A Database Publication

apple user

Vol. 5 No. 6 June 1985 £1

Making music on your micro

Son et lumière with Mugraph . . . add music to your programs with Ampersound . . . how to turn Mac into a maestro . . . and Apples in the music industry

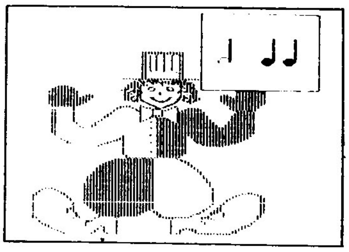
Start of a new series of Pascal tutorials

Volume 5
Number 6
June 1985

News

● Atari's threat to Macintosh, how an Apple IIc survived the ascent of Everest, North Sea gas monitoring, Royalty takes a hand, and more. 5

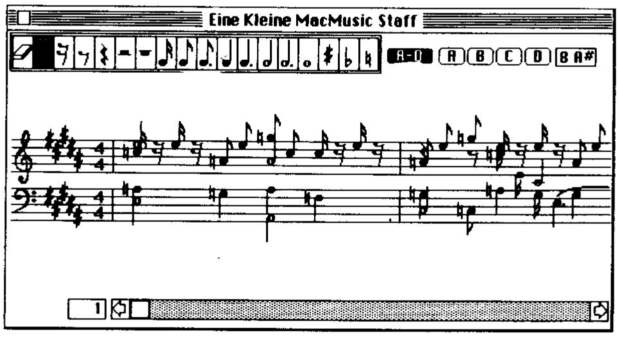
Fun & Games



● The Music Construction Set, Song Writer and Music Readiness keep our reviewers busy and provide the fun and games this month. 9

Application

● Michael Brice gives an insight into how Apples help keep track of five music companies. 17



Make Mine Music

● 1985 is European Music Year so this month we're making our contribution with hardware, software, applications and reviews

Pascal

● Stuart Bell begins a new series of Pascal Tutorials with a look at records – but not the musical kind. 19

Review

● Kevin Farrell looks at Tick-Tack, a useful translation package for foreign correspondence. 25

Macintosh

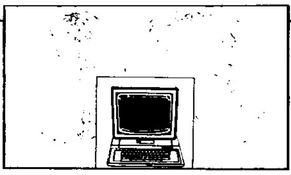
● Bill Hill demonstrates his versatility on the ivory key-caps with Musicworks. 30

Graphics

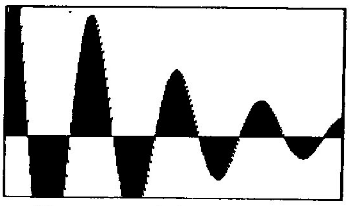
● Peter Gorry takes time out from programming the Apple User Graphics Library to review three books on graphics. 39

MicroLink

● Your chance to join MicroLink, the new micro communications system tailored to your needs. 34



Mugraph



● It's surprising what you can do with a light dependent resistor. Jim Dawson shows you how. 44

Programming

● Mark Bowyer gives an ampersand route to music and sounds in your own programs. 49

Feedback

● Connecting a Koala pad to an Apple IIc, single drive Pascal, screen output in Pascal, and more. 53

New Products

● A small sample of the newest ways to spend your money. 62

Order form

● Subscriptions, back issues, binders to keep them in. 66

Classifieds

● Bargain offers from your fellow Apple users. 70

Apple User Managing Editor **Derek Meakin** *Features Editor* **Cliff McKnight** *Technical Editor* **Max Parrott**
Production Editor **Peter Glover** *Advertisement Manager* **John Riding** *Sales* **Chris Riding**
Design **Heather Sheldrick** *News Editor* **Mike Cowley** *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.

News trade distribution: Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Circulation 0293 27053.

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

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.

While every care is taken, the publishers cannot be held legally responsible for any errors in articles or listings.

ABC 12,850 – JULY -DECEMBER 1984

Subscription rates for
12 issues, post free:
£12 UK
£13 Eire (IR £16)
£20 Europe
£15 Rest of world (surface)
£30 Rest of world (airmail)

Apple's tops on Everest

AN Apple IIc had to function at a height of over 21,000 ft and endure temperatures as low as −25°C during mountaineer Chris Bonington's successful Everest expedition.

But the machine came through with flying colours according to reports from the world famous climber.

Bonington who attacked the mountain with eight Norwegian climbers and 28 Sherpas used the Apple to monitor supplies, write letters home, and compose his articles for the Observer newspaper.

"He's obviously delighted with it", said his secretary, Louise Wilson, after receiving letters written by Bonington at his base camp and posted in Katmandu, Nepal.

The ascent was planned from start to finish with the aid of a specially designed software package, Everest Logistics, written by Ian Holt, and the new Apple IIc with flat panel display and portable solar panel power pack.

Bonington used two Apple computers during the eight months it took to plan the expedition and said before he left for Nepal that he expected his IIc to save the team a great deal of time in logistical planning, stock control and accounting.

The computer played a key role in solving the complicated logistical problem of moving supplies up the mountain in the most efficient manner.

Chris Bonington operates his Everest Apple

ATARI ST THREATENS MACINTOSH

A SURVEY of leading UK computer journalists has revealed that almost all are convinced that the imminent arrival of the Atari 520ST will pose a serious threat to the Macintosh.

This is due to the fact that the Atari machine contains the Gem operating system which provides it with a Mac environment at less than half the price.

In each case, the writers had witnessed the unveiling of the 520ST at the recent Hanover Show.

And only one of the experts gives the Apple machine a chance of emerging from the forthcoming battle unscathed.

The lone dissenter was John Lettice, features editor of Personal Computer News.

"As far as I am concerned, the Mac looks to me the more sophisticated machine", he told *Apple User*.

"So on current pricing, it will retain its share of the higher priced machines market".

The views of journalists who expressed concern over the future of the Mac were:

"I would like an Apple Mac but cannot afford it. So if I can get something that works in the same way at a much lower price, then obviously like so many other people will I would go for that. To me the ST seems the obvious answer" – *Christina Erskine, Popular Computing Weekly.*

"I see it (the 520ST) doing considerable damage to the Macintosh considering the Apple price structure" – *Jack Schofield, Practical Computing.*

"If they produce on time and market the ST successfully, with the Macintosh at its present price its future looks bleak" – *Geoffrey Ellis, The Times.*

"There may be difficult times ahead for the Macintosh" *Nick Walker, Personal Computer World.*

"The ST is almost certain to challenge the Macintosh. It will fill the gap for businessmen who were put off the Mac by its £2,000 plus price tag" – *Les Ellingham, Page 6 Magazine.*

"It would seem that the Macintosh has a real rival which costs considerably less" *Ralph Bancroft, Microscope.*

However Times man Geoffrey Ellis was able to offer a word of advice to counteract the threat from the 520ST.

"If I was selling the Macintosh I would bring the price down", he said, "for if Apple decided to upgrade them all to 512k and make prices competitive with Atari they will win because of their name and toehold in the market".

Mine of information

AN 1845 tonne electric powered dragline for a Northumberland open cast coal mine has been finished on schedule and much of the credit has gone to a project management program used on the Apple II.

Micro Planning Software's Microplanner system was used by Taylor Woodrow Construction both for pre contract planning and ongoing planning during construction.

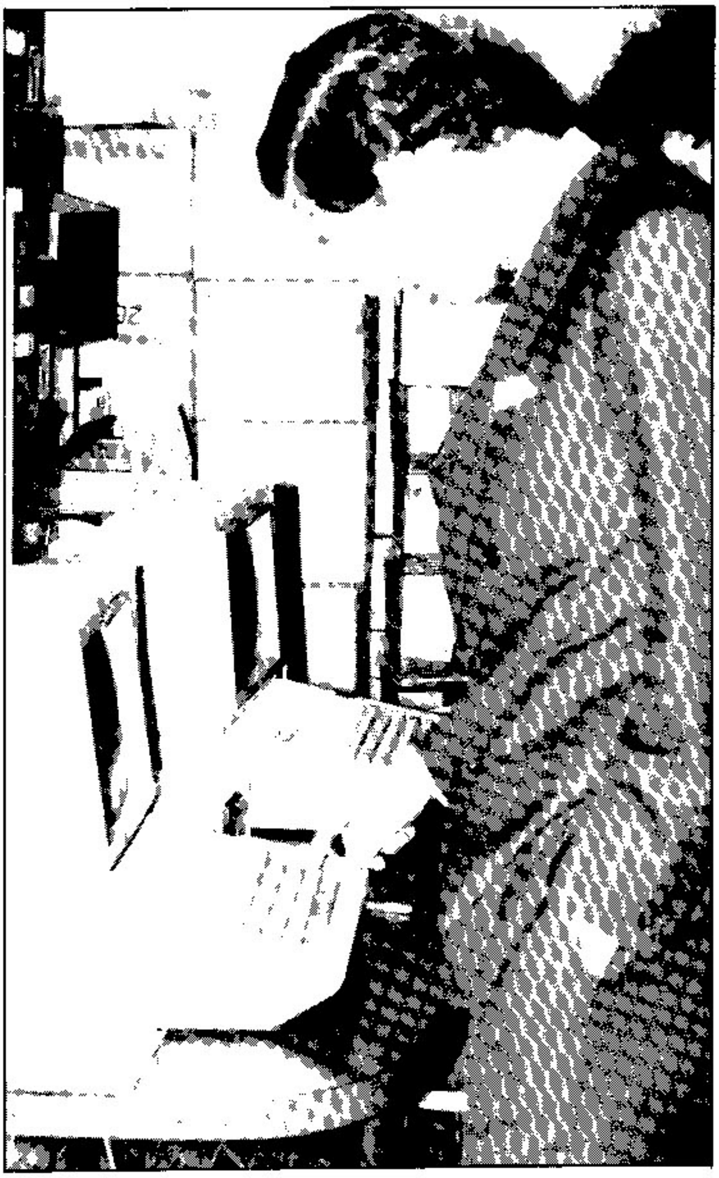
VC offer

EXCLUSIVE UK distribution rights to VisiCalc for the Apple II has been won by Thorn EMI Computer Software.

Thorn are offering it at £99, a 30 per cent reduction on the previous £149.

Apple dealers were able to see this new addition to the company's business software at the Apple '85 show.

Charles tries an Apple



APPLE Computers joined with other leading British companies to assist a new scheme set up by the Prince's Trust to help unemployed youngsters.

Youth Meets Industry gave 250 jobless 18 to 25 year-olds from all parts of the UK the chance to learn the basics of business practice.

One of the main features of the one week course was a computer room containing 20 Apple systems running educational and business software.

Anthony Kenny, one of the Trust's principal organisers, said: "One of our first aims was to include a beginners' course on computing.

"Virtually al this group passed through the educational system before computers were readily available in schools, and as a knowledge of computing is so essential to many forms of employment these days we hope to have introduced them to this skill during the week".

Experience

David Hancock managing director of Apple UK, said: "We're pleased to be associated with the Prince's Trust. By giving these young people experience of working with computers I hope we give them a better chance of a good job when they have completed the scheme".

The Prince of Wales visited the Youth Meets Industry centre at Middleton-on-Sea, Sussex, during the course and used one of the Apples himself as our picture shows.

More costs cuts for schools

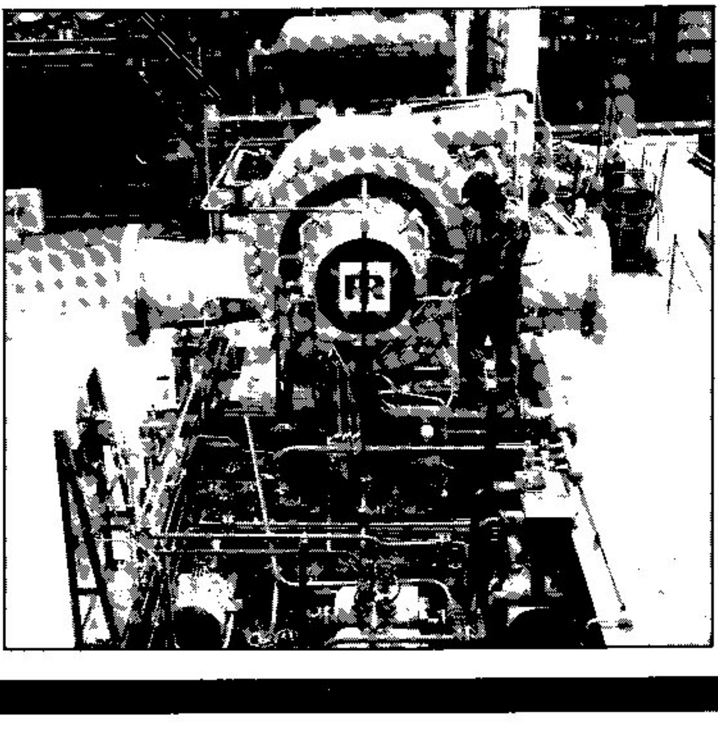
THE "Our Kids Can't Wait" campaign, to win Apple products a major share of the education market, has been given a big boost.

Symbiotic Computer Systems, leading supplier of Winchester hard discs and local area networks for Apple micros, has substantially cut its prices.

The firm is offering its full range of Apple networking products at 50 per cent discount to any school eligible under the "Our Kids Can't Wait" programme.

In an attempt to gain a 25 per cent share of the education market by July, Apple earlier slashed the cost to schools of the Apple II range by 50 per cent and the Macintosh by 30 per cent.

The company's UK managing director David Hancock has predicted that Apple will be number one in sales to British schools by early next year.



Apple has gas taped

JUST like oil, North Sea gas is pumped ashore through pipes. Only the pumps are called compressors when gas is being moved.

When compressors like the one in the picture are being designed and tested it is very important to know the properties of the gas they have to handle.

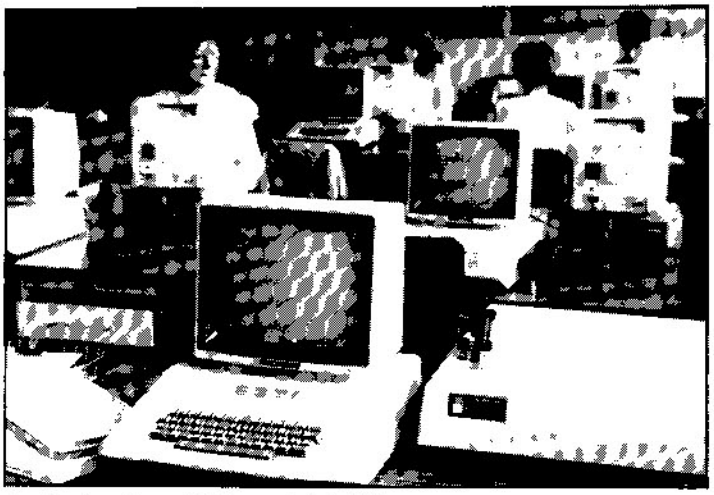
Using an Apple IIe, Rex Grimshaw of Ingersoll-Rand and Dr Bob Brown of the Lancashire Polytechnic are able automatically to record the temperature and pressure of gas mixtures and then calculate other properties of interest.

Apples control deluge of those special offers

SPECIAL offers for magazine readers are big business, and the volume of response for a major publisher like IPC is phenomenal – at any one time 200 different products may be on offer, generating thousands of orders each day.

Such an immense daily postbag could mean chaos. The fact that it doesn't is due to a network of Apple computers with specialised software.

Four years ago, it took a mini computer to control the deluge – an expensive turnkey system requiring sizeable charges for every minor change to the software.



Apples in action at IPC magazines HQ

Jim Sheehy, a manager in IPC's editorial offers division, felt the sum of several micros could have the same impact as a mini, so he set about finding a suitable system.

Nestar Systems had recently launched a networking system for Apple computers called Cluster One.

IPC chose it, installed it, and over the next two years it grew to encompass 66mbytes of memory and nine Apples.

Unfortunately, with this number of workstations there was a noticeable degradation in response times.

"We found the system was very reliable", says Sheehy, "despite our early Heath Robinson set-up with wires and cables everywhere.

"This gave us the confidence to solve our problems by upgrading to Nestar's latest networking system Plan 4000.

"We now have a central file server with 137mbytes and 24 Apples networked off it, and there is only a slight amount of degradation in response times if more than 20 are working at a given moment.

"With more than six on the old Cluster One it would grind to a halt.

"Transferring all our data files might have been a problem but in fact nothing could be simpler. We gave Nestar all the information on data cartridges and it was back to us within 48 hours.

"As we are currently running both Plan 4000 and Cluster One in the offices, we occasionally transfer data ourselves now".

The software which runs on the system has been entirely developed in-house.

All incoming orders are input to the system which generates the appropriate mailing labels that are sent in batches to the relevant manufacturer for sending out the goods, along with picking lists and cash-accounting documentation.

Detailed statistical breakdowns on responses are produced.

It is IPC's intention to develop the system to the stage of becoming a "paperless office" with all the data either going on to discs for despatch to suppliers and credit card companies, or direct via modem links.

The company is looking at replacing the old Cluster One with a second Plan 4000 to develop its growing interest-free extended credit facilities, keeping track of each account which is mailed or referenced several times.

The last part of the operation to be put on to an Apple network is refunds which are currently handled on old NCR 32 accounting machines.

The software required is proving to be quite complex because of the amount of security that needs to be built in.

More on Gold standard

TELECOM Gold has extended its telex facility to enable customers to receive incoming telex messages on their mailboxes.

Incoming telexes are sent to a central telex number and routed automatically through to the recipient's Telecom Gold mailbox number.

This gives Telecom Gold subscribers the additional advantages of being able to receive telexes confidentially and without delay, in the office, at home or while travelling.

