PC Philip Olds takes his first steps – assisted by an Apple

IT was the evening of December 23, 1980 and PC Philip Olds – then aged 28 – was on routine traffic patrol in Hayes, Middlesex.

Here was a policeman truly happy with his lot. For he had fulfilled his one overriding ambition when he joined the force – to follow in the footsteps of both his father and grandfather.

As he observed the pre-Christmas revellers lurching past he was unaware that he was about to experience a nightmare which would haunt him for the rest of his life.

Then came the call which sent him racing towards an off-licence in Yeading Lane.

Two hooded men had entered the shop, held the staff at gunpoint and demanded the takings. But even after a shot had been fired into the ceiling, no one handed over the money. In desperation, the bandits tried in vain to break open the till.

Fleeing empty handed, they found themselves confronted by PC Olds armed only with a truncheon.

Taking deliberate aim, one of the raiders shot the young constable in the shoulder. The bullet tore into his body to puncture a lung, then smashed his spinal cord leaving him paralysed from the chest down.

As he lay on the ground, blood pumping from his wound, the last thing PC Olds recalls was being subjected to a vicious kick to the head.

Six months later he was to tell a jury at the Old Bailey: "I was only about five yards from the man who shot me. I never got any closer. He straightened his arm and deliberately pulled the trigger".

PC Olds was almost reduced to tears when the two accused men were cleared of attempted murder. However Stuart Blackstock, 21, was sentenced to life imprisonment on the lesser charge of wounding. His accomplice, 21-year-old Leslie Cook, was jailed for 17 years.

But long before the violent thugs were brought to justice, Philip Olds heard sentence passed on himself by some of the country's leading surgeons.

You will never walk again . . . You will be confined to a wheelchair for the rest of your life . . . There is nothing as yet known to medical science which can help you . . .

The bullet that had severed his spine meant that his brain could now only convey messages to his arms and the upper part of his body – the remainder rendered tragically useless.

Philip Olds, like most other formerly active young people who find themselves stricken in this way, raged inwardly and refused to accept the situation. He prayed for a miracle.

And when that miracle finally arrived, it was in the unlikely form of a big shabby bear of an American who uses Apple computers in his work – Dr Jerrold Petrofsky.

At the age of three he had built his first crystal set radio. By the time he was five he was selling them packed inside fountain pens at a dollar a time.

The boy genius was eventually to become the serious scientist who put Philip Olds back on his feet three years after the gunman's bullet left him confined to a wheelchair.

For the man who wears a burned out microchip as a tie clip is a world leader in his field at the age of only 35. He has successfully combined his twin disciplines – physiology and computers – to achieve a breakthrough in research to restore spinal injury victims to their former mobility.

Other scientists around the world had achieved remarkable

results by using electrical stimulation on paralysed muscles. For the normally limp legs of paralysed people had jerked back to life when carefully controlled impulses were applied.

But it was left to Dr Petrofsky to realise that much more could be achieved if computers were used to deliver the impulses at exactly the right moment and in the right place.

With a PhD in neuromuscular physiology and a degree in computer engineering, the American scientist was better equipped than most to appreciate the potential. And he chose an Apple as the principal tool in his research.

He began his experiments at St Louis University using cats. Later he transferred to Wright State University in Dayton, Ohio, where he is Director of the National Centre for Rehabilitation Engineering.

At Dayton, he succeeded in programming an Apple to allow paralysed people to pedal a specially adapted exercise bicycle by transmitting precision timed electronic signals to the wasted leg muscles.

It wasn't long before he developed a leg trainer – a computer controlled machine to allow subjects to lift weights with their legs.

After all, Dr Petrofsky was fully aware there was little point in stimulating legs to walk which had already been devastated by lack of use.

Only when the patient's legs had been "rebuilt" would it be possible for the professor to prove his basic theory – that an Apple can duplicate the basic electrical messages the brain sends to muscles to make them move limbs.

It was thanks to the generosity of the Daily Mail that Philips Olds finally came face to face with Dr Jerrold Petrofsky. And it took some hard talking by the paper's editor to pursuade him to accept the police hero as one of his patients.

Dr Petrofsky already had a waiting list of 5,000 Americans pleading to be taken into his laboratory, but when the full story of PC Olds' courage was told to him he finally agreed.

Far from receiving star treatment when he arrived in the States, Philip Olds found himself playing very much a back seat role. That was because Dr Petrofsky and his lab team were working to meet a deadline for one of their guinea pigs, Nan Davis.

A 23-year-old student teacher, her spine had been smashed in a road accident and she had been in a wheelchair since the age of 20.

Like Philip Olds she was a former athlete and a person of profound physical and physiological courage. Her ambition was to stand up and walk at her own graduation – a dream that Petrofsky's team were working round the clock to help her achieve.

Nan had already walked in the laboratory using the Apple. But Petrofsky had come up with a slimmed down version of the computer – small enough to fit in her handbag – to solve the problem of the graduation ceremony.

With this bag computer – its wires leading to electrodes on her legs – she donned her cap and gown and walked to receive her diploma. It was a major breakthrough which would prove vital for the future wellbeing of Philip Olds.

That walk brought major recognition to Petrofsky's work for paraplegics.

And more important, it was to result in the US government agreeing to fund millions of

dollars for the project.

Now the Dayton team turned its attention to Philips Olds. The day he walked again is best described by Andrew McEwean, Daily Mail feature writer who accompanied him to Dayton.

"With a flick of switches, Dr Petrofsky powered up the computer. The secret lay in the floppy disc. Dr Petrofsky had spent thousands of hours creating it and it now ran to 3,391 lines", he wrote.

"A red light flashed 'system active' and green lines raced across the two oscilloscopes – 36 computer channels were ready to pump commands into Philip's muscles. Dr Petrofsky pressed a small chrome switch high up in a mass of wiring.

"Suddenly Philip was up on his legs – for the first time since a gunman cut him down in a darkened street three years before.

"His face was set in intense concentration as he clutched handrails to steady himself. The computer is programmed to provide balance, but it takes time to learn to trust it.

"The event was right out of Philip's control. Jerkily, awkwardly, his legs began to move with the uncertainty of a baby's first steps, responding not to Philip's will but to the computer's commands.

"It was going to need practice, a little more strength in his muscles and a small program change to make it smoother. But the miracle was happening".

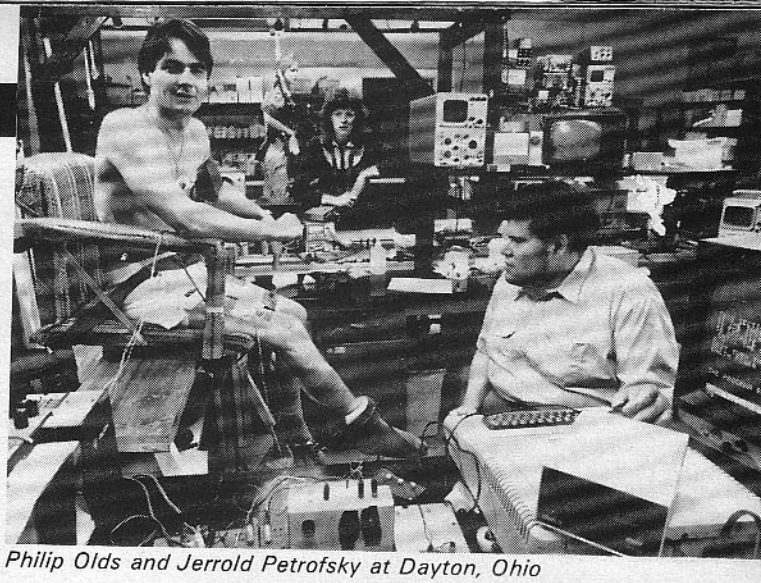
Only six subjects had walked before and Philip had reached that stage in the shortest possible time. In all, he walked 15 times on five days during a two week period, taking up to five steps each time.