The telex can be acknowledged or replied to instantly, be forwarded on to another recipient or electronically filed for future reference.

A simple routine is also available which will convert standard telex upper case format into lower case, making text simpler to edit.

Complex text can be prepared and edited on the Apple and then sent to both electronic mail and telex users in one simple operation, without re-keying the message.

Directories of familiar names can be organised to simplify addressing of telexes, and the facility to set up distribution lists enables simultaneous transmission to up to 500 different recipients.

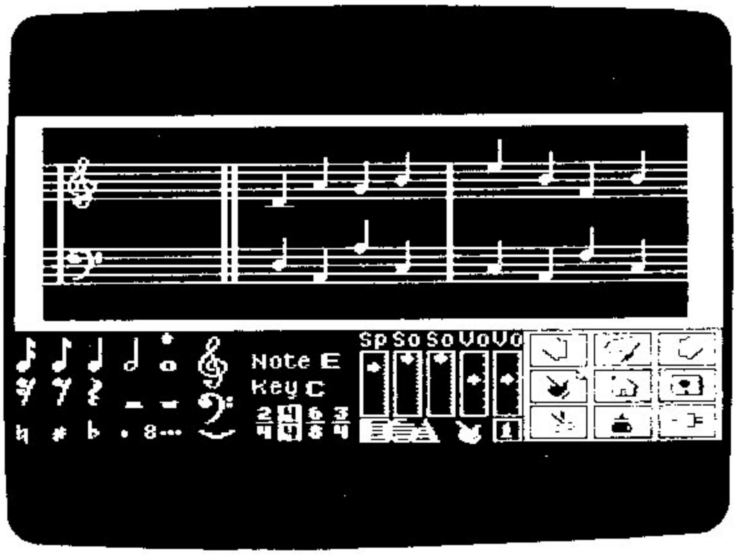
The new system enables electronic mail users to communicate with up to $1\frac{1}{2}$ million worldwide telex subscribers, including ships at sea, via the Inmarsat satellite link.

Mac XL dropped

PRODUCTION of the Macintosh XL is to be discontinued, just three months after the renamed Lisa 2/10 was being touted by Apple as having "a key role in the Macintosh Office".

Now Apple says it will market one modular Macintosh product line – "one mainstream Macintosh that can be expanded with a variety of peripheral products", said a spokesman.

"The Macintosh XL, which includes 10 mbytes of internal hard disc storage, will be replaced by a more powerful modular system consisting of a high performance 20 mbyte external hard disc drive added to a 512k Macintosh".



Explore music with those Beethoven building blocks

THE Music Construction Set is like building-blocks for Beethoven, Lego for Liszt or Meccano for Mozart. In short, it's exactly what you'd expect something called Music Construction Set to be.

It enables you to build up, edit, and play tunes in a similar manner to that used by the Pinball Construction Set, or indeed by many Macintosh programs.

You use a joystick or mouse to move a pointing hand around the screen and pick up the various bits that you want.

These may be notes, rests, ties and so forth to be placed on the staffs as you compose your magnum opus.

Alternatively, you can use the pointer to select the time signature, change the speed or select one of the icons – they're the nine little boxes in the bottom right-hand corner of the screen.

These allow you to scroll through the music (using the arrows), play the music (using the piano, of course), cut and paste from one tune to another and access the disc.

The house is not a house but a home – just like the Applesoft Home, it returns you to the beginning.

You can transpose the piece into another key as long as the result doesn't take it outside the range that MCS can handle.

Also, the program won't let you change key if the treble and bass clefs have different key signatures. Mind you, my composing only uses such techniques by accident.

To get the full benefit from the MCS you need some form of auxiliary sound system.

The Apple speaker is not really up to the job and if, like me, you have a fan fitted to keep the machine cool, the din it makes drowns out much of the music.

Minimally, you can direct the sound to the cassette port and thence to your stereo system.

However, using either the internal speaker or the cassette port you can only play four notes simultaneously.

With a Mockingboard in your II+ or IIe or a Cricket connected to your IIc, you can play up to six notes simultaneously.

You can use the on screen volume and sound controls which aren't available via the speaker or cassette port, and the music will scroll automatically as it plays.

Having written your piece, or loaded one of the pieces from the disc, you can print it out if you have a suitable printer and an interface card that is compatible with the Apple parallel card.

It worked fine with my Epson FX-80 and Digitek SPIII card, although it had a stretched-out appearance.

Because it's printed out in bit image mode, though, you can't get rid of the stretching by selecting condensed print mode. The music is printed sideways so that it gets printed as one long, continuous piece.

I must admit I've always been a sucker for construction sets, and the MCS is no exception.

I had fun playing with it (in both senses of the word) although I soon got fed up with using the joystick.

Fortunately, there are several keyboard "shortcuts" which can be used in conjunction with the joystick and these are listed on the reference card that accompanies the manual.

If you're using a IIc, you can select keyboard-only input because the "closed-Apple" key can be used in place of the joystick button.

It's a pain having to select input device, sound output channel and printer type each time the disc is booted – particularly since the review copy had a habit of rebooting when trying to perform a Catalog operation.

I would have preferred the ability to set these parameters once and be offered the option to change them on boot-up.

The Music Construction Set is not so much a composing tool as a musical exploration device.

For my money, it's ideal for playing *with* music rather than for playing music.

With this attitude (and to be fair, it's one that the manual suggests), the 700 ish symbol limitation on the length of any piece fades into insignificance.

Music, micro, please!

Cliff McKnight

Title: Music Construction Set.
Author: Will Harvey.
Publisher: Electronic Arts.
Requirements: 48k Apple II family.

❭

Composition ... but with some limitations

SONGWRITER allows the user to write music, save to and retrieve from disc, and play it back. It is a very simple matter to get from one mode to another.

The sound output from the Apple may be played directly into a tape recorder or a music centre, and a suitable lead is supplied for this purpose.

Having managed to boot the disc (I had to try four different drives before I could get the review copy to boot), the basic screen layout consists of a two-and-a half-octave section of a piano keyboard with black and white notes displayed in blue and orange respectively.

All the keys are shown the same length, and I feel it would be easier to use if the black notes were shown shorter, as on a real keyboard.

It is possible to move the whole keyboard left or right so as to have the correct layout for the key being used for the current tune.

On the left of the keyboard is displayed the note length as a fraction and, optionally, the name of the note being entered or played.

The program uses fractions to indicate note lengths and a range from 1/24 to 24/24 is available, which should be ample for most purposes.

In practice you set the longest note to be used in a piece as a fraction with a value of 1 (say 16/16), and all other notes are calculated as fractions of this. So, if a semibreve is used as the base, then a crotchet becomes 4/16, a quaver 2/16 and so forth.

This means that you have to make sure you've found the longest and shortest notes before starting, since you cannot use double-length notes.

Any permutation of fractions is possible to suit the music being written. The value of the fraction is set by means of the +, –, / and * keys.

You move up and down the keyboard by means of the arrow and < > keys, which move you a tone and a semitone respectively.

Having chosen your note and its length, you press the spacebar to "record" it.

If you don't like what you've done, you enter X to cancel it.

The notes as entered are depicted above the keyboard by variable length blocks rather like a pianola roll.

It is possible to play a note at a time, go back a note, change, insert or delete notes, and generally edit your tune as much as you like as you go.

The tune may be saved to disc at any time, or other tunes may be loaded from disc and combined with the piece already in memory.

I didn't manage to run out of memory even when I loaded all the example tunes from the disc and played a concert of nearly 20 minutes!

A rather less-satisfactory facility is the "idea" mode.

If you wish to be able to repeat sections of a tune, say a phrase or a refrain, it is possible to record this as an "idea", by pressing L.

After the L, all subsequent notes entered are saved as an "idea" file, which may be used as a whole later on.

At the end of the idea session, entry of a letter such as Q terminates the idea file, and composition reverts to normal.

To recall the idea, you move the cursor to the appropriate starting note, press the letter which ended the idea (say Q) and the entire idea is repeated as originally keyed in.

The manual says that you may enter up to 100 steps in an idea file, but as these include instructions to change note length, move up and down the scale, etc. the actual number of notes is rather limited – it certainly wasn't enough to store the refrain from Yellow Submarine!

We tried this program out on the family and showed it to a music teacher, and the general opinion was that it was quite entertaining for a while, but had several shortcomings.

As it has only two-and-a half octaves, the actual range is rather limited.

Since the "keyboard" can be moved up and down, it would seem to have been a simple matter to extend the range of notes.

The type of sound used is rather unimaginative.

Although the Apple doesn't have the advantage of sound envelopes, it is quite capable of producing sounds of varying timbres, and the SongWriter sound becomes rather monotonous after a time.

However, it is possible to play about with short pieces, and quite a lot may be learnt about the effects of changing speeds, note lengths etc.

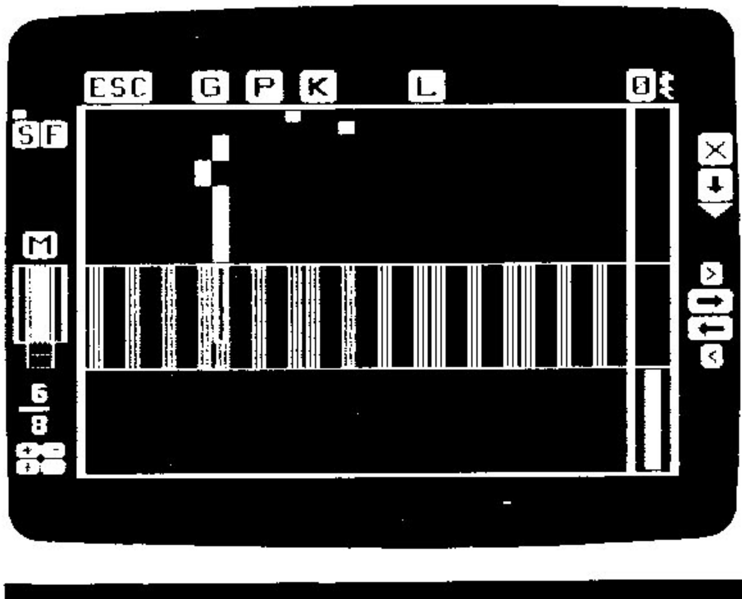
So, as a backup to formal studies or as an introduction, it could be quite useful, although I don't feel one could actually learn musical notation from it.

A useful attribute is that tunes saved to disc may also be used in Basic programs, which could save a programmer an awful lot of PEEKing and POKEing.

The manual supplied with the program is obviously aimed at the younger user, but is well written and very easy to work through.

Frank Lewis

Title: SongWriter.
Authors: Art Bardige and Samuel Wantman.
Publisher: Scarborough.
Requirements: 48k Apple II family.



The SongWriter screen – the notes look like a pianola roll.

Playing music... by playing games

MUSIC READINESS by Dinah Embry is designed to teach elementary pitch and rhythm concepts to young children by playing a series of games of graduated difficulty.

The target age range is 3 to 8 years for the Pitch series and 4 to 8 years for the Rhythm series.

The slightly older starting age for the latter is because the games depend on the player being able to count up to four and use the numerical symbols 1 to 4.

Each of the two discs consists of several games (five on the pitch disc and four on the rhythm disc) progressing from very simple to rather more complex musical concepts.

While the sequence can be used in teaching, a useful feature is that each game is independent of the others.

Hence, a child can enjoy playing a game for its own sake, having first become familiar with the relevant concept.

I found that children who already had some musical knowledge also liked to play the games, which served as useful and pleasant revision for them.

All the programs are easy to use.

After choosing a game from the menu, games begin on pressing the S key and the player can quit by pressing Esc.

If the player does not quit, the games terminate after a reasonable sequence of "goes", and the player is given feedback about performance level – either a "success" message, or a "review needed" message.

For some games the messages are accompanied by a percentage correct score.

The most notable feature of these programs is the attention to detail in providing feedback as the game progresses.

The duration or pitch of the sound is related in a meaningful and systematic way to the response required of the child, the appropriate musical symbol or keyboard position and to the feedback from the cheerful graphics.

Two manuals are provided, one for the parent/teacher, and one for the student.

The parent/teacher manual is clearly written and presented, and does not require prior musical knowledge on the part of the parent.

The student manual is not a do-it-yourself teaching aid, but supports the information given by the parent or teacher.

Although it is printed in large type, younger children would certainly not be able to follow it alone, but once it had been explained to them would be helped by the simple pictures.

It contains exercises for the child and examples of concepts such as "high" and "low" as applied to notes.

Other examples and exercises refer to rhythm and the names for note values. These are not the names always used, certainly by British teachers – for example, "half note dot" for what is, to me, a dotted minim or "three count" or "three beat" note.

However, this terminology does not appear on screen, so the child can be taught whatever conventional names the teacher prefers without affecting the games at all.

For young children with some knowledge of music and Apples, the system is so straightforward and well-designed that it is possible to play the games without reference to either manual.

I can say this with some confidence because my six year-old son decided to help me with this review, and played the games before I did (and commended them to me!).

However, he appreciated more features after explanations based on the manual, especially the relevance of the more subtle details of the charmingly synchronised graphics.

Pitch series:

There are four games in the series beginning with Highlow, an exercise in elementary pitch discrimination.

In all the games on these discs, the child's action has a logical connection with the concept being taught.

In Highlow, a note is sounded. If the note is high, any key on the top row is to be pressed. If it is low, the spacebar is pressed.

If the child correctly identifies a high pitch, a bird flies up into a tree. If low, the bird flies down on to a fence post.

As with all the games, the child is given three chances for each question, after which the next question is presented.

At the end of this game, or on quitting, the bird sings if the player has been successful in learning to identify pitch, but simply flies silently off-screen if a review is needed!

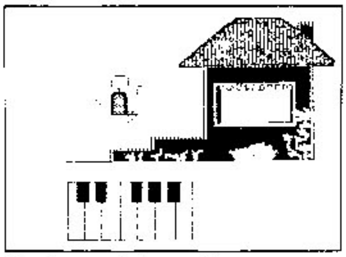
In the Updown game, the first five notes of the C major scale are played – the "five-finger exercise" – either rising or falling.

The student's task is to identify whether the scale is going higher (any top row key) or lower (spacebar). No opportunity for reinforcing the message is missed!

A keyboard (notes C to G) is also shown, below a picture of a ball and a slope.

If the direction of the scale is correctly keyed in, arrows appear above the keyboard, also pointing in the right direction, and the ball rolls up or down the slope as appropriate.

On ending or quitting, the C scale is played and the ball turns into a happy face if successful. Otherwise the ball rolls down the slope and a question-mark appears on it.


Updown: skip and jump

A nice, presumably intentional, touch applying generally to the games is that success messages are usually accompanied by cheery tunes, while most review messages are silent.

Percentage success also appears onscreen.

There are two Stairsteps games, the object of both being to teach the child to identify the distance between notes – seconds and major thirds, called steps and skips respectively in the games.

Once again, the physical representation of the concept makes sense, the child being required to press the arrow key in the appropriate direction once for a step and twice for a skip.

In the simpler game, the seconds and thirds are kept separate – seconds first which makes it easy for the child to learn what to do.

On screen is a picture of steps and after the child correctly responds to the two sounds presented, the child pictured onscreen jumps up or down either one or two steps mirroring the interval between the sounds.

The "ground" in the picture represents Middle C, step 1 is D and the top step is E.

Thus, if the pitches which are played are D to E, the child in the picture will jump from the middle step to the top one.

The author argues that she has used these three notes because they are the three most commonly encountered in learning piano. Importantly, given this, the notes are in fact correctly tuned.

In addition to the success/review message at the end of the game, a percentage correct is also given.

This reviewer must, however immodestly, report a score of 107 per cent on the easier game!

Ultra-superlative scores were encountered elsewhere also, as well as the occasional score which seemed "unfair", but as cumulative scores are not presented it is difficult to work out quite where the fault lies.

I think a cumulative score would be useful additional information, encouraging the child to try harder to maintain a good score.

In our trials, the reaction on seeing a percentage score after many trials was indignation rather than pleasure, even for high percentages.

In the final game on this disc, Keyboard, the steps and skips are related to a picture of a three-note keyboard (C to E).

A note is sounded and a spot appears on the appropriate note on the keyboard. Then a second note is sounded, and the player indicates whether it is a step or a skip, and the direction.

If correct, a second spot appears on the right note on the keyboard, and a butterfly sings.

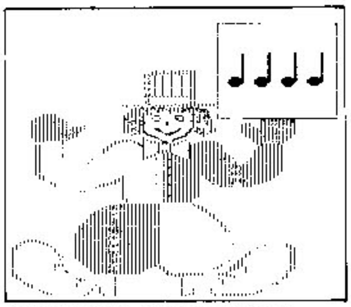
This is the one instance where a running total is shown, for all the butterflies remain on screen to the end of the game.

Rhythm series:

My heart sank at the thought of reviewing a package for elementary rhythm learning. I remember too painfully the anguished admonition "Count" in the early, and even much later, days of my music education.

I can still feel my equally-anguished reluctance to comply, perceiving this seemingly babyish counting out loud to be an outward, visible and hateful sign of inner ineptitude.

It is, then, with deep emotion that I bring you the news that the games on the Music Readiness Rhythm Series are a delight, with well-thought-out graphics making the learning of rhythm sounds and note values a pleasure.


The friendly Rhythmclown

True to form, though, I resisted the instruction in the manuals to clap the rhythms. How, I thought, can one be expected to sit at one's Apple and clap hands?

Well, by the time I decided to face the derision of my imaginary audience, I discovered that clapping definitely helped to structure the quite long (possibly over-long) note durations.

Teachers and students, please note – clapping the rhythms is not merely a trivially obvious exercise!

The Demo program uses the clown who appears in the student manual to teach note values.

A note duration symbol appears in one of his eyes, and the value of the note appears in the other eye. This is where the clapping first comes in.

Pressing the spacebar gives a demonstration of different note values.

The first of the game programs, Rhythmfish, was everyone's favourite.

After a note is sounded, its musical symbol appears (1 to 4 counts) and the student is to press keys 1 to 4 as appropriate.

If correct, fish appear, the same number as the note value, and wag their tails and open and close their mouths, also the same number of times as the note value.

If a review is needed, only bubbles appear at the end, but if there is a good performance, a larger fish comes and "sings"!

This was your reviewer's finest hour, for she was regularly able to score 111 per cent!

In Rhythmvan, numeric keys 1 to 4 again represent the note values 1 to 4 beats.

In this game, four lines differ in length in proportion to the note values they represent.

When a note is sounded, the symbol for that length of note appears at the end of the correct line.

If the player then presses the correct key, a van drives along the line.

The van disappears before the symbol does – another subtle touch, I thought, emphasising the importance of the lingering musical symbol.

Vans come out on every line and flash their lights and sing for success. One reverses silently offscreen if a review is needed!

In the last program, Rhythmclown, this friendly character is used to teach the values of a sequence of notes whose durations add up to the four-beat bar, or "common time", the first time signature most students experience.

I was impressed that, while the point about time signatures was nowhere stressed as part of the teaching, the design of the game implicitly prepared the student for further learning.

A series of notes is played, adding up to four beats, and the student must press keys 1 to 4 as appropriate in the same sequence.

After the sounds are heard, their symbols appear, taking the same length of time to do so as their values.

If successful on any question, the clown juggles stars, again synchronised with the duration of notes presented.

This is the only program in which the Return key has to be pressed to enter a response, but if the student is up to the game, then that certainly will not present any problem.

These rhythm and pitch packages are imaginative and entertaining ways to teach some fundamental musical concepts.

I was mildly sceptical of such an enterprise to begin with, but I was very impressed by the grading of the material and by the tremendous attention to detail.

The games are not a "teach yourself" scheme, nor are they intended to be.

But with the explanation given in the teacher/parent manual the child will rapidly be able to put in a lot of amusing and useful practice, without the constant attention of an adult – just what such a teaching scheme should do.

Someone needs to look at the percentage calculations – they are particularly frustrating for the older child just grasping the notion of percentages.

I did wonder, too, if the emphasis on the notes near Middle C was entirely sensible – some teachers are concerned that children become too Middle C centred.

I wondered if it might at least be possible to have an option which could give the same notes an octave higher or lower, to introduce the child to the idea that the relations between notes are consistent, regardless of their absolute pitch.

These are, however, minor cavils, and do not detract from the value of these packages.

I would like to see Dinah Embry produce versions for more advanced pupils – teaching the recognition of more-difficult pitch intervals and more-complicated rhythm patterns, at rather more challenging speeds.

While unlikely to be more popular with children than interstellar war games, they will nonetheless be enjoyed – and none of the friendly little graphics was ever sufficiently exasperated to leer at me from the screen shouting "Count!".

Valerie Kent

Title: Music Readiness
Author: Dinah Embry
Publisher: Sterling Swift
Requirements: 48k Apple II family.

Apple, core of the music business

By MICHAEL BRICE

THERE is no such thing as a "typical" Apple user, but if there were I doubt if Jim Hawkins would qualify for the title. In the course of his business, he uses to the full all the potential of his Apple equipment. Let me explain.

Lots of people use their Apple computer to help them run their businesses. But can you imagine what it would be like to operate five companies at once without an Apple to help you?

That was the situation facing London based Jim just a couple of years ago. It's not surprising that he went shopping for a computer to take over the administrative workload.

The sleeve for "Make Believe" by Naked Lunch the first record sleeve to be designed on a Macintosh

After looking at everything on the market, Jim bought an Apple IIe with Jarman Sales Ledger and Computech Bought Ledger software.

Why? "Because I was convinced that it would help me keep my accounts in order and send out statements each month far more efficiently than I could", said Jim.

Jim Hawkins is in the music business, and it was in a trade newspaper where Jim first heard about Apple.

"It was in 1981 when I first read about the musical things an Apple can do", Jim explained. "It was only after I'd read about the musical applications that I came across the business applications".

Now Jim uses Apple computers both for the administrative and artistic aspects of his work.

The paperwork for all five of Jim's companies can be brought up to date with an input time of just one half-day a month, generating statements, updating files and doing VAT analyses totally automatically.

A list of Jim's companies will give you some idea of the range and diverse nature of the cash transactions alone.

Jim operates Tickle Them Ivories, an electronic keyboard hire company for recording sessions and concerts, Bright Moments, a recording studio, the record company Plezure Records, the associated music publishing company Pleasure Unit and Pleasure Unit Productions.

The music industry is all about communication and speed.

Companies like Jim's flourish and prosper serving the big names. A call for the state-of-the-art Fairlight CMI for a Top of the Pops appearance by Duran Duran, or a lengthy tour of the US by Pink Floyd's Roger Waters are just two examples of the diversity of their work.

Work only achieved by establishing a reputation for a fast, friendly and reliable service.

This is an obvious area where Apples can help, by maintaining strict schedules and record-keeping.

The music business world is also one which runs on personal relationships. That's another application for Jim's Apple.

It can run a word processing program linked up to his Olivetti ET 231 typewriter to produce high-quality individualised mailshots, and keep a record of them at the same time.

It saves hours of repetitive typing and keeps total control over the correspondence records and filing.

What is probably more important to Jim is that the Apple allows him more time to develop those personal relationships by going out to promote his companies in person.

"I hate book-keeping and everything associated with it", said Jim. (It's a sentiment which perhaps all of us share.) "The less time I have to spend over it, the better".

Jim was so pleased with the performance of his Apple equipment that it was natural that he should look at the Apple Macintosh when it was introduced in 1984.

He was so impressed he bought one straight away, and now uses it for graphics.

Jim told me "I use it to generate ideas. I love the manipulation of graphic images".

Among other things, Jim uses the Macintosh to design record sleeves. In today's record market, they do more than just protect the record from dust. They must reflect the images created by the music, must grab the record buyer's attention, must sell the product.

The record cover shown was created for the single "Make Believe", by Naked Lunch, using the Macintosh to build on a simple idea.

Another area where speed is vital: the product must be out on the streets almost as soon as it is finished.

With the Macintosh, the artwork can be developed and finalised literally without having to put pen to paper.

Then, at a single keystroke, it can be produced cheaply and immediately.

Jim and his partner, Gray Levett, and indeed the whole Plezure Records staff are now eagerly waiting for the postman to arrive with a parcel from the United States.

They have ordered a device, called Thunder Scan, which utilises the Apple printer to input an ordinary black-and-white photograph into the computer.

You replace the ribbon cartridge with Thunder Scan's

›

reading head, position the picture where the paper normally goes, and load the program.

"Thunder" scans the photograph, and translates its tones and densities into a digitised format which the Macintosh can handle.

Once captured like this, the data can be manipulated by Mac to alter the original image in hundreds of different ways.

This new equipment is certainly going to enhance the company's ability to create new visual ideas to add impact to its product.

Talking to Jim, an ex-jazz rock drummer (he used to play with Robert Plant, before Plant went on to form Led Zeppelin) it soon becomes clear that music is his first love.

Let's go back to 1981, when Jim first became acquainted with Apple. "At the time, certain sequences were just beginning to happen, and some American companies, notably Alpha Syntauri, invented programs that enable musicians to sit at an electronic keyboard, and write music straight into the computer".

Running Tickle Them Ivories, Jim has to keep up to date with the advances happening to electronic keyboard instruments.

If your mind is turning to the electronic organ down at your local club or hotel lounge, then you might be surprised at what modern electronic keyboards can do.

Apparently, "The hottest products on the market, the Emu and the Kurzweil, operate on software that is specific to the Macintosh".

The electronic keyboard is now largely computer-driven. This is how it works.

Record any sound that you like into a microphone, from a church choir to breaking glass or worse. The computer will store the sample sound in digital form, on disc.

From one sound, you can generate many hundreds more in different keys, different sequences, and copy the sound of almost any conventional musical instrument.

All the while, the computer is filing away each sound and simultaneously remembering the melody line, as you compose. You can now "over record" background rhythms and chords.

As previously mentioned, Jim's company hires out these sophisticated instruments to famous artists for recording sessions or concert tours. So if you listen to modern music, you'll probably have heard one of Jim's keyboards in action.

He may even have helped to program it, using his Apple computer.

Jim Hawkins has the opportunity to use the computer very creatively, and he uses it to the full.

Taking an idea – a sound, perhaps, or a visual image, he lets the computer explore and develop it many times faster than one could do it by hand in any other medium currently available.

Without having to draft and redraft the idea, it is out and ready to transmit, on paper or tape, or disc.

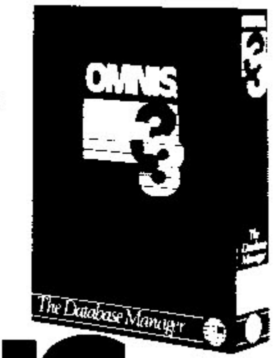
Jim, by his own admission, is not computer-minded. He simply wanted a tool that would relieve tedium, and get things moving fast, without getting in the way of creativity.

And Jim chose Apple equipment to do that for him.

Gray Levett adds: "The scope of the modern electronic keyboards, such as the Fairlight CMI, offers tremendous opportunities for musicians, and the Apple Macintosh will increase those opportunities further still.

"A tool of any kind is only as good as it can be applied and I see the Apple as just such a tool.

"For example, legendary rock keyboard player Nicky Hopkins (Nicky played the haunting piano melody on John Lennon's "Imagine") plans to use the Apple linked to synthesisers on his next film score".



For the record, we end without a beginning

STUART BELL starts a second series of Pascal tutorials by considering the use of records for storing related information

IN this, the first in a second series of Pascal tutorials, we shall look at the use of records to store groups of related information, and see how we can use the idea of a record to access the directory of a disc.

Every high level language has a number of data types. By this we mean a particular kind of information. For example, even the simplest Basic interpreter has integers, real numbers and strings, together with arrays of numbers.

Each of these is a "data-type". A record is a collection together of one or more of these different data-types into one unit.

It's rather like an array, except that in an array all the elements are of the same type, whereas in a record we can mix them.

For example, if we wished to store a person's name, together with his pay for the last 12 months, then this could be stored in a record comprising one string, and a 12 element array of reals. In Pascal, it would be declared like this:

```
type  personstuff = record
                        name : string[30];
                         pay : array[1..12] of real
                      end; (* of record *)
```

Note that this is one of the few times when we have an "end" without a "begin" in Pascal (the other time is the "case" statement).

We have now declared a type called personstuff. In the same way that we can declare variables to be integers or strings, we can now declare a variable to be a record with the structure set out above. For example, we might have:

```
var       count:integer;
        comment:string[20];
        details:personstuff
```

The variable "details" is a record which contains one string and 12 real numbers. If we wish to have a number of records in memory at any one time it is perfectly permissible to have an array of such records, so we might have:

```
var    dettable:array[1..20] of personstuff;
```

This would let us store the details of 20 people simultaneously. Obviously we shall soon run out of memory if we try to store too much. We can move whole records round with single statements without having to worry about each part of the record.

For example, the following will copy the record 'details' into the first location in "dettable":

```
dettable[1]:=details;
```

However we cannot input and print out whole records like that, and we will very often need to access individual parts of the record, such as to total up the pay received to date.

To do that we must specify the name of the record and that part of the record (called the field) that we are interested in. Let's assume that the person's name is stored in record "details", field "name". To input it, we would use the statement:

```
readln(details.name);
```

The full stop separates the record s name from the field name. To input the 12 amounts we might use:

```
for count:= 1 to 12 do
begin
  write('pay',count,' : ');
  readln(details.pay[count])
end;
```

This would prompt the person at the keyboard to input the 12 amounts. To total them up all we need to do is put this statement before the "end":

```
total:=total + details.pay[count];
```

You may now be thinking that it's rather laborious to write the "details." each time. Pascal allows us to avoid this by the use of the "with" statement, which we use like this:

```
with details do
begin
  (* in here, any reference to name or pay will be
     presumed to apply to the record called details,
     for example: *)
 writeln('the name is ',name)
end;
```

To display the whole of the array declared above, called dettable, we would use:

```
for person:=1 to 20 do
with dettable[person] do
begin
  writeln('name: ',name);
  for count:=1 to 12 do
    write(pay[count],'  ');
  writeln; writeln
end;
```

The use of records enables us to group related information together. In a stock control system, say, one record would be used for each type of stock item In a club, one record would be used for each member. In a system to record machine breakdowns one record might be used for each machine.

There are times when we wish to store records of different formats, according to the information at the start of the record. These are called "variant records". For example, imagine that we are storing maintenance details for equipment, then we could use:

```
type   sttype = (ok,us,underrepair);
                (* ie sttype can be one of these 3 values *)
       mdets = record
                 machinenumber : integer;
                   machinetype : string[8];
                   case status : sttype of
                       ok : numberoffaults : integer;
                       us : datefaultreported : string[8];
               underrepair : engineersent : string[12];
               end; (* of record *)
var    machinerec:mdets;
```

The record "machinerec" will always contain the machine number, machine type and its status. If the status is "ok", then the record will also have an integer to hold the number of faults, if status is "us", then the date on which the fault was reported, and so on.

We can then refer to variables like mdets.numberoffaults, or mdets.engineersent. Obviously you should only try to access datefaultreported if status is "us".

Each month I shall conclude with a useful program which demonstates the use that can be made of the feature of Apple Pascal discussed.

The example program below demonstrates the use of variant records to show you how your programs can access the directory of a disc in Apple Pascal. Each line of the directory is of the form described in "direntry".

Entries are of two types. The first one only applies to the first line of the directory, which stores information about the whole disc. The second applies to all normal file entries.

The directory consists of 78 entries and is stored in an array. To get the directory into the array we use the Unitread procedure described on page 41 of the Apple Pascal Language Reference Manual.

The program will list on the screen all files of type "text". You can easily alter it to list all files or just files of another type.

Making the user input the name of a file, and then looking for that name in the directory is quite a useful program with many uses in commercial applications

The most common use of records is to store them in data files on disc.

● *Next month we shall see how such files can be created and used.*

```
program dirread;

type
daterecord = packed record                  (* how date is stored *)
                 month:     0..12;
                 day:       0..31;
                 year:      0..100
             end;

 filetypes = (untyped,badblocks,code,text,info,
              data,graph,photo,dirheader);

  direntry =  record                        (* this is one line of dir *)
       firstblock:integer;
        lastblock:integer;
    case filekind:filetypes of              (* rest of record depends  *)
                                            (* on the file type:-      *)
       dirheader,                           (* this form for the *)
         untyped  : (volname:string[7]; (* first entry       *)
                     fl,
                     numoffiles,
                     numofblocks:integer;
                     lastbooted:daterecord);
                                           (* this form for all *)
       badblocks,                          (* other entries     *)
            code,
            text,
            info,
            data,
           graph,
               photo : (filename:string[15];
                        nbyteslastb:1..512;
                        lastaccess:daterecord)
      end; (* of record direntry *)

 var  directory : array[0..77] of direntry;
          line,
          drive : integer;
             ch : char;

 begin
    write('please give drive number: 4/5 ');
    repeat
      read(keyboard,ch)                   (* keyboard avoids echo *)
    until ch in ['4','5'];                (* to the screen        *)
    writeln(ch);
    if ch='4' then drive:=4 else drive:=5;

    writeln(chr(12));                     (* clears the screen    *)
    writeln('Text files on disc in £',drive);
    writeln;
    unitclear(drive);                     (* resets the drive     *)
    unitread(drive,directory,2048,2); (* warning: unitread corrupts
                                          the drive number!!        *)
    for line:=1 to directory[0].numoffiles do
      if directory[line].filekind = text then
        writeln('        ',directory[line].filename);
    writeln;
    writeln('Total number of files: ',directory[0].numoffiles);
end.
```

"Ou est la plume de ma tante".
"Grosse uhren gehen ticktack ticktack"

Try talking business – in another language

WE are notoriously bad in this country when it comes to foreign languages. How much business is lost by our obstinacy in refusing to go half-way towards our overseas customers in the use of their language in business deals?

Not any more, if Primrose Publishing have their way. Their Apple software packages are aimed at making life simple for the amateur linguist in writing well constructed letters in foreign languages for business or other purposes.

The principle of operation, so far as the Apple is concerned, is similar to the much-joked-about travellers' phrase book.

A collection of foreign-language sentences appropriate to a variety of classified situations is held in text files on a disc, in a numerical order.

The equivalent sentences in the same order in the mother tongue (English for us) are catalogued in a book.

The user thus has to compose the letter in English using sentences found from the catalogue of available sentences and noting the catalogue numbers of the sentences used.

After loading Applewriter and then the Tick-Tack disc, the operator uses it to write the letter, using the normal "load" command to load sentences off the foreign-language disc in the order noted down when the letter was designed.

The mechanics of the system are thus quite simple and any normal users of Applewriter could devise a set of sentences for themselves.

The value of any such system must lie in the thought and care put into the choice of the sentences provided, in the well-thought-out documentation, and in the linguistic expertise contained in the discs.

For review purposes, we received the "Executive Business Pack", which has an impressive standard of presentation. It contained four A4 booklets each of about 50 pages and two floppy discs.

The booklets included in the French/English package supplied for review were: The Operating Manual for the system with Apple; a Language Guide for Business (English); a Language Guide for Business (French); and a Letter-Writing Guide (French ... but written in English).