Philip returned to the police force in London early this year and has become an instructor training cadets. But the intention is that he should eventually return to America to test an implanted version of the computerised stimulation system.

This will involve surgically fitting a microchip computer the size of a pacemaker, and flexible wires running under the skin to link it to various muscles.

Dr Petrofsky is now in the final design stages of the system and hopes to try it out within months. It will fit into a round, flat titanium box and will use a battery which is expected to last nine years.

If it works as hoped, PC Olds should be able to walk when he wishes. And more than half a million people in Britain and America confined to wheelchairs with spinal cord damage will be given fresh hope – thanks to Dr Petrofsky and his Apple computer.



Philip Olds and Jerrold Petrofsky at Dayton, Ohio

Dangers of two for the price of one . . .

HAVING recently resorted to cutting a "notch" in the other side of my single sided discs and thus using the "back" so to speak, I was informed by a colleague that this was undesirable on the grounds that when I ran the 'reverse' side I was in effect "going against the grain" of the protective/cleaning envelope and thus possibly damaging the disc.

Could you please comment as to whether or not this is the case. — **A.M. Hope, Eyemouth, Berwickshire.**

★ ★ ★

A FRIEND of mine has just cut an extra notch in a single sided 5¼in disc, and used the flip side for extra storage.

The process seems to work and he has not found any problem. Have you any views on this, or is it a common trick that I have not known of before. — **C.C. Wood, Chard, Somerset.**

● Well, yes, it's pretty common — either that or to bypass the write-protect switch with another in parallel.

There's been a lot written about possible damage to the head — dirt collecting because disc is rotating the other way — but many people do it, especially for infrequently used back-up copies.

Max Parrott

Darts off target

COULD I request some help from Max Parrott regarding the extension to his Apple Darts program by Dave Eckersall in the February 1984 issue of Apple User.

I have been using the original program successfully for several months but after typing in additional lines for the darts update, the program goes through its normal routine of drawing the board etc except that the count now goes up to Z, then I get an "out of data" error message in line 2220.

Upon checking this line all is OK. However when the program is then listed the last five lines of data are missing — the program ends part way through line 2370 at 13.

If I then reload the program from disc the full listing is obtained. It appears that some of the later lines are being overwritten in some way — could you please suggest what is happening?

Finally could you also tell me if the Planet Plotting program from issue No 11, May 1982, has any bugs in it as I cannot get the Slide Show program to work.

The program requests the required information but then when it appears to be loading information from disc for the different planets it just goes on and on (the disc drive that is) and never stops.

Sorry to be so late with these pleas for help but I have just recently bought all the back issues and am working my way through them as a complete novice.

Thanks for a great magazine but I much preferred the title of Windfall — if seemed so much more friendly. — **David Leach, Middlesbrough, Cleveland.**

● You have run into the "hi-res page conflict" with Darts. Briefly; your program is stretching from the usual start of program location ($800 if you're technically minded) past the start of hi-res graphics page 1 ($2000).

So as soon as an HGR command is issued the end of the program is wiped out by a load of 0's.

The easiest answer is probably that provided by Peter Gorry in the March 1984 issue of Apple User (page 32).

Quite why your listing is longer than Dave Eckershall's I don't know, but it doesn't really matter unless you re-created the REMs by using lots of spaces — which is taking up a lot of memory — the original used a commercial editing program to insert line feeds into the REMs.

Regarding Planet Plotting, the drive is kept running by the PEEK (49385). So remove this line and check that there are no misloadings — that is, to the wrong address so that they cannot be seen.

Check that the machine code is being loaded. If you still cannot get it to work send us a copy of your program on disc and we'll try to find the error.

Max Parrott

Slowly in circles

WE encountered some problems with Max Parrott's Darts program in the March 1983 edition of Windfall.

□ *The circle-drawing routine did not work properly, and we replaced it with a simpler, although slower, routine as follows:*

```
10  GOTO 990
20  FOR I = 0 TO 2 * PI STEP
    .01
30  HPLOT X + R *  COS (I),Y -
    4 *  SIN (I)
40  NEXT
50  RETURN
```

□ *Line 150 did not give the correct dimensions for filling the alternate sectors, and this was changed to read:*

```
150 R=54:R1=50:R2=81:R0=
    86:GOSUB....
```

□ *When playing against the Apple, a "NEXT WITHOUT FOR" error sometimes occurred in Line 830.*

We suspect the fault lies in the subroutine at line 590 for the Apple's move. This is the only subroutine not used when there are two independent players — line 650 may cause a jump out of a loop. We would welcome confirmation on this.

As suggested by Max Parrott, we extracted the board-drawing routines and created a separate program to draw the board and BSAVE it as BOARD. It is then BLOADed during the game program, as follows:

```
1100 D$ =  CHR$ (4)
1101  PRINT D$;"BLOAD BOARD"
1102  GOSUB 120
```

All board-drawing routines were then deleted from the game program, that is, lines 19-110, 129-160, 219-230 and a new line 215 RETURN was inserted.

November's Windfall contained two useful programs on verifying alphabetic input and dates. In Alan Dubost's routine, part of line 17250 has not been printed.

I have used P$ = P$ + DX$(S): NEXT S: GOTO 17340.

I have also amended line 17350 to read . . . THEN P$ = P$ + " " otherwise the length of P$ would never reach 20.

In C. Geraghty's date verifying program, I found it necessary to make three amendments:

Line 170 should read IF VAL(MID$(DA$,4,2)) = 2, (not 0).

Line 195:

```
195 IF OD<11 THEN 180
```

was inserted and the end of line 190 amended to read:

```
....THEN RESTORE: GOTO 180
```

Otherwise the program would not check for all four months containing only 30 days.

If the month is February and the date is 29, a test should be made for a leap year, and line 175 was inserted as follows:

```
175 IF VAL(MID$(DA$,4,2))=2
    AND VAL(MID$(DA$,1,2))=29
    AND VAL(MID$(DA$,7,2))/4
    <> INT(VAL(MID$(DA$,7,2))
    /4) THEN 180
```

Thank you for a very interesting magazine and all the useful

programs and tips. – **H. Anderson and A. Rowsome, Dublin, Ireland.**

P.S. An alphebetical index of all articles, programs, tips, etc. since the first issue of Windfall would be very useful.

● I'm puzzled by your statement that the original circle drawing routine didn't work properly. As far as I can see on my monitor the two are almost identical in appearance. By the way, I assume your line 30 should be Y-R*SIN(I).

I do agree with you about line 150; your values do give a better appearance.

I'm also puzzled by the 'NEXT WITHOUT FOR' error. Line 650 does jump out of the loop but this shouldn't cause that error message – OUT OF MEMORY is more likely.

I have never seen Darts cause either error. The NEXT WITHOUT FOR suggests that some rarely accessed line is jumping into the loop at 760-830 but I can't find one – has anyone else spotted anything?

Max Parrott

Mac pioneers

CAN we please have technical comment from our friends at Apple UK who soldiered so well through Apple 84 recently, on the following, extracted from Byte Letters of May/84:

"The Macintosh previewed in February Byte looks like a fine computer with its powerful 68000 CPU and sophisticated system software.

"However, I think the Macintosh development team committed a fundamental design error when, having exhausted ROM space, they placed their floating-point software in RAM. Software running out of RAM executes approx 25 per cent more slowly on the Mac than does software running out of ROM.

"It would have been more appropriate for them to have moved some of their user-interface or I/O software to RAM instead.

"Software that deals with mice keyboards, etc can run more slowly than 6MHz before there is any perceptible loss in speed of RUN.

"I hope that Apple moves to correct this error before the company floods the market with its current system". – Robert Lurie, NJ.

Is this really true? If so, I think it behoves Apple UK/USA to issue a public explanation immediately.

Again, from the May issue of Byte, their senior technical editor Gregg Williams:

"Initial reaction to Mac has been strongly, but not overpoweringly, favourable. However a few traditional computer users see the mouse, the windows, and the desktop metaphor as silly, useless frills, and others very concerned at the lack of color or graphics(?).

"Most, but not all, have expressed concern at the relatively small 128k memory, particularly the user-section left with most programs. The other strong feeling is over the lack of double disk drive from the start".

Again, will Apple please comment urgently – as a committed purchaser I am very concerned at the comment in that same Byte issue . . . "It is widely believed that Apple will offer a 512k Mac at the end of this year".

Are we ardent supporters of Apple to be treated as guinea pigs and suffer as pioneers with Mac, losing out over the original price – printer now extra – and specification? – **P. Knight, Sevenoaks, Kent.**

● Since languages come on disc and not in memory, Basic and Pascal all run in RAM as far as we know and not in ROM.

We believe that it's slower because of extensive error checking but in the absence of a machine at the moment we don't actually know.

The 512k version is the Big Mac and it's possible that existing customers will be offered an upgrade.

There is also a rumour that Big Mac might be IBM-compatible . . .

A rose by any other . . .

CONGRATULATIONS on the new name – it sounds more like a computer magazine rather than a gardening magazine.

I really enjoyed reading the interview with Steven Jobs. It's great to hear about predictions and ideas for the future, especially from the head of Apple himself.

Good stuff, Apple User!

I am an Australian Apple user who has recently moved to London. I just bought a new Apple dot matrix printer and a Grappler and interface.

Does anyone know how to list Pascal programs to the printer like one can with Basic?

I use an Apple IIe and Pascal 1.1 – any help on this matter would be much appreciated.

Also the 3-D surface pictures in the May issue: What are the range, magnification, etc. values for the sample pictures published?

I have been playing around with many values and changed lines 140 and 150 as mentioned, but to no avail. Can anyone help? – **Leon Seltsikas, London.**

● To list text files, enter the filer and press T for transfer. At the first prompt type the full description of the file, for example:

 # 5:MINE.TEXT

At the second prompt type PRINTER: (the appropriate device number could also be used). In the same way you can list to the CONSOLE: You can, if you want, type ,PRINTER: or an addendum to the first prompt's reply.

Peter Gorry says that the sine function has range 10, magnification 2 and grid size 10. The square function has values of 5, 0.1 and 10 respectively.

Max Parrott

No sleight of hand

I WOULD like to ask you some questions concerning the May 1984 issue of Apple User. Would you be so kind to excuse my "continental" English writing.

I was very interested in the article of John Fothergill on page 24: "Using DOS Commands from within a Basic Program".

There is one problem which troubles me. On a few occasions, the author directly leaves the ONERR GOTO routine without RESUME-ing.

Referring to the Applesoft Basic Programming Reference Manual Part I, Page 71, it is recommended never to leave such a routine with a GOTO without CALL -3288, which cleans the stack.

Is it possible that the author has used a trick to avoid this problem, which I don't actually see?

Another problem which troubles me, and which is not treated by the author, is how to INITialise a disc from within a Basic program, using DOS commands.

I know that it is dissuaded in the manuals, but some programs like Applewriter IIe and Information Master make this option available. So, I would like to use it in my own programs too, because of its use. – **Mark Braem, Deurne, Belgium.**

● Regarding the leaving of the error-handling routine without a CALL -3288 – no, the author hasn't used a trick. It perhaps would be best to CALL -3288 before the GOTO.

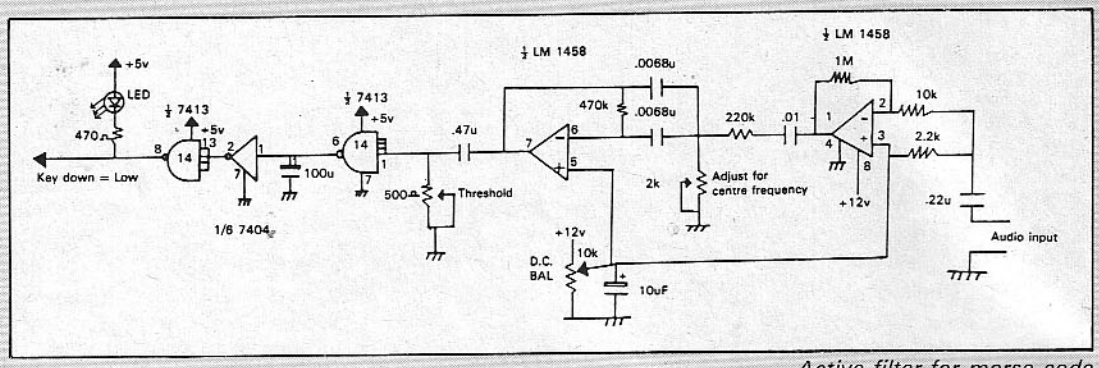
To initialise a disc from within a program is easy if you want to make the program the HELLO on the new disc:

```
10 PRINT CHR$(4)''INIT HELLO''
```

However, to initialise a disc without DOS as in Applewriter IIe is more complicated – Windfall Volume 2, August 1982, Page 30 has the routine to do it.

Max Parrott

Coping with morse



Active filter for morse code

IAN Bucklar's letter (Apple User, June 1984) prompted me to let you know of my experiences with using my Apple II for sending and receiving morse code and also RTTY (radio teletype).

The Apple has a built-in interface which is ideal for this type of application – the game I/O socket at the rear-right of the motherboard.

This 16 pin DIL socket has four analog inputs, three TTL inputs and four TTL outputs (TTL refers to a standard for logic circuitry, and uses voltage levels of 0V and 5V to represent on/off or hi/lo or the binary 1/0 states). It is the TTL inputs and outputs which are of use here.

Basically, all that is required to decode morse code, apart from software, is a circuit which will accept the audio output from the receiver and generate a 0V level when no tone is present and a 5V level when a tone is received (or vice-versa; this will depend on the software).

These devices are usually based on one of two principles – active filters and tone-decoder chips. Both types of device can be built for about £5. I enclose a rough circuit diagram for a device of the active filter type.

Software, which is fairly readily available, works by repeatedly checking the status of the input and decoding the information into text. The better software will make allowances for variations in the speed of the code by more rapid sampling. Either way, the program will have to be in machine code in order to work quickly enough.

There are many ways of using the Apple for amateur radio – I mentioned RTTY at the beginning (this is telex sent by radio) – slow-scan TV and satellite tracking are two possibilities.

The best article on the subject that I have seen was in the December 1983 edition of Radio Communication, the journal of the Radio Society of Great Britain.