The Language guides contain the catalogue of sentences. So that, for example, the English language guide has as its sentence number A16 "I enjoyed talking to you on the telephone today".

Open the French Guide at the same page and sentence A16 reads "J'ai été très heureux de vous parler au téléphone aujourd'hui".

Similar language guides are available for many tongues, including Dutch, Swedish, Romansch and Basque.

Different conventions of letter layout seem to be acceptable for different languages and the Guide to Letter Writing (French) certainly helps in this respect.

With the pack, in a nice folder, came two discs. It was expected that the discs would contain the vocabulary of sentences referred to in the documentation and indeed for the most part this was so.

The booklets, however, go on to give a variety of complete letters and useful phrases. These sections, which would have been quite convenient, are not provided on the discs.

Initially the system would seem to be a bit tedious in use, with the need to search through the lists of sentences in English to find the appropriate ones.

After a while, no doubt familiarity would make the selection much quicker and certain sentences will find very common use while others are used rarely.

To add to the difficulties, it is necessary to switch from one disc drive to the other for sentences in the second half of the list and back again.

Switching drives with normal Apple DOS is a bit tedious and it would certainly be desirable when more familiar with the system to collect all the most commonly used sentences on to one disc.

One problem with many languages is the variety of written accents needed to make the letter look correct, and in many cases to read correctly, which are not readily available on the Apple screen or keyboard.

A binary file of appropriate characters for any particular language is provided on the disc for downloading into an 80-column card.

The use of this facility requires a Sup'R'term 80-column card. Although this fact is not made plain in the publicity material, the package itself makes the point.

To key in the accents, they must be allocated to some of the lesser-used key combinations of the Apple keyboard. Page 12 of the users manual acknowledges this fact and the manual promises "Details are set out below".

Alas, it looks as though the details just eluded the author at the critical moment.

The only way to find out what key to hit to produce, say, a grave accent is to find one among the sentences on the disc, to examine its form on the screen and to study the Applewriter manual to see how to produce it.

Even then there is no guarantee it will produce the required symbol if your printer does not follow Epson conventions or is not the appropriate daisywheel. (My printer produces a nice upward-pointing arrow in response to "shift N" which is quite unsuitable for the circumflex accent for which it is supposed to substitute in this software.)

On the other hand, inserting a few accents by hand is not an overriding difficulty, and to be fair the package does say an Epson or Qume is needed.

Studying the actual sentences provided gives no clue to "My aunt's umbrella has been lost on Strasbourg station", but does reveal a host of very useful business sentences such as "Our prices are subject to a discount of 2 per cent for payment by the 30th of the month following".

The key to their usefulness is that they can be readily modified with virtually no knowledge of the language concerned to suit a particular occasion.

Before rushing out to buy this package, however, careful consideration should be given to its applicability.

There is a danger involved in sending a letter to a foreign company written fluently in their own language. The danger is that they will assume that you have a fluent linguistic ability and reply at length in their own language.

Tick-Tack will not help directly with the translation in these circumstances from the foreign language back into English.

A well-founded convention in serious business dealings is that letters should be written in the language of the originator.

Since such letters may contain legal and contractual

›

implications, it is most important that the signatory of the letter understands clearly every word that he is signing and therefore prepares the letter in his own language.

There is a place, however, for the package in drafting informal correspondence.

It could speed up hotel bookings, enquiries for quotations and so forth as the correspondent may be then in a position to reply that much more quickly. In such cases there is no contractual obligation involved. Much correspondence comes into this category.

Some care is required in use of the package because language is a very subtle subject and apparently minor points such as word order, tense of the verb, singular and plural, can radically alter meaning or lead to ambiguity.

The package reviewed was a French/English package and the French can be more correct in their usage of some words and we English rather sloppy.

Words such as "engineer" or "secretary" can be used very loosely in English. A secretary can be anything from a typist to an important company executive in English, whereas the French will be careful to distinguish between "la secrétaire" and "la sténodactylo" when referring to a shorthand typist.

There is thus a minefield of misunderstandings awaiting the casual user.

Furthermore, the subtle nuances of pleasure or anger or frustration that we can convey by alternative minor choice of wording are not easy to reproduce in a phrase-book style of translation.

Conclusions

For smallish business users who cannot afford to use a professional translator, this package should enable very acceptable letters to be written with a minimum of inguistic knowledge.

At a cost of £150 you receive four booklets, which, bearing in mind their expert nature, are probably worth £15 each, and two discs with a copy of some of the material from the books in text files. These, too, I would give a value of about £15 each.

From this you may discover that I regard the package as somewhat overpriced, but as is often the case, the whole is probably more valuable than its parts.

The literature promoting the package does not make it very clear that Applewriter is a prerequisite for the use of the package. Lack of this item of software would mean that its cost, about £80, has to be added to the overall cost.

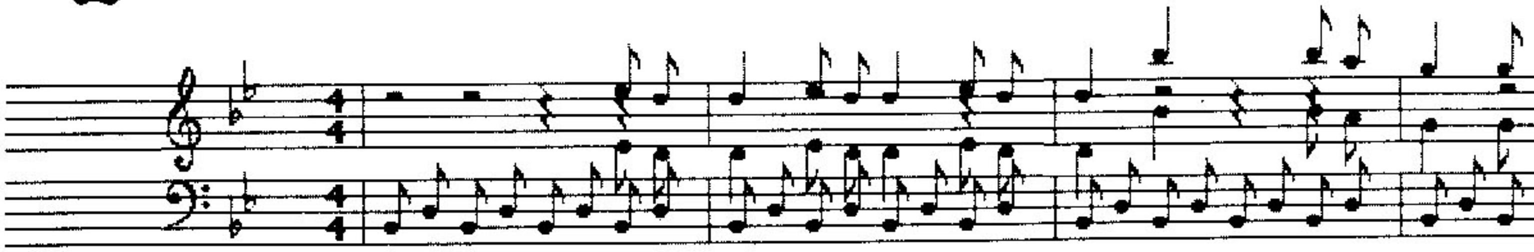
It would have been useful to have the English sentences on disc as well, so that having written a letter in French it would be easy to produce the English version to file away with the French copy for future reference. These are available separately at extra cost.

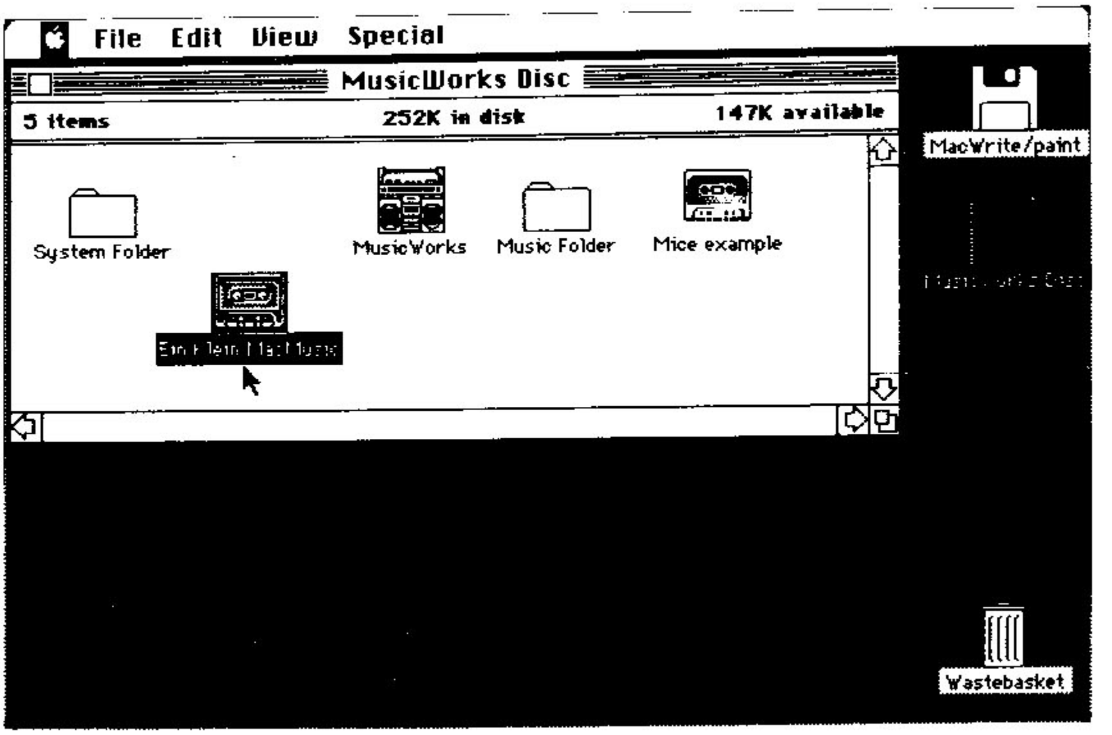
Tick Tack used to be distributed by Longmans but has been "taken back" by Primrose in order to enable end user support. A telephone number is available for any queries you may have. It's a pity more software houses don't adopt this approach.

Versions are available for Apple II+, IIe or III, although only a restricted set of languages are available for the Apple III version.

Kevin Farrell

Product: *Tick-Tack*
Price: *£150*
Distributor: *Primrose Publishing, 11 Church Street, Thriplow, Cambridge SG8 7RE. (Tel: 0763 82512)*





...or how BILL HILL created a Rhapsody in Black and White

JUST picture the scene, in an 18th Century computer store . . . "Ach, so, young Master Mozart – you want to write Eine Kleine MacMusik? I have here joost the piece of software for you – and you won't need to be no child prodigy to use it at the age of seven, neither".

The Macintosh, among all its other goodies, held out from the very first the promise of being an incredible musical instrument – the integral sound-chip, with four separate voices, only awaited the right software to unlock it.

You can do this using Basic of course, as anyone who's tried the demonstration program on Version 2 of Microsoft Basic will no doubt tell you (Tell you? You'll be lucky to get in a word edgeways!)

However, that still leaves the Mac only able to be used as an instrument for anyone with the time and inclination to put together the right program listing.

Not a lot of use for those of us who can't be bothered trying to reduce our crotchets into data statements. But a few packages have appeared over the past few months which promise to overcome that hurdle.

One of these is "MusicWorks". It's a US package, written by Macromind and distributed in this country by Softsel and P&P.

MusicWorks will allow you to compose music for four separate "voices" or instruments.

It's just like a musician's version of a word processor (a music processor?) which is a cross between MacWrite and MacPaint, and just as easy to use.

Not only can you compose music complete with four-part harmonies in a selection of different instrument sounds and tones, but you can play it back, edit and rewrite it, alter tempos, key signatures, move, cut and paste sections – and, at the end of it all, print out proper sheet music with both bass and treble staves.

The possibilities are just about endless. What Mozart could have done with it just doesn't bear thinking about . . .

On booting up, you see the standard Macintosh desktop with a cassette-recorder icon entitled MusicWorks.

There's also a Music Folder. When you open this, there are three other folders inside, entitled Classical, Originals and Simple Stuff.

These folders are crammed full of already-recorded examples, which appear as cassette-tape icons.

There's Bach, Brahms, Liszt, Mozart, Beethoven – a couple of dozen different examples, which you can play to let you hear what's possible.

If you want to write your own new music, you simply choose New from the File Menu, and you are presented with three windows.

There are blank Treble and Bass staves, called Untitled Staff; a second window called Untitled Overview; and a "dialog" window with Play, Stop and Repeat Buttons, scroll bars which allow you to set the overall speed and volume of your music, and four buttons A, B, C and D which tell you which voices are active – you can choose any combination from one to all four.

In the Untitled Staff window is a "paintbox" of musical notes and notations. There are crotchets, quavers, semiquavers, pauses – the whole armoury of musical notation – as well as more familiar Mactools like the eraser and line cursor.

There are also another four buttons entitled A, B, C, D, and A-D, corresponding to the four voices again.

If that all sounds a bit complicated, it's not. But the easiest way to explain is just to go through the steps you follow to write your own music.

First, you choose the musical Key and Tempo by opening windows, using the pull-down menus at the top.

Next, you select instrument A, and you can choose from either of two synthesisers (another window lets you draw your own Attack-Sustain-Decay-Release envelope for them) or trumpet, piano, kazoo, flute or

The possibilities are endless. What Mozart could have done with it doesn't bear thinking about . . .

chimes.

Once this is done, to write the music you simply pick up notes of the required length from the paintbox, using the mouse in standard Mac fashion, and paste them in place in either the Bass or Treble staff (or above, below, or anywhere in between!)

This is where it all gets very clever.

MusicWorks will not let you paste more notes into a bar of music than the time signature allows.

If you attempt to paste in a longer note, it automatically breaks the long note into two shorter ones and ties them together across the bar division.

If you end a bar with too few notes for the time-signature, it will paste in the appropriate pauses.

If you've written a phrase – or even a long section – which you want to repeat, then you can use the cursor to Copy it into the Clipboard, and Paste as many times as you like.

You can listen to it played back, and if you want to change any notes you just replace them with new ones from the paintbox.

You can experiment by changing instruments, and there is a selection of about 12 different qualities of tone for each instrument. The number of possible combinations is staggering.

But of course, that's not all.

You can then pick a second instrument, a third, a fourth, building up harmonies, counterpoints, bass parts, and so on, one at a time.

Selecting A-D plays the whole thing in unison.

Or you can play, say, just two parts together, to hear how they fit in with each other before adding more.

If you want to change one part and keep the rest, you select that particular voice. Notes in that voice remain in black-and-white, while the remaining parts fade to grey, and any actions you take affect only the voice you have active.

Now, while you've been merrily writing notes into the blank staff, strange things have been happening elsewhere.

The Overview window has been gradually building up a schematic of your music, for example. And the staff has been scrolling along the whole time as you lay in more notes. You can scroll backward and forward through it at will.

If you add notes in the middle of a section, MusicWorks automatically shifts everything else along for you – adding ties or breaking up notes automatically where necessary.

You can play your piece back slower than normal, too, to allow you to pick out and alter any rough spots.

As a musician of long-standing who has always had a blank spot when it came to writing music (I play a host of instruments ranging from guitar and Celtic harp to saxophone and Indian sitar – none of them all that well, I hasten to add), I found the package amazing. At last a tool which took away all the hassle!

It's not as limited as you might expect, either. You ain't heard nothing till you've heard a Mac zapping out Mozart's Symphony #40, or Beethoven's Ninth.

The best thing of all is that your music prints out professionally (and very quickly) on the Imagewriter.

For anyone who has ever had the urge to write music, a package like this has got to be worthwhile.

But just imagine what kind of tool the Mac, running MusicWorks, would be for the professional composer, arranger or songwriter.

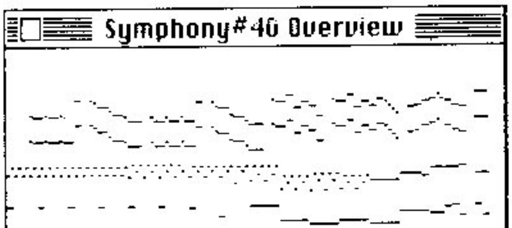
If I were working with music professionally, I wouldn't dream, now, of writing music any other way. It's like the difference between typing copy the old-fashioned way, and using a word processor.

In the former case, you're reluctant to change it once you've typed it up. In the latter, you come to regard everything you write as fluid, to be swirled around with impunity until you're completely satisfied with it.

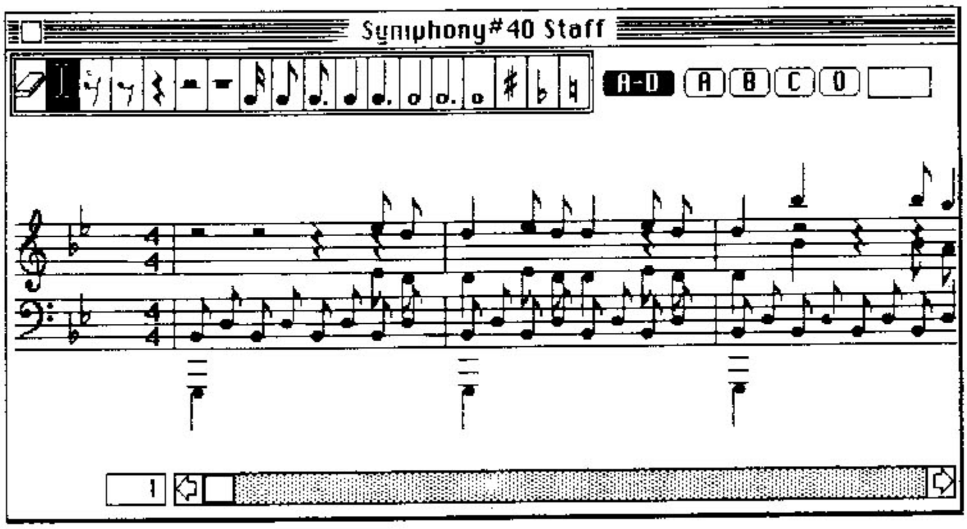
Music Works fits in so beautifully with the Macintosh philosophy that you shouldn't need to become a computer buff to use it for your own particular purpose – you should just be able to plug in, boot up and start work.

The one drawback I experienced with the package was with the keyboard window, which you can play with the mouse. Unfortunately, I couldn't get it to sound quite right.

But that's perhaps a little nitpicking, since you can get along perfectly well without it – and, at £69.95 MusicWorks really is very good value for money when compared with some of the other packages on the market or on the way.



Left: Schematic showing the different voices
Below: Part of the completed score



Join the communications revolution

We've found the missing link!

By DEREK MEAKIN
Managing Editor
Apple User

MANY months ago we forecast that 1985 would be the Year of Communications. And so it is turning out.

Sales of modems are soaring. More users are joining Prestel in their thousands. Bulletin boards are springing up all over the country.

But until now one thing has been missing.

What has been desperately needed was a national database for microcomputers that would combine a low-cost electronic mail service with a giant bulletin board you could access first time every time, that would allow you to chat freely with other users, that would have no restriction on the length of material you could send or receive – and would also enable you to send telex and telemessages from your own micro.

All this, and much much more, you can do with MicroLink. For this one comprehensive service is the missing link in Britain's communications network.

And the carrier through which all MicroLink's facilities are being offered is the most respected name of all – Telecom Gold.

Until now Telecom Gold has been the preserve of big business – used by the giants of industry, commerce and finance to link their operations in Britain with the rest of the world. And Telecom Gold's charges reflected its elevated status.

That has now changed. Thanks to MicroLink, every Apple user can afford to key into Telecom Gold and make use of its growing range of services. For less than the cost of a first class stamp you can have a message the size of this page delivered to one destination or hundreds . . . all in a fraction of a second. And that's just one example of the time and money you'll be able to save with MicroLink.

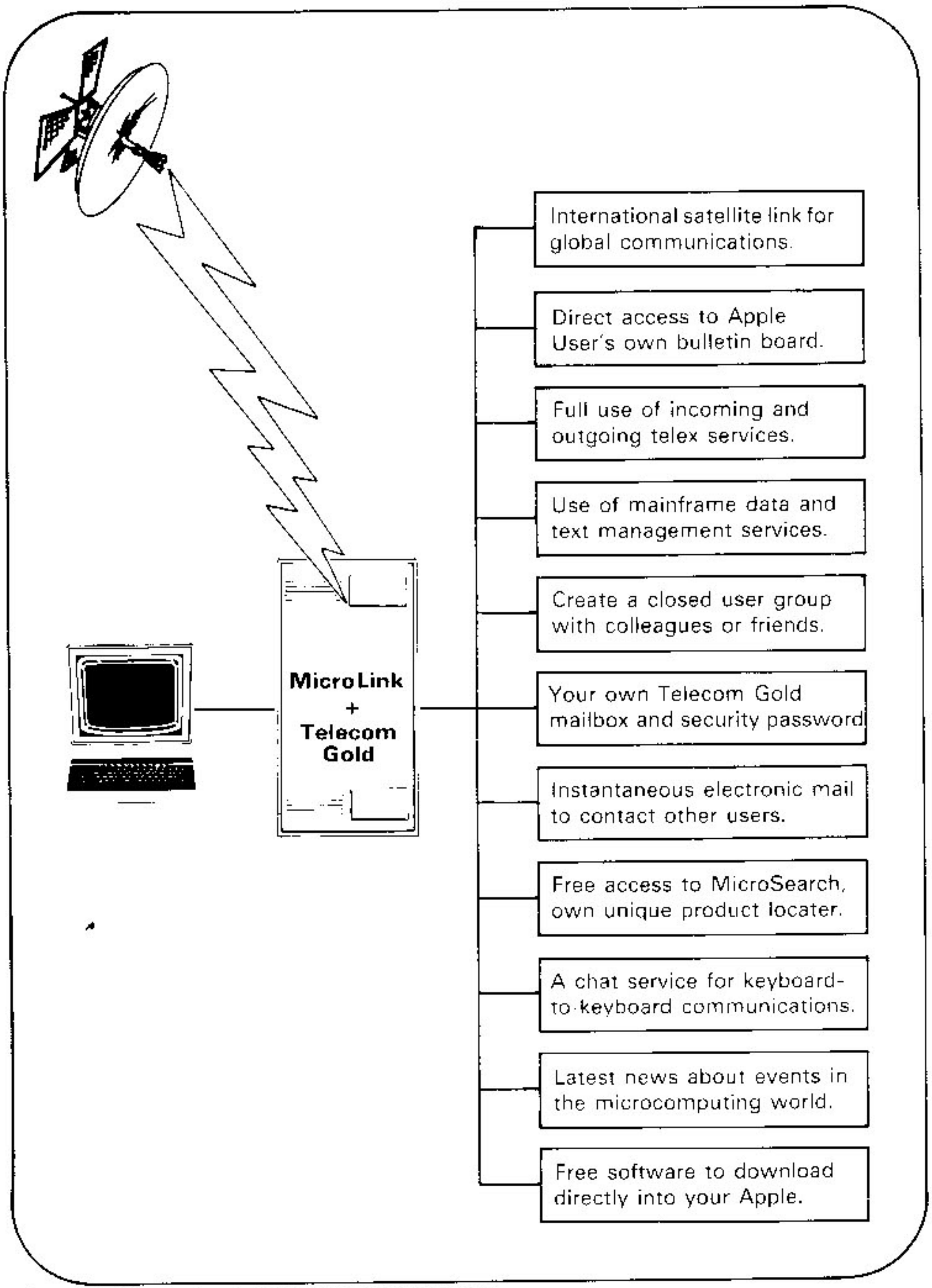
All of us at *Apple User* are very excited about the enthusiastic reception the new service has been given by our readers – and about its unlimited potential for future development.

Come and join us – and explore with us the whole new world that is being opened up by MicroLink.

invites you to join . . .

in association with

TELECOM GOLD*



- International satellite link for global communications.
- Direct access to Apple User's own bulletin board.
- Full use of incoming and outgoing telex services.
- Use of mainframe data and text management services.
- Create a closed user group with colleagues or friends.
- Your own Telecom Gold mailbox and security password
- Instantaneous electronic mail to contact other users.
- Free access to MicroSearch, own unique product locater.
- A chat service for keyboard-to-keyboard communications.
- Latest news about events in the microcomputing world.
- Free software to download directly into your Apple.

MicroLink is this year's most exciting – as well as most ambitious – development in the rapidly-expanding world of telecomputing.

For the first time, it combines the enthusiasm of many thousands of computer users with the power and versatility of Britain's national database, Telecom Gold.

The result is an international communications link that is your passport to new realms waiting to be explored, new experiences to be shared with kindred spirits who enjoy telecomputing just as much as you do yourself.

Communicating the MicroLink way is ultra-fast – and much cheaper than you might expect. Wherever you live, you get direct access to the Telecom Gold computer at local call rates.

With your own electronic mailbox you can send a message to one destination – or to 500! – for less than you would pay for a first class stamp.

You can send and receive telex messages worldwide, or have a two-way chat with other users in real time.

And the cost of using MicroLink? Just £3 a month. Plus small additional access charges as detailed overleaf.

Join MicroLink now – and let you and your Apple be in the forefront of the new revolution in communications!

* *Telecom Gold is a trademark of British Telecommunications plc.*

GRAPHICS

By PETER GORRY

THIS month I shall be reviewing three new books devoted to graphics on the Apple II/IIe – "The 3-D Animated Apple" by Phil Cohen, "Art and Graphics on the II/IIe" by W.H. DeWitt, and "Applied Apple Graphics" by Pip Forer.

They have rather different audiences in mind and provide something for everyone – from the complete novice to the dedicated hacker. In all cases you can obtain a disc containing the demonstration programs – but only Pip Forer's book comes with the disc automatically.

In the other cases the discs must be bought separately to avoid the tedium of entering the programs yourself. Unfortunately the review copies were not supplied with the discs and your humble servant has worn his fingers to the bone in an effort to check the reliability and quality of the software.

In reality checking all the programs would have involved typing in more than 80 programs, so I've picked ones more or less at random (less than random for the very long ones!).

THE first book I tackled was the "3-D Animated Apple". This is a medium-sized paperback of about 200 pages but quite cheap at £4.95.

The book is a "curate's egg" – good in parts – but somewhat idiosyncratic in its subject matter.

Since all the programs are in Basic (no machine code routines are used) "animated" is only barely distinguishable from "still" in some cases – but more of that later.

The early part of the book is dedicated to using and storing shape tables.

The demonstration programs are short, well thought out, and very quickly lead up to a little Lunar Landing game.

The game is reasonable in that it checks for one flying off the side of the screen or crashing into the ground. However, it didn't reckon on someone with so little ability as to fly off the top of the screen (Cliff McKnight eat your heart out).

This section ends with two programs to create and "store" shape tables.

Splitting the task into two programs, one that creates a *single* shape, and one that combines shapes into a table is a rather curious way of going about it, and far from ideal.

The programs do, however, provide for storage on disc or tape – the latter is often ignored in commercial utilities.

The method used for saving the completed shape table from COMBINE SHAPES is most bizarre. The program exits with a screen displaying the POKES and DATA statements needed to recreate the table.

These are then turned into a program by copying the lines from the screen using the right arrow and Return keys.

Long tables would unfortunately scroll off the top of the screen before you had a chance to get them!

No real table editing is possible using these programs (deleting shapes, reordering tables etc.) so the facilities offered are rather primitive.

I'm afraid this characterises most of the programs – they work but they don't offer very much.

Having mastered shape tables they are then largely ignored for the rest of the book.

The middle section of the book addresses itself to two-dimensional graphics including simple data plotting, pattern-drawing and figure manipulation.

The first two are rudimentary and rather ordinary, but the last topic is more unusual.

Modern animation studios often use computers to take the drudge out of animation drawing.

In the old days, every frame of a cartoon had to be drawn by hand at great time and expense. However, it is possible to get a computer to do a lot of the laborious parts by getting it to fill in the gaps between well-spaced "hand-drawn" frames.

The computer's task here is to transform one shape gradually into another in a smooth manner. Provided the shape doesn't change too rapidly, this is not difficult and the task of creating animated films is rendered much easier.

Two programs, MAKE FIGURE and MOVE FIGURE, are examples of how this can be done.

As before, the programs turn out to be somewhat limited in reality, but I rather liked this section and I think more should have been made of it.

The animation is not "real time" but rather provides individual frames. I'm sure a video recorder and these programs could be used to create some interesting animation sequences. It's a pity this isn't elaborated upon.

The last third of the book is concerned with three-dimensional drawings.

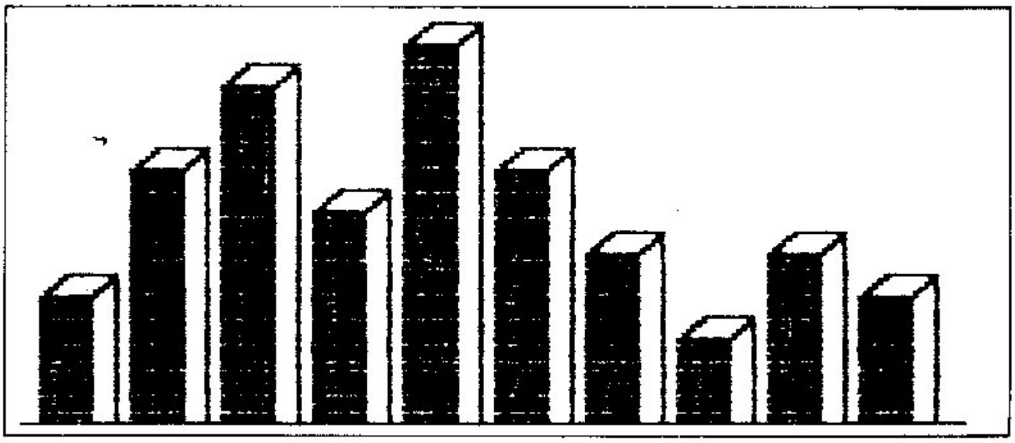
The approach is very similar to the preceding section, although the programs are very much longer.

One program enables you to create, edit and view from various angles and perspectives, a single shape.

These shapes can then be used to provide the "hand-drawn" frames for a 3-D animator in much the same way as outlined above.

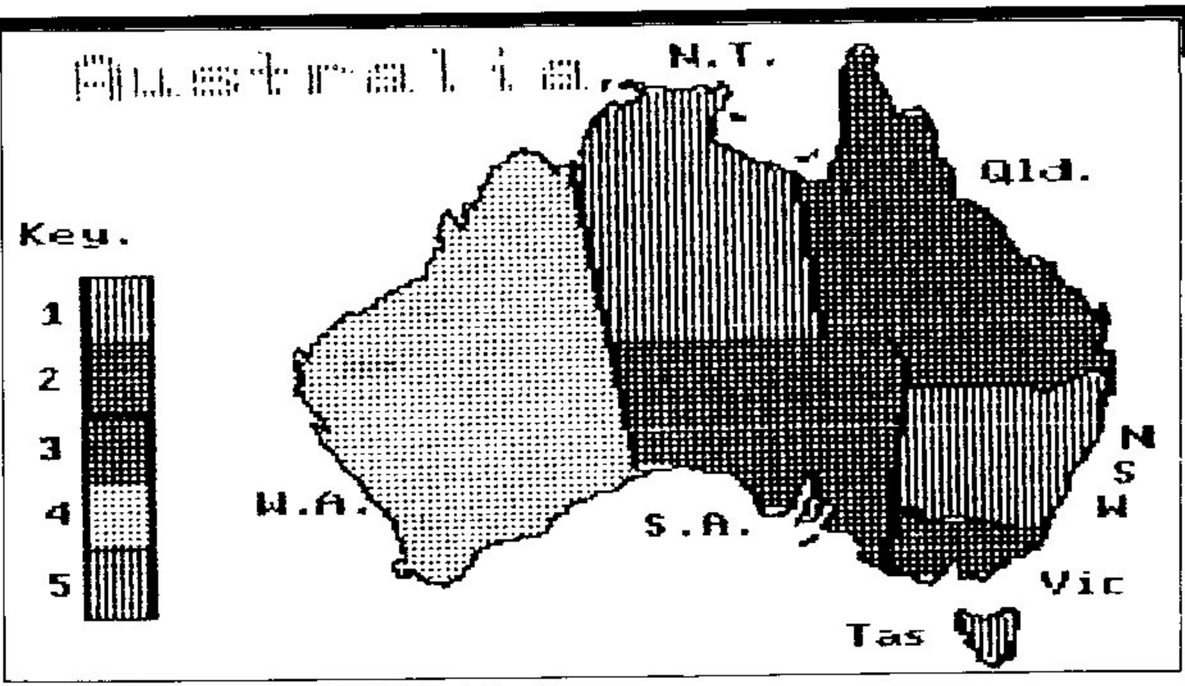
The shapes are of the "wire frame" type and no discussion of hidden line removal is provided.

I'm afraid that at the end of the day I'm not sure what most people would do with the programs on offer here, since they are not fast enough for real time animation nor sophisticated enough for serious applications – at best they might inspire one to greater things.



Solid histograms – from The 3-D Animated Apple

Title: The 3-D Animated Apple
Author: Phil Cohen
Publisher: Interface Publications
Price: £4.95



Population density map from Applied Apple Graphics

"APPLIED Apple Graphics" is a very different sort of book from the other two and is aimed at the more-experienced user, or someone with a specific task to perform and looking for software to do it with.

The book is a mixture of hard information for writing your own graphics programs and discussions of commercial software packages that perform similar tasks.

The book itself is a large-size paperback some 340 pages long.

I found this book to be head and shoulders above the other two, although it would probably be too much for a complete beginner.

It does start with some simple examples of lo-res and hi-res plotting, but you can see its heart isn't in it.

The book itself contains thirteen chapters and six appendices, so I won't give details of each one.

There are numerous example programs of which only a few are listed in full in the text, which is why the disc comes bound into the back of the book.

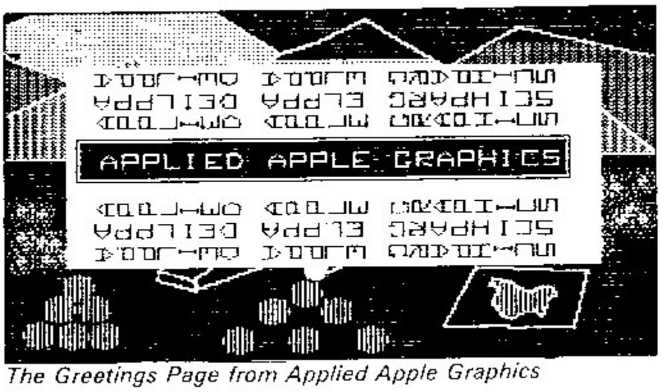
The programs are there as examples only and they are not meant to provide a suite of utility programs.

It seems to me that the book would appeal to someone who has mastered simple graphics programming and who now wants to improve his knowledge and to learn how the experts do it.

It contains much of the information that has taken me several years to pick up from numerous sources and presents it in a lucid and interesting manner.

I certainly wish this book had been available a few years ago – it would have made my life so much more simple.

In fact the book contains a large bibliography, so that the original sources can be consulted for further detail (including many articles from Windfall, as *Apple User* used to be called).



The Greetings Page from Applied Apple Graphics

The approach is well structured and each topic is dealt with at just the right level for the task at hand.

For instance, the relationship between screen points and memory locations is considered in three separate chapters: 1, 3 and 12.

In Chapter One, the basic idea is introduced so that the reader can understand how micros display graphics by altering values in memory.

In Chapter Three, we learn about the memory and page layouts in the Apple, along with all the POKEs that allow drawing and displaying of various pages.

In Chapter 13 we finally come face-to-face with the (near-ludicrous) relationship between memory addresses and actual screen positions and how the colour masks for individual bytes are constructed.

This last level of detail is really of more interest to the assembly-level programmer – a topic which Pip Forer leaves largely alone.

The book tries to keep to high-level languages, mainly Basic but with forays into Logo, Pilot, Pascal and Forth. The closest we come to machine level is a discussion of variou monitor calls and references t more advanced material in the bibliography.

As I said at the beginning, the book also spends quite a lot o time discussing commercia software and it is this tha makes the book doubly useful

The book does not attempt t give a comprehensive review o all the available packages, but i does give a good feel for wha sort of things are possible an how to review software befor parting with hard-earned cash

The guidelines laid dow here are very perceptive an would quickly steer you clear o bad software.