In that article the example screens were produced on an Apple, using software developed by an American, Chris Galfo. – **Simon Holt, Burnley, Lancs. (G3ZLB).**

User group for hams?

I READ the letter about decoding morse with great interest. I am a radio amateur (8 TXO) and am interested in using my Apple to decode morse.

I have an almost complete design for the hardware required – an offshoot of a design for the Spectrum. The hardware consists of a tone encoder and decoder which are connected to the Apple via the games I/O port.

The circuit has been designed to allow complete flexibility to the user, so much so that the board can be used for radio teletype and as a modem.

I would be interested to know if there are any Apple users in the amateur radio fraternity who would be interested in forming some kind of user group. – **Zubin N.E. Manekshaw, 488 Strathmartine Road, Downfield, Dundee.**

Encoder poser

REGARDING your Feedback item on decoding morse, I purchased the Amateur Radio Communications package on disc when I was in the USA.

The system works very well on RTTY and Morse. I have built the active filter and it worked first time with my FRG7 receiver.

Can someone please help me? I have an Apple II which was imported into the UK. Everything works except the PAL encoder card. All the 50/60 solder points have been changed as well as the crystal.

The card is OK as it works on the Apple I have at work which has – wait for it – two Ramex 128 cards, two printer cards, one Vision 80 card, one accelerator card plus disc drive card.

Re your communications article, more on this subject please as this is the future. – **B. Sweetins, Harlow, Essex.**

A plea for help

I AM a severely disabled person, and rely on the Apple for all the work I do, that includes the running of a Borough Association, that is being the chair and treasurer of our Community Transport, which includes a Dial-a-Ride service.

Two years ago I started the Open University with a view to studying music. Up to three years ago I was able to be involved in music to a large extent, being part of a recorder ensemble and playing the organ in the church.

But my disability has worsened, so I am no longer able to perform music.

I passed the Open University foundation course, of course with the help of the Apple (how else!) and registered for the music course. I thought my problems were solved, regarding the actual playing of music, but when the units arrived, lo and behold I had to write music myself, even in the exam. This is not possible for me at all.

Since then we have been trying to get a program which will enable me to do the course and to write the music, but nothing has been found as yet.

Perhaps you are able to help me through the magazine, as one of the readers might have just what I need.

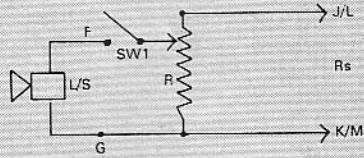
Music has been a large part of my life, and it very hard indeed not to be actively involved in it any m

I would like y much, though, to do e Open University course year and in consequent ye f anyone can help with t problem, please do so. I looking forward to hearin m you. – **Sister Elly Mari tekoek, London.**

Power p t

I THINK, with knowing Apple's inside ( ) in any detail, but basic e ronics tell me that Dr P. Rickwood (July 1984 Apple User) has not got the best circuit connections.

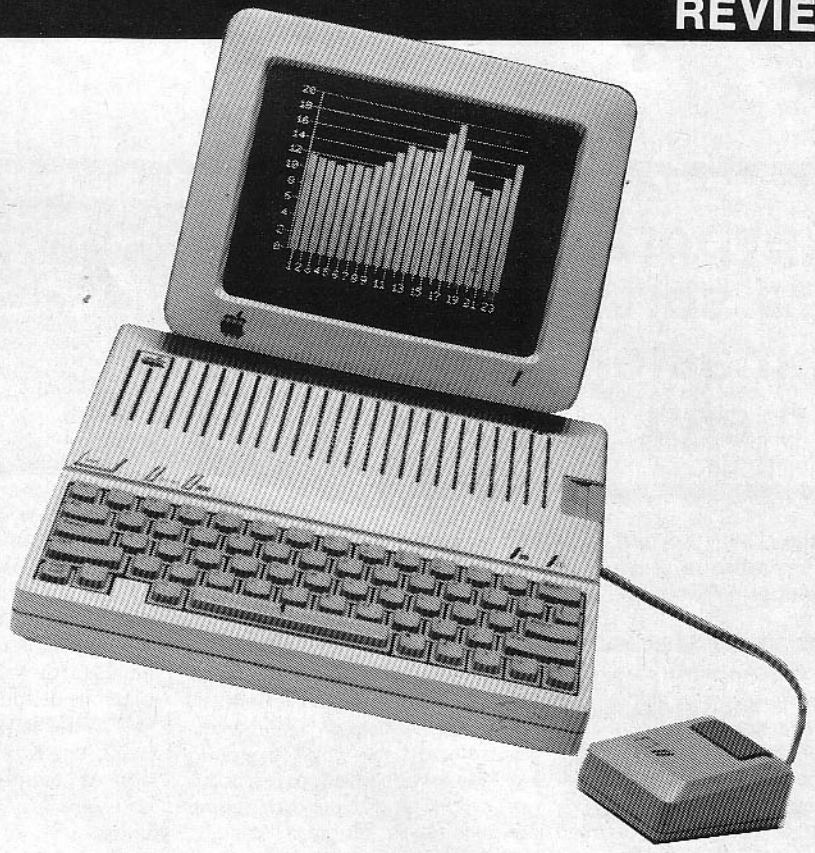
This circuit is merely dissipating power across series R (A-B) and gives very poor matching. The rough schematic below is more usual where R = 10 x source resistance Rs:



R can be a linear law potentiometer if preferred and switching off the loudspeaker will not disturb the output board unduly. – **P. Knight, Sevenoaks, Kent.**

PS. SW1 is conveniently linked to slider by using a switched control.

The mouse meets WP, courtesy enhanced Format 80



FORMAT-80 Enhanced from Elite Software is one of the first independent Apple word processing packages available with the ProDOS operating system and mouse facility.

As an existing Format-80 user, I examined this new version to see for myself the differences between it and the DOS version.

As ProDOS is used, the name appears as /Format-80. This version needs 128k of RAM due to the amount of room taken up by the operating system, and initially is only available for the Apple IIe and IIc.

Booting the program, the main menu appears in the usual way but with an extra option: V for volume, part of the ProDOS system. This option shows the slot and drive numbers plus what is available in each.

If a volume is empty, it appears as "No volume on line".

With the DOS version of Format-80, letter 'I' produces the initialise option. In the ProDOS version the message Catalog is shown plus the name of the program.

Then the sub-directory is created, which will be a work area using a hard disc system.

Under ProDOS there is no 'I' for initialise – you'll need the Apple System Master to prepare data discs.

When C for catalog is pressed, instead of drive one appearing as the message with the DOS version, the name and number headings appear with sub-directories.

Press B for booting and instead of the message "Insert program disc into drive one", it's enter prefix, then the application and pathname. This is pure ProDOS operation.

When C for catalog is pressed in mailing list mode, the other directories available will be shown.

Users of hard disc systems have an advantage with the mailing list option. Instead of being limited to just over 500 names, with a hard disc system the total is now around 40,000 – almost eight times more than the DOS version of Format-80.

Several different mailing lists can be used all within the 40,000 available entries. But with this vastly increased number of names and addresses, the retrieval time gets longer in proportion to the number of names on the list.

Within the mailing list option, the 'I' for initialisation creates a new mailing list, or if an existing name is available, points you to it.

By NEVILLE ASH

ProDOS uses the pathname system to denote sections of a hard disc and the new Format-80 allows up to 64 names per pathname.

Apart from these changes, Elite Software has also added the feature of using the up and down arrows of the IIe and IIc for some of the formatting commands.

Going through them, I found the following operations now have this facility:

Centre, **T**idy and compress, **Y** following, **A**lign number, **S** start of, **D**elete, **J**ustify, **Z** end of, **V** move, and **B**lank out.

When you press P for paragraph in the DOS version, the message "two lines one column" appears. With ProDOS the message is "two lines and five columns".

With the DOS version there was a trill at the end of the line or the end of the page, but with the ProDOS version this trill has changed into a buzz.

So, the advantages of the ProDOS version are:

- Users of hard disc systems can now have mailing lists eight times as long, and this can include a number of different mailing lists.
- Lots of the formatting commands now operate using the up and down arrows.
- Works with the new Apple operation system supplied as standard.
- Probably the first Apple word processor using a mouse and ProDOS.
- Both operating systems, DOS and ProDOS, supplied as part of the package.
- Still the same price of £129.

The disadvantages are:

- Needs 128k.
- ProDOS version initially available only for the Apple IIe and IIc.
- You need the ProDOS System Master to initialise data discs.

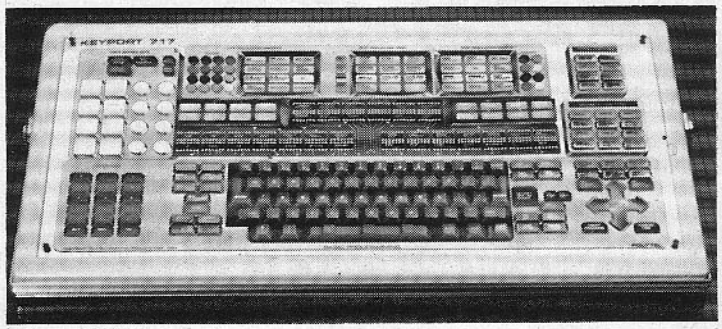
To summarise, then, an existing popular word processing package, page-based with single key commands, has been brought right up-to-date.

As Apple IIs are now being supplied with ProDOS as standard, new owners will already have this operating system.

If you want a word processor to make your Apple work like an electronic typewriter, this version of Format-80 fits the bill – and it costs the same as the old one.

Supplier: Elite Software.
09328-67839
Price: £129

Keyport – a sort of master menu



Keyport 717

FROM the USA is Keyport 717, a flat membrane keyboard which plugs into the Apple II, II+ or IIe.

It is claimed to make life easier for Apple users and programmers by making one key do the work of several.

Keyport is about one and a half times the size of a normal keyboard and has 717 user-programmable keys. It can be used for any application program.

A typical application uses 150 to 300 keys, and there is a different plastic overlay for each application. Unused keys do not appear on the overlay.

The function of each key is shown on the overlay by printed words, pictures or symbols.

Each command has its own key, so the user does not have to learn a language, or respond to menus.

In effect, say the makers, the Keyport is a permanent master menu which shows all the available commands and functions in the application.

This, they claim, makes programming easy, because there are no menus to set up or command syntax to develop. Applications can thus be programmed faster while using less memory.

For non-programmers, there are standard applications – keyboards for Visicalc, Basic and a couple of children's educational games.

The relevant overlay can be placed on the Keyport and the interface disc booted to tell the program what to do when a key is pressed.

Keyport has no moving parts or chips, and uses no power supply. It is plugged straight into the games socket.

● *European distributor: RAM Data, Kongensgade 71.3 TU, 5000 Odense C, Denmark. Tel: 010 45 9 179672.*

Share a Winchester

AN ICE product that allows up to eight personal computers to share a Microcube Winchester hard disc sub-system is now available from Rapid Terminals.

ICE PC Link can be used with Apple II or III machines or a combination of both, and the system is being extended to include other computers.

Each of the connected computers will access a different part of the hard disc which can have up to 84 mbytes of formatted capacity.

If Read-Write access to shared information is required ICE software "semaphores" can be provided to regulate that access for operating systems capable of multi-user access.

The semaphores are already being used by leading software houses to develop multi-user software for ICE mass storage units.

Each PC Link is supplied with a manual and host adaptor packs

The eight-user PC Link for Apple machines costs £1,045 (£1,075 for other PCs), and the host adaptor packs are £150 each.

● *Rapid Terminals, Rapid House, Denmark Street, High Wycombe, Bucks HP11 2ER. Tel: 0494 450111.*

Factory check up

A hand-held data collection terminal has made time and cost savings in routine checks on industrial dust extraction equipment.

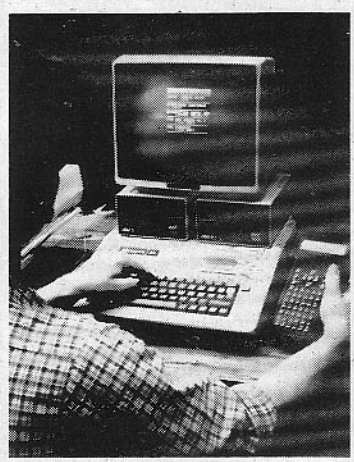
The terminal, from GR Electronics of Newport, Gwent, is being used with a computerised system, based on an Apple II, developed by the Steel Castings Research and Trade Association (SCRATA).

The SCRATA 'Ventdata' system has been developed as a quick way of logging air flow or pressure measurements at various points in an exhaust ventilation system, comparing them with previous measurements and even diagnosing faults.

It can hold performance histories of up to 150 separate items of plant over a five year period.

The GR Electronics type 42B terminal has 8k of user RAM and works off an internal battery giving 40 hours operation from one charge and protecting data in memory for several months.

At the start of a data collection run, it is plugged into the micro and loaded with prompts for all the plants under review – many of them several miles away.



Data from site measurements are loaded into the Apple for analysis and diagnosis of faults

When the measurement schedule is complete the operator goes back to the micro and plugs in the terminal. Information retrieved is sorted and filed by the micro, and any major drop in performance flagged for attention.

Use of the terminal allows data to be collected by unskilled labour. It also removes the need for manual notes to be transcribed and entered via the keyboard.

● *GR Electronics, Fairoak House, Church Road, Newport, Gwent. Tel: 0633 214147.*

Personalised mail shots on tap

MAILMERGER is a software package for Apple users who want to carry out personalised mail shots.

It brings together two useful functions of the Appleworks package – the word processor section makes the creation of form letters easy and the database caters for files of names and addresses.

Now Mailmerger for Appleworks combines the two functions to print out any number of customised form letters by merging information from a database file with a document created by the word processor.

By using identifiers when producing letters, names and addresses can be changed – or even parts of sentences.

The program tries to ensure that the user does not make mistakes such as confusing file names. It also checks that there are not more identifiers used in the document than there are data fields in the data base.

● *Leicester Computer Centre, 9 Jarrom Street, Leicester. Tel: 0533 556268.*

Drawing attention

DESIGNED to aid production, storage and amendment of scale drawings and other graphic material, the Robo 1500 technical drafting system is "the fastest, easiest and most advanced" package yet claim Robocom, the London computer graphics company.

The Robo 1500 – aimed specifically at architects, builders, civil and mechanical engineers – is the highest level of drawing software produced by Robocom.

Using the Robocom Bitstik controller, the Robo 1500 system is available as an add-on for the Apple II and as a compatible upgrade without price penalty for existing users.

Robocom have nearly 3,000 installed units worldwide.

Robo 1500 offers a broad spread of drawing office requirements, from numerical data entry to dimensioning and improved editing and filing.

Users who need high resolution numerical data entry can achieve time savings "of the order of several hundred per cent", claim Robocom.