The book has a stron leaning towards "teaching" i its examples and although th only thing the book tells yo about the author is that he i from New Zealand, I wouldn be surprised if he was a teache – and, from the preponderanc of maps, I would guess o geography.

I was a bit surprised a finding so many maps in th example programs, but they d turn out to be ideal subjects fo displaying the uses of compute graphics in the teachin process.

In fact the nearest th example programs come t providing a useful set of utilities comes from this section.

The section on shape tables is well done and there is a good discussion of commercial shape generators with the sensible advice to buy one if you plan making much use of such tables (he hadn't of course seen the March *Apple User*!).

The tables themselves are then used in numerous examples throughout the book, finishing off with simulation and animation.

The ideas behind "flicker free" animation are discussed, along with the limitations of trying to do it all from Basic.

The easiest way to get improved animation is to use utilities written in machine code – but controlled from Basic – and there is a detailed account of using the animation routines in Penguin's Graphics Magician to achieve this.

Along the way we also find information on memory management, how to create hybrid colours on the hi-res screen, 3-D drawing and graphic database design, add-on hardware, hi-res text and screen painting utilities.

The only area which I felt was somewhat neglected is the double resolution hi-res plotting possible on an Apple with an extended 80-column card.

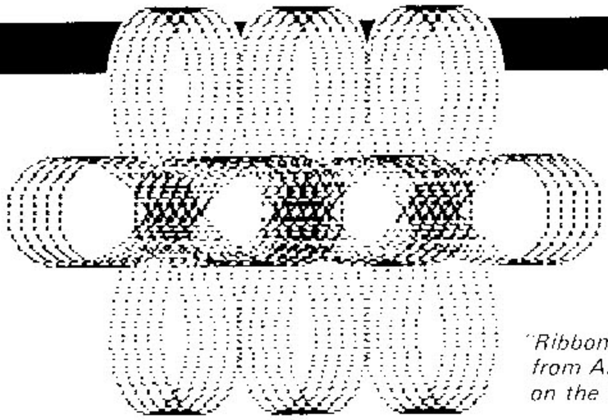
I know that it isn't possible to access this from ordinary Applesoft, but there are now several utilities such as Doublestuff and Cat Graphics that can.

The IIe special features are only mentioned in a two-page appendix, which hardly does it justice.

I can thoroughly recommend this book to anyone who wants to learn more about the graphical capability of the Apple and the software/hardware which enhances this capability.

It isn't a book for reading at one session, but it would act as a trusty guide for some time to come – until the dreaded stage when you decide to withdraw from humanity altogether and take up machine code programming.

Title: Applied Apple Graphics
Author: Pip Forer
Publisher: Prentice/Hall
Price: £19.45



"Ribbons and Bows", from Art and Graphics on the II/IIe

"ART and Graphics on the Apple II/IIe" is a very different sort of book and aimed at the complete newcomer who wants to produce interesting displays with minimum programming effort.

I'm afraid I found the style truly excruciating and many of the programs very mundane (there are some 60-odd in the book) but there are enough good ideas tucked away to make it worthwhile trying most of them.

The book itself is 127 pages long, with four pages of colour photographs in the middle. It has a rather generously spaced layout with large print – reminding me a little of the books for the poor-sighted.

The style is that mixture of enthusiasm and exhortation perfected by holiday camp attendants to bully people into playing games when they'd much rather be doing something else.

I also think it's rather pretentious to call 10 lines of Basic program that produces a pretty pattern "art".

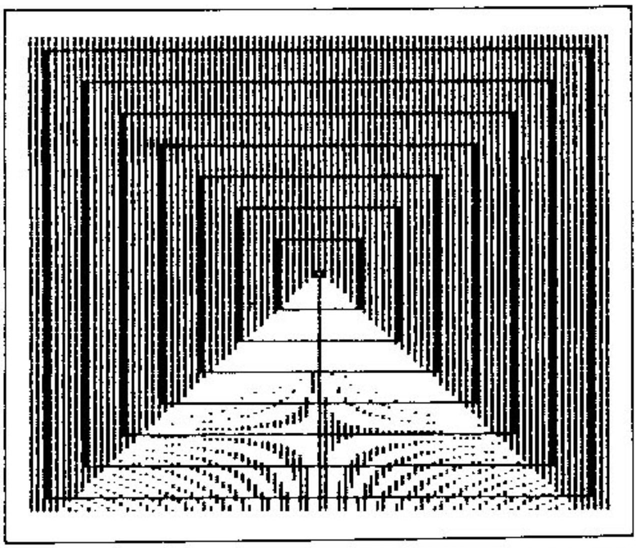
No doubt I'm just being old-fashioned, but I thought art involved starving in garrets, cutting your ears off, a modicum of talent and lots of hard work.

If you can put up with the approach, the book rapidly takes you through lo-res and hi-res plotting with a host of short programs, sometimes producing rather striking displays.

The strength of the book is that you are producing results right from the word "go" in a manner guaranteed not to frighten the novice.

There are also numerous suggestions for changing the programs to get different results.

The book then moves on to using shape tables to produce a variety of different patterns. Great use is made of SCALE, ROT and RND in all of these and the effects can wear a bit thin after a while.



"Corridorically Speaking"

The allusion to famous artists or pictures also seemed somewhat gratuitous – for instance a 20-line program that draws random squares on the screen "an ever-changing screenful of rectilinear patterns" did not to me seem "reminiscent of Reinhardt's Number 17, 1963".

The only problem with the treatment of shape tables in this book is that it doesn't explain how to make your own, so you are stuck with the few shapes provided.

The next sections introduce a certain amount of animation into the pictures and it is here that the idea of "time" being a useful variable in making eye-catching displays is more thoroughly discussed.

In fact it is this element which provides "computer art" with a niche of its own.

The linking of images to music is also encouraged, although the practical advice here is rather limited.

Finally the book ends with advice on how to go about photographing the screen to get permanent images and how to connect up to a video recorder.

Much of this advice is usually difficult to find and it's handy to have it presented in one place like this.

All in all, the book improves as you progress through it, and if you have an interest in computer-generated images and no prior experience this isn't too bad a buy.

Title: Art and Graphics on the Apple II/IIe
Author: William H. DeWitt
Publisher: John Wiley
Price: £12.25

Make it big with Musicworks.™

They had everything going for them. Good looks, determination and a great image.

But one thing stood between the Sydney Saunders Skiffle Duo and success in the music business. Talent.

But blessed with a good ear for a winner, Sydney laid out a few notes for Musicworks. A major development from Hayden Software.

FOR BUDDING HOLLIES EVERYWHERE

Even if you can't read a note, Musicworks enables you to compose music directly on your Apple Macintosh,* and hear the results instantly.

MusicWorks
Create one part melodies or multi-voice symphonies, even if you can't read a note.
For the Apple Macintosh
HAYDEN

You compose on a seven-and-a-half octave grid. Just like the keys on a piano. As you put down your notes you hear them. All in the click of a mouse.

When you've got your latest masterpiece sounding just right, Musicworks transforms it into a musical staff, inserts the right notation and prints it in a professional format which any musician can follow. Even Sydney.

Now he can drum up any tune he likes. Change and edit notes. And then play back the whole melody whenever he wants.

THE FAB FOUR

Another feature that Sydney likes to harp on about is Musicworks' built-in instrument selection.

Use up to four at a time from a total of 10 or listen to each one selectively. Musicworks also comes complete with templates which guide you through playing classical and popular selections.

It's available from your local Softsel dealer. Along with over 2,600 other titles from over 250 publishers.

However there's still one slight hitch in Sydney's rise to stardom. He's been charged under the Trade Descriptions Act over the name of his latest album – "Sydney Saunders Entertains".

Musicworks – A sound investment.

The number one distributor of software. In the world.

Softsel Computer Products Ltd, Softsel House, Syon Gate Way,
Great West Road, Brentford, Middlesex TW8 9DD.

MUSICWORKS IS A TRADEMARK OF MACROMIND INC. *MACINTOSH IS A TRADEMARK OF APPLE COMPUTERS INC.

Grab a handful of light dependent resistors, string up some Christmas-tree lights around your Apple, key in this program by JIM DAWSON, and you'll end up with a coming-out party which will deafen the neighbours.

It may get you evicted, but you will find out in the process that your Apple can make real music in response to light.

SOMEWHERE back in the depths of time, someone invented the Theremin.

For a while it looked as if the Theremin might replace all of the violins, Wurlitzers and counter bassoons the world around. But then that particular new wave of music died a natural death and music buffs resorted to slapping discs recorded with more-or-less normal-sounding instruments on to their turntables.

This, however, was before the advent of the personal computer and Star Wars.

The Theremin (pronounced in three syllables!) was an instrument made up of two copper plates and an enormous black box filled with electronic components of all descriptions combined in such a way as to make up two separate radio frequency oscillators and an audio amplifier.

The RF oscillators were tuned to almost . . . though not quite . . . the same frequency, and the copper plates were merely the externally mounted plates of the RF tuning capacitors.

The musician flailed his arms about wildly somewhere in the vicinity of the copper plates detuning one or the other of the oscillators, and the result was a banshee-like wail which rose and fell in pitch and volume as the capacitance of the plates changed.

The audio output of this apparatus was the direct result of beating the two different radio frequencies (one from each oscillator) together.

Since the advent of the Apple, and now that our ears have become more or less accustomed to a variety of strange noises currently accepted as music, the time has clearly come to replace the Theremin with something more fitting to the 'Eighties.

This article is about getting your Apple to respond musically to stimuli from the outside world in the form of light.

The purchase of a CdS light dependent resistor (LDR) is the starting point in getting your micro to respond to light.

The theory behind LDRs is briefly this . . . uhm, you can skip this part if you want to. If radiation falls on a semiconductor, its conductivity increases. Covalent bonds are ionized and electron holes in excess of those generated thermally are created.

The increase in the number of available current carriers results in the conductivity increasing or, if you wish, in the resistance decreasing.

Electron hole pairs are created when the material in the resistor (semiconductor) is hit by photons – light particles – through a process called intrinsic excitation.

While other types of excitation are possible, this is the one of interest here!

The dark resistance of the LDRs you can buy will vary from several megohms to only a few ohms when the resistors are exposed to room lighting.

The one I liked had a dark resistance of one megohm and a light resistance of 200 ohms but you can experiment with any of them since they all work. Each, however, will respond slightly differently to the same amount of light.

A typical LDR is available from Tandy (cat. no. 276-116) or Maplins (cat. no. ORP12 or RPY58A) or Electrovalue (cat. no. MKY7C38E).

Lit with a 60-watt bulb three feet away, the light resistance of the LDRs I tried varied from 8 kilohms to 200 ohms.

In short, the problem involved in this project entails putting the resistor you select somewhere the light can get at it and then connecting it to the Apple.

A long time ago I concluded that Steve Wozniak had thought that users of his brainchild would wish to do no more than plug in a set of game paddles and leave them installed in the game I/O port for all time while they played Little Brick Out.

Since then times have changed.

The Apple IIe (and later models') game I/O port is easily accessible and you will not neec the breakout box described in this article.

If you own a II+ or an earlier Apple, this little box is worth its weight in US dollars!

If you don't need the breakout box, you can skip the next few paragraphs and get down to some programming.

If, however, you have only the sixteen pin game plug I/O to play with, read on . . .

This breakout box does nothing more than take the game plug I/O and place it outside your Apple where you can get at it easily.

Since the two DIP sockets are wired in parallel, Pin 1 in the free socket on the breakout box corresponds with Pin 1 in the game I/O inside the Apple, Pin 2 with Pin 2, and so on.

For this application we need only two of the 16 wires in the ribbon cable and two of the 16 pins, Pins 10 and 1, but you may as well wire all of the pins in parallel while you are at it.

Other applications using the game I/O may be coming up later on.

Figure I shows the pinout arrangement used in the Apple game plug. The breakout box calls for a 16 pin jumper cable, a small plastic or metal utility box, some perfboard, and a couple of 16 pin DIP sockets. A parts list is given at the end of this article should you wish to shop at a Tandy store nearby.

The Tandy jumper cable comes complete with two male 16 pin DIP plugs. You could buy some 16 conductor ribbon cable and a couple of 16 pin DIP headers and make up your own jumper, but I doubt if the pennies saved are worth the bother.

Its length is only 18in so if there should be a need for greater length you will have to make your own lead. Note that actually only 14 wires are needed as pins 9 and 16 are not used.

For slightly more money, all or part of Tandy's memory IC PC board (cat. no. 276-184) can be used to make up the board. This will save you a lot of soldering fiddly wires.

Even more money will buy a zero insertion force (ZIF) socket which can replace one of the

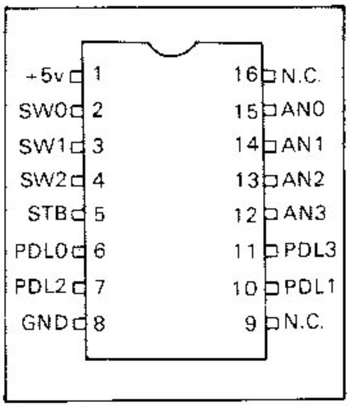
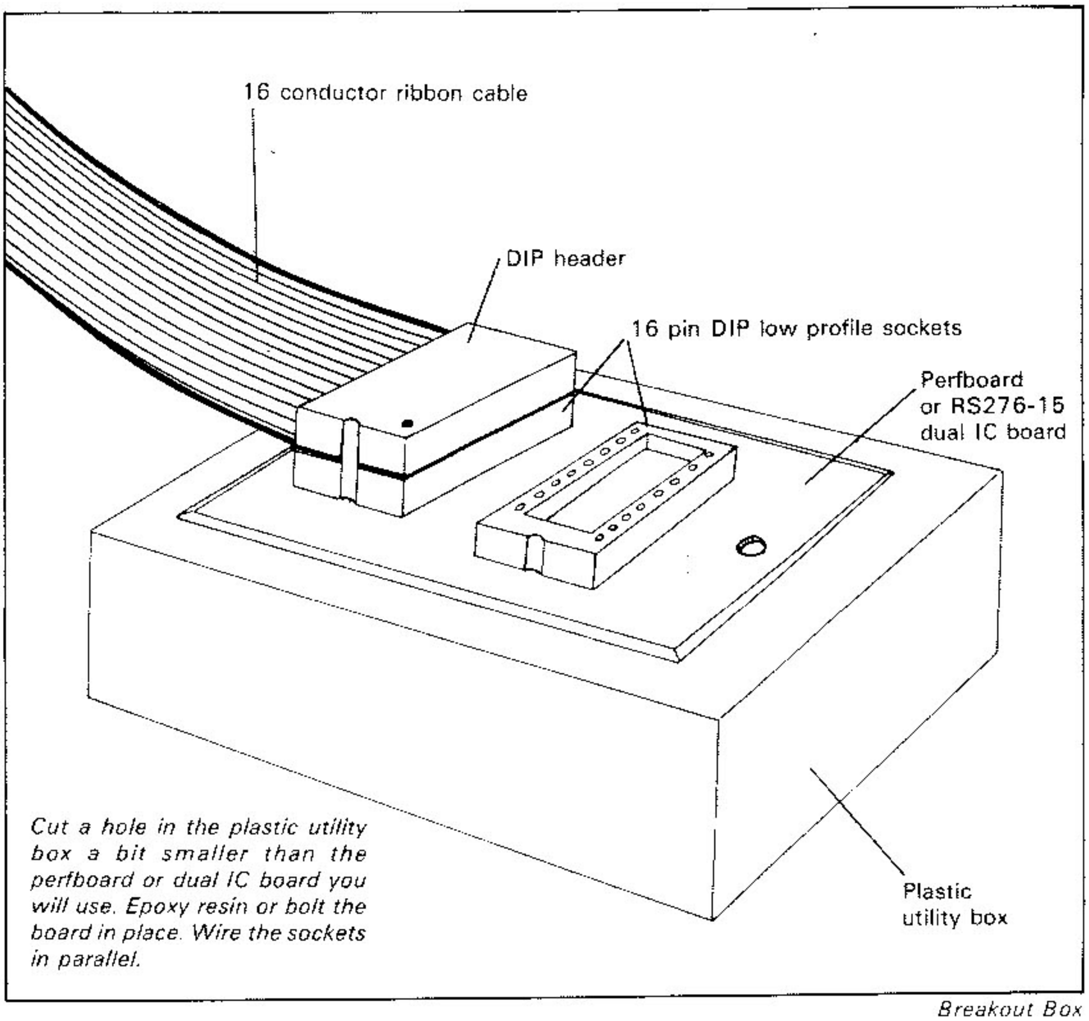


Figure I



Cut a hole in the plastic utility box a bit smaller than the perfboard or dual IC board you will use. Epoxy resin or bolt the board in place. Wire the sockets in parallel.

Breakout Box

DIP sockets and make changes between joysticks, paddles, LDRs and dongles much more easy.

The object of the exercise at this point is to connect the LDR you have selected to the game I/O in place of one of the game paddles.

The program refers to PDL(1), but any of the four available could have been used equally well.

In this instance, the resistor is placed between Pin 1 (+5 volts) and Pin 10 (PDL1).

What with a horde of cards in place in the available slots in your Apple, this is fairly much easier to do outside the machine than inside. Hence the breakout box.

Apples do not, in general, make agreeable noises without some effort from humans. Plucking the speaker by PEEKing 16336 can produce a rasping note of sorts at a frequency of about 70 Hz in Applesoft Basic or 250 Hz in Integer Basic. To produce said rasp, one might type:

10 X = PEEK (-16336): GOTO 10

The limitation on frequency is imposed by the lack of speed of the Basic languages and neither Applesoft nor Integer Basic can operate at sufficiently high speed to produce a musical tone. Only machine language can operate at the speed required.