Functions are displayed on screen as menus or palettes and can be selected with the Bitstik – which controls a series of "rubber banded" cursors – simply by pointing and pressing a button.

Full zoom and pan facilities allow the user to scan the design in detail or zoom out for a global view.

Effectively unlimited drawing storage is claimed for the Robo 1500 library system based on standard 5¼in, 128k floppy discs, representing up to twelve A1 drawings or 192 drawing details.

Robocom claim the use of graphic index pages, each with 64 boxes of filed information, gives the user faster and friendlier access to filed drawings than traditional lists of cryptic file names.

● *Robocom, Clifton House, Clifton Terrace, London N4 3TB. Tel: 01-263 8585.*

Compact drive

PETE and Pam is now marketing a compact disc drive for Apple II, II+, IIe and IIc micros.

Made in Japan by Chinon, the unit has a direct drive JVC motor. Its metal casing matches the Apple.

The P&P drive is considerably quieter than Apple Disk II drives, and it is about half the size.

Key specifications include 163k, 5536 BPI recording density, 48 TPI track density and a maximum of 40 tracks, and an average access time of 100msec.

A single sided, double density drive with 512 bytes per sector, the P&P disc drive sells for around £129.

● *P&P Micro Distributors, Todd Hall Road, Carrs Industrial Estate, Haslingden, Rossendale, Lancs. BB4 5HU. Tel: 0706 217744.*

Robo 1500 system

Advanced Practicalc

AN advanced version of Practicalc is now available for Apple.

Practicalc II is a fast, versatile, easy to learn integrated program for Apple II+ and IIe with 48k memory.

Based on a comprehensive spreadsheet, it combines word processing and data base functions and includes a number of unusual features.

Practicalc II retails at £69.95.

● *Practicorp, Goddard Road, Whitehouse Industrial Estate, Ipswich, Suffolk, IP1 5NP. Tel: 0473 462721.*

Copy protection

DOUBLE-Gold has introduced Lock-It-Up disc copy-protection systems for the Apple II and Macintosh. The menu-driven systems allow the implementation of several levels of security said to render standard discs virtually uncopyable by even the most sophisticated of pirating methods, including the use of advanced nibble copy programs.

The protection is based upon an encrypted signature track which is written to a random location on the disc. Protection code which verifies the accuracy and existence of this signature track is automatically encrypted into the programs to be protected.

Lock-It-Up also implements special modifications to the operating system which increase the level of data security. Protected files may be backed-up by the user, although they will only operate if they reside on the original disc.

Since the protection is not based upon a physical "fingerprint", complete write protection of the information is possible. The protection will not interfere with the normal operation of the protected software.

The system includes a software duplication method that allows protected discs to be rapidly duplicated for commercial production purposes.

Licensing of Double-Gold's protection systems begins at $495.

● *Double-Gold Software, 3900 Moorpark Avenue, Suite 29, San Jose, CA 95117. Tel: 0101 408 554 9133.*

Equation tackler

INTERACTIVE Microware, introduces Varicalc, a program for the Apple II that interactively solves science, engineering and business equations.

It can simulate complex physical, chemical or mathematical processes as well as accept real-time input directly into a predefined model.

A novel feature is the use of "variators" to change variables interactively. They can be two game paddles or a joystick, the left or right arrow keys, or an automated loop variator with selectable range and step size.

The analog voltage from Interactive Microware's Adalab data acquisition interface card can also be used as variator, and any variable can be used to control Adalab's analog voltage output. This permits feedback control of real time processes according to complicated control equations.

Varicalc will solve for any one of 19 variables on the right or left side of a formula without rearranging the formula. Results can be plotted on the screen or you can print hard copies of the graphics screen or any text screen. Up to 255 equations may be stored on disc for quick recall.

The Varicalc software costs $100. The manual is available separately for $5.

● *Interactive Microware, P.O. Box 139, State College, PA 16804-0139. Tel: 0101 814 238 8294.*

```
590  GOSUB 1900: GOTO 600
600  ON Z GOTO
   510,520,530,540,550,560,570,
   580,590,500
999  REM  TITLE
1000  GOSUB 5000:V = 1:H = 1:
   GOSUB 5200: GOSUB 4900
1010 V = 2:H = 9: GOSUB 5200:
   PRINT "MINI-PILOT (GMM)"
1020 V = 3:H = 1: GOSUB 5200:
   GOSUB 4900
1030  FOR I = 1 TO 9:V = 4 +
   I:H = 10: GOSUB 5200: PRINT
   I
1040 H = 14: GOSUB 5200: PRINT
   M$(I): NEXT I
1050 V = 23:H = 1: GOSUB 5200:
   PRINT "PRESS NUMBER OF ITEM
   REQUIRED ";
1060  GOSUB 6000: IF G < 49 OR
   G > 57 THEN 1060
1070 Z =  VAL (A$): RETURN
1100  REM NEW
1105 LT = LL:Z = 1:V = 1:H =
   1: IF A$ = "EDIT" THEN V =
   3: GOTO 1125
1110 CP$ = "": GOSUB 5000:C =
   0: GOSUB 5200: PRINT
   "PROGRAM NAME? ";:LL = LF:
   GOSUB 6100
1115 CP$ = A$: IF CP$ = ""
   THEN 1110
1120 LL = LT: GOSUB 4500
1125 V = V + 1: FOR A = C TO
   MX: GOSUB 4800: IF A = MX
   THEN  GOSUB 4300
1130  PRINT A;"?";: GOSUB 6100
1135  IF A$ = "EDIT" THEN Z =
   3:A = MX: GOTO 1175
1140  IF A$ = "DONE" THEN Z =
   10:A = MX: GOTO 1175
1145  IF  LEFT$ (A$,1) = "*"
   OR  LEFT$ (A$,1) = "!" THEN
   P$(0,A) = "":P$(1,A) = A$:
   GOTO 1175
1150 M = 0: FOR B = 1 TO  LEN
   (A$)
1155  IF  MID$ (A$,B,1) <  >
   D0$ THEN 1165
1160 P$(0,A) =  LEFT$ (A$,B -
   1):P$(1,A) =  MID$ (A$,B +
   1, LEN (A$) - B + 2):B =
   LEN (A$):M = 1
1165  NEXT B: IF M THEN 1175
1170 V = V + 1: GOSUB 5200:
   PRINT "COMMAND ERROR":A = A
   - 1:V = V + 1: GOSUB 5200
1175  NEXT A: RETURN
1200  REM LIST
1205 Z = 10: IF CP$ = "" THEN
   RETURN
1210  GOSUB 5000: PRINT "PRESS
   S FOR SCREEN/P FOR
   PRINTER";
1215  GOSUB 6000: IF G <  > 80
   AND G <  > 83 THEN 1215
1220  GOSUB 5000: IF G = 80
   THEN  GOSUB 6500
1225 V = 1:H = 8: GOSUB 5200:
   PRINT CP$:NC = 0: FOR A = 0
   TO MX:NC = NC + 1
1230  IF P$(0,A) = "" AND
   P$(1,A) = "" THEN A = MX:
   GOTO 1255
1235  GOSUB 4800:H = 1: GOSUB
   5200
1240  IF P$(0,A) <  > "" THEN
   PRINT A;:H = 4: GOSUB 5200:
   PRINT P$(0,A);D0$;P$(1,A):
   GOTO 1250
1245  PRINT A;:H = 4: GOSUB
   5200: PRINT P$(1,A)
1250  IF G <  > 80 THEN  GOSUB
   4600
1255  NEXT A: GOSUB 6600:
   GOSUB 4700: RETURN
1300  REM  EDIT
1310  GOSUB 5000: PRINT "PRESS
   I TO INSERT": PRINT : PRINT
   "OR A TO ALTER/ADD"
1320  PRINT : PRINT "OR D TO
   DELETE MORE THAN 1 LINE"
1325  PRINT : PRINT "OR R TO
   RENAME PILOT PROGRAM"
1330  GOSUB 6000: IF G <  > 73
   AND G <  > 68 AND G <  > 65
   AND G <  > 82 THEN 1330
1335  IF G = 68 THEN  GOSUB
   4400: GOTO 1390
1340  IF G = 65 THEN  GOSUB
   4400: GOTO 1380
1345  IF G = 73 THEN  GOSUB
   4400: GOTO 1360
1350  GOSUB 5000: PRINT "TO
   REPLACE ";CP$;", TYPE: ":LT
   = LL:LL = LF: GOSUB 6100:LL
   = LT
1355 CP$ = A$:Z = 10: RETURN
1360  PRINT : PRINT "NO. OF
   (BLANK) LINES TO INSERT?";
1365  GOSUB 6100:NB =  VAL
   (A$): IF NB < 1 OR NB > MX
   - C THEN  PRINT "INVALID";
   GOTO 1365
1370  FOR I = 1 TO MX - C: FOR
   I1 = 0 TO 1:P$(I1,MX - I +
   1) = P$(I1,MX - I - NB +
   1): NEXT I1,I
1375  FOR I = C TO C + NB -
   1:P$(0,I) = "":P$(1,I) =
   "": NEXT
1380  GOSUB 5000: IF C THEN
   PRINT C - 1;" ";P$(0,C -
   1);D0$;P$(1,C - 1)
1385  PRINT
   C;P$(0,C);D0$;P$(1,C):Z =
   1:A$ = "EDIT": RETURN
1390  GOSUB 5000: PRINT
   "DELETE FROM ";C;" TO ? ";:
   GOSUB 6100
1392 ND =  VAL (A$): IF ND < C
   + 1 OR ND > MX THEN 1390
1394  FOR I = C TO ND:P$(0,I)
   = "":P$(1,I) = "": NEXT
1396  FOR I = ND + 1 TO
   MX:P$(0,I - ND - 1 + C) =
   P$(0,I):P$(1,I - ND - 1 +
   C) = P$(1,I): NEXT : GOTO
   1380
1400  REM  RUN
1410  FOR I = 0 TO R: FOR P =
   0 TO 1:I$(P,I) = "": NEXT :
   NEXT
1420 F = 0: GOSUB 5000
1430  PRINT : PRINT "RUN OF
   ";CP$: GOSUB 7000
1440  FOR A = 0 TO MX: IF  LEN
   (P$(0,A)) = 2 AND  RIGHT$
   (P$(0,A),1) <  > M$ THEN
   1470
1450  FOR J = 1 TO 7: IF
   LEFT$ (P$(0,A),1) = S$(J)
   THEN  ON J GOSUB
   3000,3100,3300,3400,3500,360
   0,3700
1460  NEXT J: IF Q THEN A = MX
1470  NEXT A:G = 0
1480  PRINT : PRINT : PRINT
   "END OF RUN": PRINT : IF SC
   THEN  PRINT "SCORE = ";SC
1490 SC = 0: GOSUB 4700:Z =
   10: RETURN
1500  REM  REPLACE STORED
   PROGRAM
1510 Z = 6: GOSUB 6700: RETURN

1600  REM  SAVE ON DISK
1610  GOSUB 6800:Z = 10:
   RETURN
1700  REM  LOAD FROM DISK
1710  GOSUB 6900:Z = 10:
   RETURN
1800  GOSUB 5000: PRINT
   "BYE-BYE": END
1900  REM INSTRUCTIONS
1910  GOSUB 5000: PRINT "T:
   TYPE WHATEVER IS ON THIS
   LINE"
1920 V = 2:H = 4: GOSUB 5200:
   PRINT "(MAX ";LL;"
   CHARACTERS/LINE)"
1930 H = 1:V = 4: GOSUB 5200:
   PRINT "A: WAIT FOR USER
   INPUT"
1940 V = 6: GOSUB 5200: PRINT
   "M: MATCH CHARACTERS WITH
   INPUT"
1950 H = 4:V = 7: GOSUB 5200:
   PRINT "FROM A: (, BETWEEN
   ITEMS AND"
1955 V = 8: GOSUB 5200: PRINT
   "END LINE WITH ')"
1960 H = 1:V = 10: GOSUB 5200:
   PRINT "J: JUMP TO LABEL
   (STARTING *)"
1970 V = 12: GOSUB 5200: PRINT
   "C:CLEAR THE SCREEN"
1975 V = 13: GOSUB 5200: PRINT
   "R: REMARK OR COMMENT"
1980 V = 14: GOSUB 5200: PRINT
   "S: + OR - NUMBER TO ADD TO
   SCORE"
1990  GOSUB 4700:Z = 10
2000  GOSUB 5000: PRINT "Y OR
   N ARE MODIFIERS AFTER MAIN"
2010 V = 2: GOSUB 5200: PRINT
   "COMMANDS (AND BEFORE :) TO
   GIVE"
2020 V = 3: GOSUB 5200: PRINT
   "ACTION IF THE LAST MATCH
   WAS"
2030 V = 4: GOSUB 5200: PRINT
   "CORRECT (Y) OR INCORRECT
   (N)"
2040 V = 6: GOSUB 5200: PRINT
   "* IS USED BEFORE EACH
   LABEL"
2050 V = 8: GOSUB 5200: PRINT
   "$ MUST PRECEDE A STRING
   LABEL IN"
2060 V = 9: GOSUB 5200: PRINT
   "  A: LINES IF IT IS TO BE
   USED"
2070 V = 10: GOSUB 5200: PRINT
```