You can, however, POKE the values required for a short machine language program from Basic and call the speaker plucking routine at SC030 from the Basic driving program.

In this program, memory locations 864 to 891 ($360 to S37B) which are normally free are used to hold the values required for the machine language subroutine.

Lines 10 to 40 read the data contained in Lines 4000 to 4020 and then POKE memory locations 864 to 891 with the required values.

Page One of the two high resolution pages is cleared in Line 41 to provide a hi res display of the notes to be played as you hear them and white is chosen as the colour.

Lines 50 and 60 can be interchanged. Leaving them as they are listed, a bargraph representation of the notes played is output to the screen.

Interchanging them results in a normal graph with point to point plotting of the notes produced.

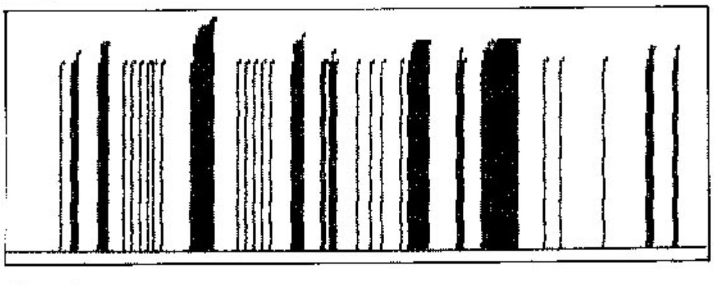
In Line 105 the game controller you wish to use is selected. Paddle 1 is used in this listing, but any of the four could have been used instead. The program reads the value at location PDL(1) and uses this value to derive P, the name given to the variable actually used in producing a note.

The value at PDL(1) normally ranges from 0 to 255 with the game paddles in place.

If you are interested in things technical, when Hex location $C070 is referenced by either a read or write operation, bit seven at four memory locations – corresponding to the four paddle pins – is set to a logic one.

At this instant, four capacitors in the Apple begin to charge from the positive five volt pin of the game I/O through four resistors, the game control potentiometers located normally in your game control box.

The value of these resistances is, of course, variable, and the length of time the capacitors take to reach two thirds of the five volt source voltage depends on the value of



Sample bar-graph output

the resistances in question.

When a given capacitor reaches two thirds full charge, the value of bit seven of the appropriate location in memory will flip from a logic one to a logic zero.

Locations $C064 to $C067, corresponding to pins GC0 to GC3 on the game I/O port, are, in fact, the locations used to report the status of the four capacitors in question.

In Line 110 of the program the horizontal position, X, is defined as being X + 1.

Interesting results are obtained by changing the increment of the horizontal position from 1 to L and to change the value of L in the loop via the keyboard. To do this, insert the line:

238 L = PEEK(–16384)–192: IF L < 1 THEN L = 1

Normally the keyboard letters A through E provide more than enough range for L, but there is nothing to stop you from hitting other keys for longer notes.

What about the notes themselves? A note, when all is said and done, is an audio frequency sustained for a given length of time. The duration of the note has already been discussed in the derivation of L above.

It is the value POKEd into location 865 which determines the frequency of the notes the program will play.

If the notes are to correspond to the western chromatic scale, values for P corresponding to 24, G in the top line of the bass clef, to 253 or B flat three octaves above middle C, are required.

The values for P must be specific values, however, if we are to have discrete notes and not a wail! The analog value obtained for P in Line 105 must be modified.

Since some 42 notes are available to us in the range selected, the original range of P – zero to 255 – is modified in Line 210.

We now have a range of P from one to 42.

The ON P GOTO statement in Line 1000 gives us the new value for P in one of the 42 steps required to produce a specific note in the western scale.

P is POKEd into memory location 865 in Line 240 and the note-producing subroutine is CALLed in Line 250.

One problem which this program does not attempt to deal with is the unfortunate relationship which exists between the frequency of a given note and the time during which it will sound.

Regardless of the value we select for L, the duration of a note, the Apple insists on making notes at the extreme ends of the scale shorter than those in the middle ranges.

Some compensation has, in fact, been introduced in the machine language routine at 866, but any further improvement would call for an algorithm which would be much more complex than the one used.

Meanwhile, there is the problem of the screen representation of the graphic form of our music.

Some of the values for P reported from the GOSUB at Line 1000 would bring things to a crashing halt if we did not limit them for the screen.

P is to be plotted on the Y axis and must, therefore, not exceed the limitations of the Apple hi-res screen.

A maximum value for P for plotting purposes is obtained in Line 257 by dividing P by two, and, to have the graph plot logically – low notes at the bottom and high notes at the top – the value is further modified in the same line using the algorithm P = INT (ABS (256-P) /2.

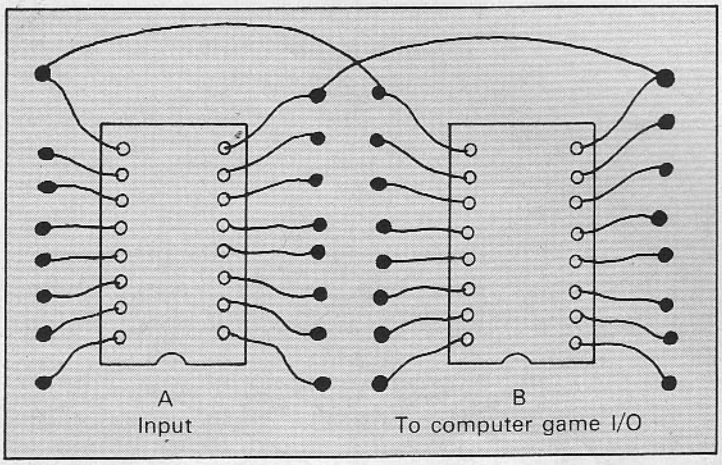


Figure II: Bottom view of Tandy 276-159, dual IC board showing parallel wiring of sockets. Note only two wires have been shown above for the sake of clarity. ALL pins should be connected in a similar way. Pin 1 of A to pin 1 of B, pin 2 of A to pin 2 of B, etc.

A couple of fine stiff wires placed in the appropriate holes of the socket in your breakout box will allow you to fasten a pair of wires (with alligator clips?) leading to the LDR.

I used speaker wire soldered at one end to the LDR and equipped with clips at the other end.

The clipped-off leads from a small ceramic disc capacitor are ideally suited to the job of fitting into the DIP socket holes.

Leaving the capacitor intact actually helps matters by contributing some rigidity to the structure.

The LDR could be placed in an enclosure of sorts which sharply limits the amount of ambient light that can reach it.

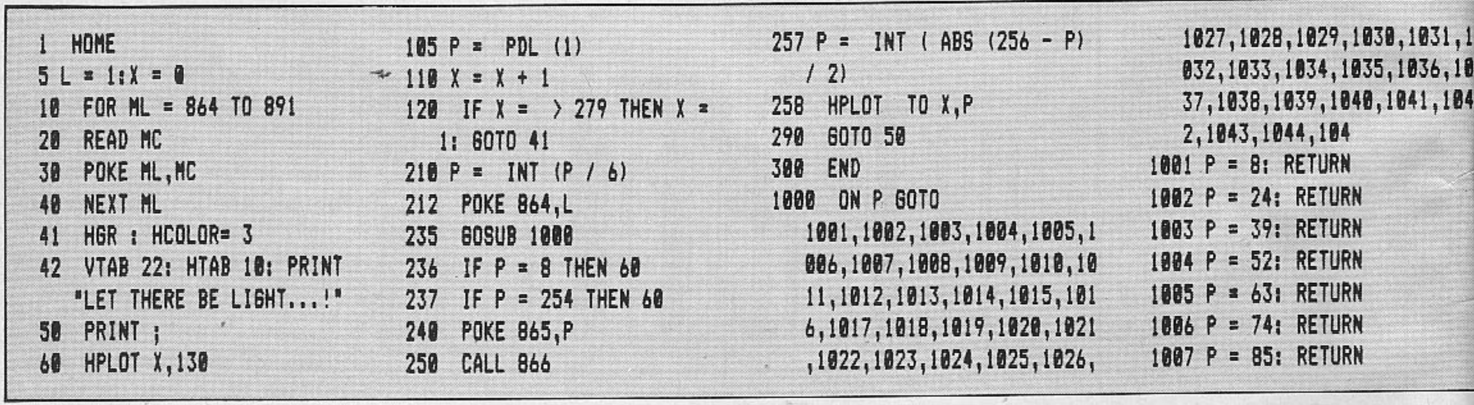
I used a plastic box that once contained 35mm slides (these CdS LDRs are incredibly sensitive to light!) and even then it was necessary to tape much of the box with masking tape to cut down on the light getting through.

If too much or too little light hits the LDR, Lines 236 and 237 send the program back to Line 60 and NO note is produced at the extreme ends of the scale.

This provides a method of obtaining rest notes, periods of silence so essential to music of any kind.

A further rest note could be intercalated at Line 238 by PEEKing –16287, the location for Pushbutton 0.

If the button is depressed, that is, if Pin 2 of the game I/O is



```
1  HOME
5 L = 1:X = 0
10  FOR ML = 864 TO 891
20  READ MC
30  POKE ML,MC
40  NEXT ML
41  HGR : HCOLOR= 3
42  VTAB 22: HTAB 10: PRINT
   "LET THERE BE LIGHT...!"
50  PRINT ;
60  HPLOT X,130
105 P =  PDL (1)
110 X = X + 1
120  IF X =  > 279 THEN X =
   1: GOTO 41
210 P =  INT (P / 6)
212  POKE 864,L
235  GOSUB 1000
236  IF P = 8 THEN 60
237  IF P = 254 THEN 60
240  POKE 865,P
250  CALL 866
257 P =  INT ( ABS (256 - P)
   / 2)
258  HPLOT  TO X,P
290  GOTO 50
300  END
1000  ON P GOTO
   1001,1002,1003,1004,1005,1
   006,1007,1008,1009,1010,10
   11,1012,1013,1014,1015,101
   6,1017,1018,1019,1020,1021
   ,1022,1023,1024,1025,1026,
   1027,1028,1029,1030,1031,1
   032,1033,1034,1035,1036,10
   37,1038,1039,1040,1041,104
   2,1043,1044,104
1001 P = 8: RETURN
1002 P = 24: RETURN
1003 P = 39: RETURN
1004 P = 52: RETURN
1005 P = 63: RETURN
1006 P = 74: RETURN
1007 P = 85: RETURN
```

connected to ground at Pin 8, the value at this location will be greater than 127.

A line such as:

238 K = PEEK(–16287):
IF K > 127 THEN 60

would do the job nicely.

Once you are familiar with what a single LDR can do in terms of producing notes, you might want to try including a second one. The limit is, of course, four.

A second one would be referenced in the program with PO = PDL(0), a third with P3 = PDL(3), etc.

You might arrange your algorithm in such a way that PO will produce high notes when light hits the resistor attached to the PDL(0) input at Pin 6.

Putting both resistors in the same box (one between Pins 10 and 1, the other between Pins 6 and 1) could then produce an alternating set of notes, one high and one low, with a given amount of light.

With four LDRs connected, the whole family could play at once. The resulting cacaphony defies imagination.

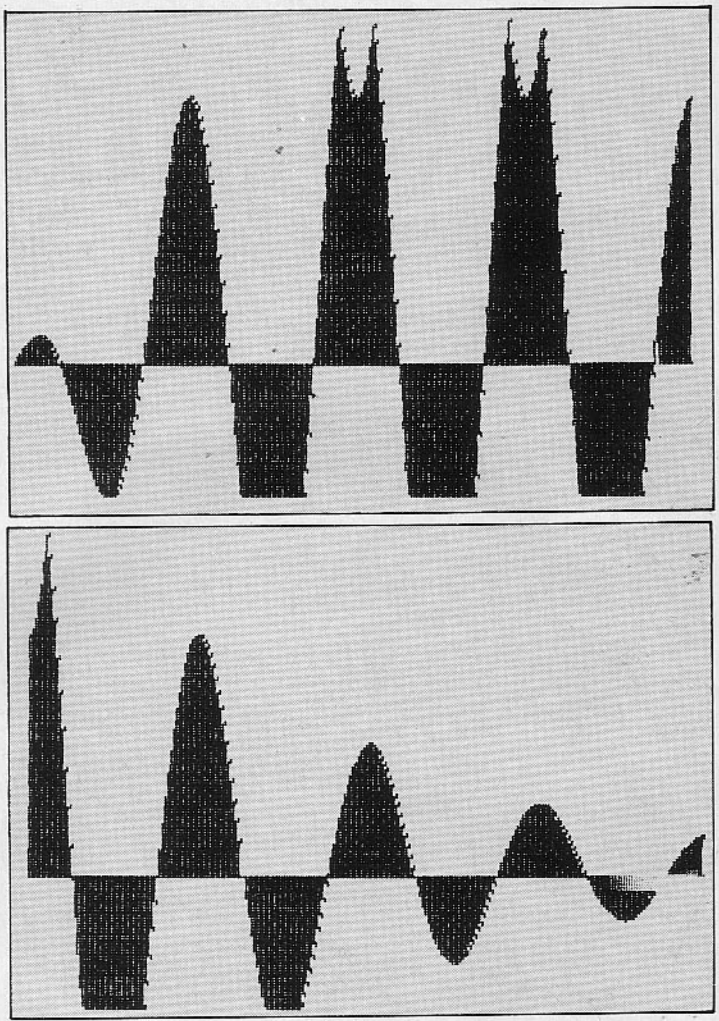
Inserting a Ctrl-G in the program can produce some interesting results.

Try, for instance, Line 260. 260 PRINT " " with, to be sure, a Ctrl-G in the quotation marks. Two Ctrl-Gs will produce a waltz rhythm, three . . . ?

If the pitch of the "beep" isn't pleasing, substitute a fixed value for P in place of the Ctrl-G and send the program off to a subroutine to produce the sound you want.

Values of 24, 32 and 59 fit in nicely with this idea.

There are other applications for this technology, of course.



Sample graphical output

While it's an interesting exercise in its own right, this program lends itself to modification in a number of ways which will allow your Apple to acknowledge the real world in a number of ways.

Other kinds of sensors can be used to return varying resistance values.

The LDR used in this article can be used to make a darkroom exposure meter replete with software to compensate for varying grades of paper and the temperature and type of developer being used.

If you want to be very slick, temperature dependent resistors are available and the readings from the LDR and TDR can be integrated into your algorithm.

There is nothing to stop you from turning your Apple into a £1,000 thermometer, for instance, or a fire alarm.

Serious laboratory uses come to mind at once in applications involving temperature control of procedures.

You can have fun, too, in substituting mathematical formulae to produce values for P. Instead of P = PDL(0), try substituting something like:

FOR I = 1 TO 90: P = ABS (INT (TAN (I) * 100))

For that matter, there is nothing to stop you from "playing" the mortgage table for your house or the oestrus cycles of your pet hamsters in D minor.

Like somebody's unfinished symphony, this is an unfinished program. Where it goes from here is up to you and your imagination.

Parts list for Breakout Box

276-159: Dual IC board, $1\frac{3}{4} \times 3\frac{3}{8}$ (or IC perfboard such as 276-1394 depending on construction method used.)

270-230: Experimenter box (plastic, metal cover, about $3\frac{1}{4} \times 2$ inches. Any box will do.)

276-1976: 16 pin DIP jumper.

276-1999: 16 pin low profile DIP sockets or equivalent sockets from another supplier.

Jim Dawson is a Canadian journalist, and this article is published in collaboration with Computing Now! magazine.

```
1008 P = 95: RETURN
1009 P = 105: RETURN
1010 P = 115: RETURN
1011 P = 125: RETURN
1012 P = 134: RETURN
1013 P = 143: RETURN
1014 P = 151: RETURN
1015 P = 159: RETURN
1016 P = 166: RETURN
1017 P = 172: RETURN
1018 P = 177: RETURN
1019 P = 182: RETURN
1020 P = 187: RETURN
1021 P = 192: RETURN
1022 P = 197: RETURN
1023 P = 202: RETURN
1024 P = 206: RETURN
1025 P = 210: RETURN
1026 P = 214: RETURN
1027 P = 218: RETURN
1028 P = 222: RETURN
1029 P = 225: RETURN
1030 P = 228: RETURN
1031 P = 231: RETURN
1032 P = 234: RETURN
1033 P = 237: RETURN
1034 P = 239: RETURN
1035 P = 241: RETURN
1036 P = 243: RETURN
1037 P = 246: RETURN
1038 P = 248: RETURN
1039 P = 250: RETURN
1040 P = 252: RETURN
1041 P = 253: RETURN
1042 P = 254: RETURN
1043 P = 8: RETURN
1044 P = 1: RETURN
4000  DATA
   0,115,172,97,3,174,97,3,23
   2,208,253,169
4010  DATA
   4,32,168,252,173,48,192,13
   6,208,239,206
4020  DATA  96,3,208,231,96
```

& Ampersound

Hit that Basic boogie beat with MARK BOWYER's & command routine

AMPERSOUND is a machine code routine (assembled by Merlin) that allows sound effects and music to be played from Basic with the & command. It runs in the same way as Max Parrott's Whirl Graphics system *(Windfall,* November '83, page 54), as most of the system calls come from there.

Listing I is the assembly listing, and Listing II is the equivalent hexadecimal dump.