→

```
" LATER IN (SAY) T: LINES"
2080 V = 12: GOSUB 5200: PRINT
  "E: OR END: STOPS THE PILOT
  RUN"
2090 V = 14: GOSUB 5200: PRINT
  "DONE IS USED TO END INPUT
  MODE"
2100 GOSUB 4700
2110 GOSUB 5000: PRINT
  "SAMPLE PROGRAM": PRINT
2120 PRINT " 0?R:NAME
  INPUT": PRINT " 1?*BEGIN":
  PRINT " 2?T:TYPE YOUR
  NAME"
2130 PRINT " 3?A: $NAME":
  PRINT "
  4?M:SUPERMAN,SUPERWOMAN,'"
2140 PRINT " 5?TY:DON'T BE
  FACETIOUS": PRINT "
  6?JY:*BEGIN"
2150 PRINT " 7?T:HELL0,
  $NAME": PRINT " 8?END:"
2160 PRINT : PRINT "NB:
  ALWAYS PUT A SPACE BEFORE
  $": GOSUB 4700: RETURN
3000 REM ASK
3010 PRINT : PRINT "  ";:P$
  = P$(1,A):L = LEN (P$):
  FOR K = 1 TO L
3020 IF MID$ (P$,K,1) < >
  "$" THEN 3040
3030 I$(0,F) = MID$ (P$,K):
  GOSUB 6100:I$(1,F) = A$:Z$
  = A$:F = F + 1:K = L: NEXT
  K: RETURN
3040 NEXT K: GOSUB 6100:Z$ =
  A$: PRINT : RETURN
3100 REM TYPE
3120 P$ = P$(1,A):L = LEN
  (P$): FOR K = 1 TO L
3130 IF MID$ (P$,K,1) = "$"
  THEN 3150
3140 NEXT K: GOSUB 3220:
  RETURN
3150 FOR U = K TO L + 1
3160 IF MID$ (P$,U,1) = " "
  OR MID$ (P$,U,1) = "" THEN
  Y = U:U = L + 1: NEXT U:
  GOTO 3180
3170 NEXT U:K = L: NEXT K:
  GOSUB 3220: RETURN
3180 U = Y:V$ = MID$ (P$,K,U
  - K)
3190 FOR H = F - 1 TO 0 STEP
  - 1
3200 IF V$ < > I$(0,H) THEN
  NEXT H: GOTO 3220
3210 PRINT MID$ (P$,1,K -
  1);I$(1,H); MID$ (P$,U,L -
  U + 1):H = 0: NEXT H:K = L:
  NEXT K: RETURN
3220 PRINT P$: RETURN
3300 REM MATCH
3310 M$ = "N":N = 1
3320 P$ = P$(1,A):L = LEN
  (P$): FOR E = 1 TO L
3330 IF MID$ (P$,E,1) < >
  D1$ THEN 3370
3350 IF MID$ (P$,N,E - N) =
  Z$ THEN M$ = "Y":E = L:
  GOTO 3380
3360 N = E + 1
3370 IF MID$ (P$,E,2) = " "
  OR MID$ (P$,E,1) = "" THEN
  E = L
3380 NEXT E: RETURN
3400 REM JUMP
3410 Q = 0: FOR D = 0 TO MX
3420 IF P$(0,D) < > "" THEN
  3440
3430 IF MID$ (P$(1,A),1,LL)
  = MID$ (P$(1,D),1,LL) THEN
  A = D:D = MX: NEXT D:
  RETURN
3440 NEXT D:V = V + 1: GOSUB
  5200
3450 PRINT : PRINT "JUMP TO
  UNFOUND LABEL FROM LINE
  *";A:Q = 1:V = V + 1:
  RETURN
3500 REM CLEARSCREEN
3510 GOSUB 5000
3520 RETURN
3600 REM END
3610 A = MX: RETURN
3700 REM SCORE
3710 TS = VAL (P$(1,A)): IF
  TS > 998 THEN SC = 0:
  RETURN
3720 SC = SC + TS: RETURN
4300 GOSUB 5000: PRINT MX;"
  IS THE LAST POSSIBLE
  LINE.": PRINT
4310 PRINT "DISK DRIVE USERS
  MAY CONTINUE": PRINT "BY
  SAVING THE PILOT PROGRAM"
4320 PRINT "AFTER INPUT OF
  THIS LINE ";MX: PRINT :
  PRINT "THEN AMEND LINE 130
  OF THE"
4330 PRINT "PILOT INTERPRETER
  TO INCREASE": PRINT "THE
  VALUE OF 'MX' BEFORE
  RELOAD"
4340 GOSUB 4700: GOSUB 5000:Z
  = 10: RETURN
4400 GOSUB 5000: PRINT
  "STARTING AT LINE? ";:
  GOSUB 6100:C = VAL (A$)
4410 IF (C < 1 AND A$ < >
  "0") OR C > MX THEN 4400
4420 RETURN
4500 FOR A = 0 TO MX:P$(0,A)
  = "":P$(1,A) = "": NEXT :
  RETURN
4600 REM PAGING LISTING
4610 IF NC < NL THEN RETURN
4620 NC = 0: GOSUB 4700: GOSUB
  5000:V = 1:H = 0: GOSUB
  5200: PRINT CP$: RETURN
4700 REM PRESS SPACE BAR TO
  CONTINUE
4710 V = 23:H = 1: GOSUB 5200:
  PRINT "PRESS SPACE BAR TO
  CONTINUE"
4720 GOSUB 6000: IF G < > 32
  THEN 4720
4730 G = 0: RETURN
4800 REM SET TABS FOR LINE
  NUMBERS
4810 V = V + 1:H = 1: IF A <
  100 THEN H = 2
4820 IF A < 10 THEN H = 3
4830 GOSUB 5200: RETURN
4900 FOR I = 1 TO LL + 5:
  PRINT "*";: NEXT : RETURN
5000 HOME : RETURN
5200 V = V - (V = 25): VTAB V:
  HTAB H: RETURN
6000 GET A$:G = ASC (A$):
  RETURN
6100 REM INPUT CHECK
6110 S$ = ""
6120 GOSUB 6000:X = ASC (A$)
6130 S = LEN (S$): IF X = 8
  AND S > 1 THEN S$ = LEFT$
  (S$,S - 1): PRINT A$;: GOTO
  6120
6135 IF X = 8 AND S = 1 THEN
  S$ = "": PRINT A$;: GOTO
  6120
6140 IF X = 8 THEN 6120
6150 IF X = 13 THEN A$ = S$:
  PRINT : RETURN
6160 IF X > 31 AND X < 91 AND
  S < 33 THEN S$ = S$ + A$:
  PRINT A$;
6170 GOTO 6120
6500 REM SET PRINTER
6510 PR#1: RETURN
6600 REM DISABLE PRINTER
6610 PR#0: RETURN
6700 REM REPLACE
6710 GOSUB 5000: PRINT "DO
  YOU WANT TO DELETE THE
  EXISTING FILE": PRINT :
  PRINT "OF ";CP$;" ?"
6720 PRINT : PRINT "PRESS Y
  OR N"
6730 GOSUB 6000: IF G < > 89
  AND G < > 78 THEN 6730
6740 IF G = 78 THEN Z = 10:
  RETURN
6750 PRINT : PRINT "REPLACING
  NOW"
6760 PRINT D$;"UNLOCK";CP$:
  PRINT D$;"DELETE";CP$:
  PRINT "WAIT 5 SECONDS":
  RETURN
6800 REM DISKSAVE
6810 PRINT : FOR A = MX TO 0
  STEP - 1: IF P$(0,A) = ""
  AND P$(1,A) = "" THEN NEXT
6820 PRINT D$;"OPEN";CP$
6830 PRINT D$;"WRITE";CP$
6835 PRINT A: PRINT CHR$
  (34)
6840 FOR I = 0 TO A: PRINT
  P$(0,I): PRINT CHR$
  (34);P$(1,I); CHR$ (34):
  NEXT
6850 PRINT D$;"CLOSE";CP$
6860 PRINT D$;"LOCK";CP$
6870 RETURN
6900 REM DISKLOAD
6910 PRINT : HOME : INPUT
  "PROGRAM NAME? ";CP$: PRINT
  "WAIT": GOSUB 4500
6920 PRINT D$;"OPEN";CP$
6930 PRINT D$;"READ";CP$
6935 INPUT A,B$: IF A > MX
  THEN 6945
6940 FOR I = 0 TO A: INPUT
  P$(0,I),P$(1,I): NEXT I:
  GOTO 6950
6945 FOR I = 0 TO A: INPUT
  A$,A$: NEXT I
6950 PRINT D$;"CLOSE";CP$
6955 IF A > MX THEN PRINT
  "PILOT PROGRAM TOO LARGE":
  GOSUB 4700
6960 Z = 10:G = 0: RETURN
7000 FOR I = 1 TO 2000: NEXT
  : RETURN
8000 END
```