The commands are:

&NOT TONE,DURATION	*The old Apple Tone routine.*
&POP TONE,DURATION	*A whizzer – reminiscent of Star Trek.*
&POS TONE,DURATION	*Similar, but still different.*
&EXP DURATION,1	*Explosion, or white noise.*

TONE can be between 0 and 255, DURATION between 0 and 191 – except in EXP, where it can be between 0 and 255. This is because Ampersound uses the same hi-res location calls as Whirl Graphics, so that it can be parsed numbers or variables direct, but this also restricts the values.

The demo program in Listing III gives a good idea of their workings. Note that it assumes you've saved Ampersound under that name.

The Musix 2.2 program in Listing IV is really a luxury utility. It has seen many revisions and probably will see many new additions before I'm totally pleased with it.

As it is, though, it is still quite powerful, and you can add anything you might need to it yourself.

Basically, it allows the easy writing and editing of music, and the storing of the finished tune either for re-editing later, or more importantly, as an EXEC file set out to add the music to your own programs at your chosen line numbers.

There is also a Words function – added because a friend asked for it – that writes text between each note, rather like a "sing-a-long".

This is particularly useful for giving messages while the music is playing in your program, as this is also added to the EXEC file.

Each word is a separate string, so 255 characters can be used in each. They are stored separately as Text files (except for the, EXEC), so different words can be added to one piece of music easily – and vice versa.

All files are given a suffix to identify words, tunes and EXECs of the same name.

The notes are input as NOTE,DURATION, with Duration at its parsed value (30 for semiquaver, up in increments of 30), and Note in the form A1N to G2S.

This is a system that harks back to Musix's beginning – as a far smaller program in an old Music Maker magazine.

Basically they mean "first octave A neutral" to "second octave G sharp" – simple really. The tuning isn't perfect, but close enough for all but the perfectionist on an unaided Apple!

Rests are also allowed, and to finish, type END,1.

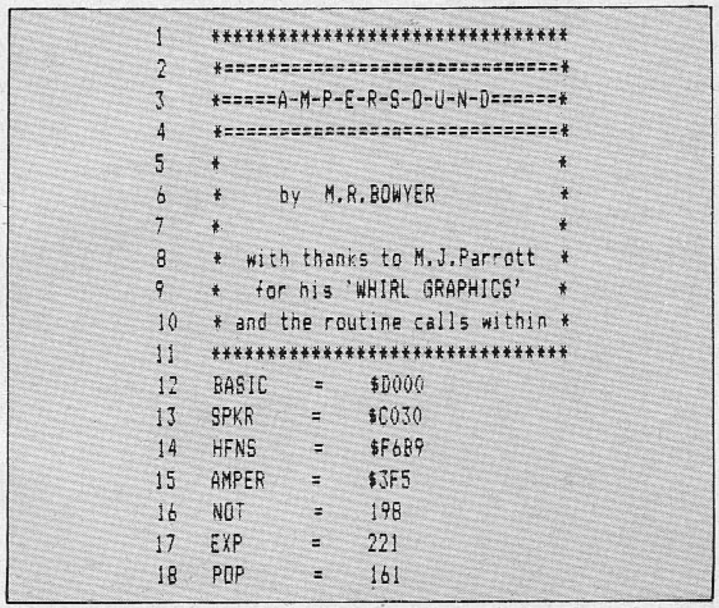
The lists produced in response to the L command are of the POKEd values, so that the actual values can be used in other ways, but these can be easily converted to the above format.

Each note is also numbered for editing and copying purposes.

The copier, by the way, allows sections of music to be put into the list in more than one place without having to retype it each time. It adds the section given by the first and last note to the end.

The program is quite friendly, including an error-catcher for both typing and later running problems, and is mostly self-explanatory.

The whole system is well worth experimenting with, and feel free to convert the set to your own needs.



```
********************************
*==============================*
*=====A-M-P-E-R-S-O-U-N-D======*
*==============================*
*                              *
*       by   M.R.BOWYER        *
*                              *
*  with thanks to M.J.Parrott  *
*    for his 'WHIRL GRAPHICS'  *
* and the routine calls within *
********************************
BASIC    =     $D000
SPKR     =     $C030
HFNS     =     $F6B9
AMPER    =     $3F5
NOT      =     198
EXP      =     221
POP      =     161
```

Listing I: The assembly listing

```
                   POS      =    217
                   ERROR    =    $D412
                   SYNCHK   =    $DEC0
                            ORG  $0300
                            OBJ  $0300
0300: A9 4C                 LDA  #$4C        ;SETUP
0302: 8D F5 03              STA  AMPER       ;THE
0305: A9 10                 LDA  #START      ;AMPER
0307: 8D F6 03              STA  AMPER+1     ;POINTER
030A: A9 03                 LDA  #/START     ;TO TOP
030C: 8D F7 03              STA  AMPER+2     ;PROGRAM
030F: 60                    RTS
0310: 8D 96 03     START    STA  COMM        ;SORT
0313: 20 C0 DE              JSR  SYNCHK      ;COMMAND
0316: 20 B9 F6              JSR  HFNS        ;LINE
0319: 8D 94 03              STA  TONE        ;&CONVRT
031C: 8E 95 03              STX  DUR         ;#S&VARS
031F: AD 96 03              LDA  COMM        ;FOR USE
0322: C9 DD                 CMP  #EXP        ;CHOOSE
0324: F0 46                 BEQ  EXPLODE     ;SNDFX
0326: C9 C6                 CMP  #NOT        ;ROUTINE
0328: F0 0D                 BEQ  NOTE        ;FOR USE
032A: C9 D9                 CMP  #POS        ;AND GO
032C: F0 24                 BEQ  ZAP         ;ELSE...
032E: C9 A1                 CMP  #POP
0330: F0 4A                 BEQ  WHIZZ
0332: A9 10                 LDA  #16         ;..CATCH
0334: 20 12 D4              JSR  ERROR       ;ERRORS
                   *********************************
                   *  AND NOW THE SOUND ROUTINES  *
                   *********************************
0337: AC 94 03     NOTE     LDY  TONE
033A: AE 95 03              LDX  DUR
033D: AD 30 C0     NOT1     LDA  SPKR
0340: 88           NLOP1    DEY
0341: D0 05                 BNE  NLOP2
0343: CE 94 03              DEC  TONE
0346: F0 09                 BEQ  NEND
0348: CA           NLOP2    DEX
0349: D0 F5                 BNE  NLOP1
034B: AE 95 03              LDX  DUR
034E: 4C 3D 03              JMP  NOT1
0351: 60           NEND     RTS
0352: AC 94 03     ZAP      LDY  TONE
0355: A9 00                 LDA  #0
0357: AE 95 03     ZAPTOP   LDX  DUR
035A: 8A           ZLOP1    TXA
035B: 18                    CLC
035C: E9 01        ZLOP2    SBC  #$1
035E: D0 FC                 BNE  ZLOP2
0360: 8D 30 C0              STA  SPKR
0363: E8                    INX
0364: E0 64                 CPX  #$64
0366: D0 F2                 BNE  ZLOP1
0368: 88                    DEY
0369: D0 EC                 BNE  ZAPTOP
036B: 60                    RTS
036C: AC 95 03     EXPLODE  LDY  DUR
036F: AD 30 C0     XLOP1    LDA  SPKR
0372: BE 00 D0              LDX  BASIC,Y
0375: CA           XLOP2    DEX
0376: D0 FD                 BNE  XLOP2
0378: 88                    DEY
0379: D0 F4                 BNE  XLOP1
037B: 60                    RTS
037C: AC 94 03     WHIZZ    LDY  TONE
037F: AE 95 03     WLOP1    LDX  DUR
0382: 8A           WLOP2    TXA
0383: 18                    CLC
0384: E9 01        WLOP3    SBC  #$01
0386: D0 FC                 BNE  WLOP3
0388: 8D 30 C0              STA  SPKR
038B: E8                    INX
038C: E0 50                 CPX  #$50
038E: D0 F2                 BNE  WLOP2
0390: 88                    DEY
0391: D0 EC                 BNE  WLOP1
0393: 60                    RTS
                   TONE     DS   1
                   DUR      DS   1
                   COMM     DS   1

--End assembly, 151 bytes, Errors: 0
```

```
*300.396

0300- A9 4C 8D F5 03 A9 10 8D
0308- F6 03 A9 03 8D F7 03 60
0310- 8D 96 03 20 C0 DE 20 B9
0318- F6 8D 94 03 8E 95 03 AD
0320- 96 03 C9 DD F0 46 C9 C6
0328- F0 0D C9 D9 F0 24 C9 A1
0330- F0 4A A9 10 20 12 D4 AC
0338- 94 03 AE 95 03 AD 30 C0
0340- 88 D0 05 CE 94 03 F0 09
0348- CA D0 F5 AE 95 03 4C 3D
0350- 03 60 AC 94 03 A9 00 AE
0358- 95 03 8A 18 E9 01 D0 FC
0360- 8D 30 C0 E8 E0 64 D0 F2
0368- 88 D0 EC 60 AC 95 03 AD
0370- 30 C0 BE 00 D0 CA D0 FD
0378- 88 D0 F4 60 AC 94 03 AE
0380- 95 03 8A 18 E9 01 D0 FC
0388- 8D 30 C0 E8 E0 50 D0 F2
0390- 88 D0 EC 60 00 00 00
```

Listing II: The hexadecimal dump

```
10  IF  PEEK (768) +  PEEK
   (769) <  > 245 THEN
   PRINT  CHR$ (4)"BRUN
   AMPERSOUND"
20  HOME : INVERSE : PRINT
   SPC( 52)"AMPERSOUND DEMO"
   SPC( 53): NORMAL : PRINT
   : PRINT : PRINT
30  PRINT "1: &EXP DUR,1"
35  FOR A = 1 TO 5
40  &  EXP 255,1
50  &  EXP 255,1
55  NEXT
60  PRINT : PRINT "2: &POP
   TONE,DUR"
70  FOR A = 10 TO 100 STEP
   10: &  POP A,10
80  NEXT
90  FOR A = 10 TO 15: &  POP
   230,A: FOR X = 1 TO 10:
   NEXT X,A
100  PRINT : PRINT "3: &POS
   TONE,DUR"
110  FOR A = 10 TO 150 STEP
   5: &  POS A,11: NEXT
120  PRINT : PRINT "4: &NOT
   TONE,DUR"
130  FOR A = 10 TO 255 STEP
   5: &  NOT A,12: NEXT
140  PRINT : PRINT "  TONE
   CAN BE BETWEEN 1 AND
   255": PRINT "  DUR CAN BE
   BETWEEN 1 AND 191"
150  &  NOT 50,191: &  NOT
   200,119
160  PRINT "       BYE BYE"
20000  END
```

Listing III

```
1  ONERR  GOTO 2100
5  CLEAR
10  GOSUB 1610
15 X1 = 999
20 D$ =  CHR$ (4)
30  DIM
   A(2000),B(2000),A$(2000)
40  HOME : INVERSE : PRINT "
   "
50  PRINT "
   MUSIX                     "
60  PRINT "
   -------------
   "
70  PRINT "
   <MUSIC MAKER>
   "
80  PRINT " &EXEC
   ADDING-------------VERSION
   2.2  "
90  PRINT : PRINT : PRINT
100  NORMAL
110  POKE 34,5
120  GOTO 150
130  REM     ALL PARTS JUMP
   BACK HERE
135  REM

140  HOME
150  PRINT : PRINT "DO YOU
   WISH TO:-"
160  PRINT : PRINT "
   BEGIN.......B
   FINISH....F"
170  PRINT : PRINT "
   LIST........L
   PLAY......P"
172  PRINT : PRINT "  WORD
   ENTER..W          EXEC
   TUNE.E"
175  PRINT : PRINT "
   COPY AREA...Y"
180  PRINT : PRINT "
   ADD OR CHANGE A
   NOTE....A,C"
190  PRINT : PRINT "
   READ OR SAVE A
   TUNE.....R,S"
210  PRINT : PRINT : PRINT
220  PRINT ":";: GET Y$:
   PRINT : IF Y$ = "B" THEN
   GOSUB 430
230  ON ((Y$ = "F") + 2 *
   (Y$ = "L") + 3 * (Y$ =
   "R") + 4 * (Y$ = "S") + 5
   * (Y$ = "C") + 6 * (Y$ =
   "A") + 7 * (Y$ = "P") + 8
   * (Y$ = "W") + 9 * (Y$ =
   "E") + 10 * (Y$ = "Y"))
   GOSUB
   1600,1290,1480,1360,990,10
   20,1050,1650,1140,1930
310  PRINT "TRY AGAIN": GOTO
   130
430  HOME : INVERSE : PRINT
   "               MUSIC
   INPUT               ":
   NORMAL :Z = 0
435 X1 =  - 1
440  PRINT : PRINT : PRINT
450  PRINT : PRINT "TYPE IN
   NOTE(EG.G2S TO
   A1N),DURATION    (E.G.
   30=SEMIQUAVER,180=DOTTED
   CROTCHET)"
460  PRINT " OR REST (E.G.
   REST,120)"
470  PRINT : PRINT
475  IF Z = X1 THEN 960
480  PRINT : INPUT
   "NOTE,DURATION- ";Z$,W
490  IF Z$ = "END" THEN 960
500 L$ =  LEFT$ (Z$,1):M$ =
   MID$ (Z$,2,1):R$ =
   RIGHT$ (Z$,1):Z = Z + 1
520  IF X1 > 0 THEN Z1 = Z:Z
   = X1
525  IF Z$ = "REST" THEN
   A(Z) = 0:B(Z) = W: GOTO
   470
530 L =  ASC (L$) - 64
540  IF L > 7 THEN 610: IF L
   < 1 THEN 610
550 M =  VAL (M$): IF M = 2
   THEN M = 15: GOTO 580
560  IF M = 1 THEN 580
570  GOTO 610
580 R =  ASC (R$)
590  IF R = 78 THEN R = 1:
   GOTO 620
600  IF R = 83 THEN R = 8:
   GOTO 620
610  PRINT : PRINT : INPUT
   "RETRY NOTE: ";Z$: GOTO
   490
620  REM  DROP A LINE
630 B(Z) = W
640  IF  INT (W / 15) = W /
   15 AND W < 191 THEN 660
650  PRINT : PRINT : INPUT
   "RETRY DURATION: ";W:
   GOTO 630
660 N = L + M + R - 2
670  ON N GOTO
   680,690,700,710,720,730,74
   0,750,760,770,780,790,800,
   810,820,830,840,850,860,87
   0,880,890,900,910,920,930,
   940,950
680 A(Z) = 228: GOTO 470
690 A(Z) = 200: GOTO 470
700 A(Z) = 192: GOTO 470
710 A(Z) = 171: GOTO 470
720 A(Z) = 152: GOTO 470
730 A(Z) = 144: GOTO 470
740 A(Z) = 128: GOTO 470
750 A(Z) = 214: GOTO 470
760 A(Z) = 192: GOTO 470
770 A(Z) = 181: GOTO 470
780 A(Z) = 161: GOTO 470
790 A(Z) = 144: GOTO 470
800 A(Z) = 136: GOTO 470
810 A(Z) = 121: GOTO 470
820 A(Z) = 114: GOTO 470
830 A(Z) = 102: GOTO 470
840 A(Z) = 96: GOTO 470
850 A(Z) = 85: GOTO 470
860 A(Z) = 76: GOTO 470
870 A(Z) = 72: GOTO 470
880 A(Z) = 64: GOTO 470
890 A(Z) = 108: GOTO 470
900 A(Z) = 96: GOTO 470
910 A(Z) = 90: GOTO 470
920 A(Z) = 80: GOTO 470
930 A(Z) = 72: GOTO 470
940 A(Z) = 68: GOTO 470
950 A(Z) = 60: GOTO 470
960  IF X1 > 0 THEN Z = Z1
980  RETURN
990  HOME : PRINT "WHAT IS
   NOTE NO.TO CHANGE";:
   INPUT X1
1010  GOTO 470
1020  HOME
1030  PRINT "NOTES WILL BE
   ADDED ON END."
1035 X1 = 0
1040  GOTO 470
1050  HOME : FOR J = 1 TO Z
1060  IF A(J) <  > 0 THEN
   1090
1070  FOR WT = 1 TO B(J) *
   2: NEXT
1080  GOTO 1120
1090  REM
1095  IF ZZ$ <  > "W" THEN
   1100
1096  PRINT A$(J)
1100  REM
1110  &  NOT A(J),B(J)
1120  NEXT J
1130  RETURN
1140  HOME : PRINT : PRINT :
   PRINT
1144 VV = 34
1145  INPUT "WITH WORDS
   ?(Y,N) ";YY$
1150  INPUT "EXEC FILE NAME
   :- ";NM$
1155 NM$ = NM$ + ".MXEXEC"
1160  PRINT : INPUT
   "STARTING LINENUMBER :-
   ";NM
1170 D$ =  CHR$ (4)
1180  PRINT D$"OPEN "NM$
1190  PRINT D$"WRITE "NM$
1200  FOR JT = NM TO NM + Z
```

Listing IV

〉

```
  - 1
1210 J = JT - NM + 1
1220  IF A(J) <  > 0 THEN
   1250
1230  PRINT JT" FOR WT=1 TO
   "B(J)"*2: NEXT
1240  GOTO 1265
1250  PRINT JT;
1255  IF YY$ = "N" THEN 1260
1256  PRINT ": PRINT " CHR$
   (VV);A$(J); CHR$
   (VV);":";
1260  PRINT
   "&NOT"A(J)","B(J)
1265  NEXT
1270  PRINT D$"CLOSE "NM$
1280  RETURN
1290  HOME : PRINT " LIST OF
   NUMBERS TO BE &NOT'ED TO
   MUSIC ROUTINE .": PRINT :
   PRINT
   "NUMBER","NOTE","DURATION"
   : PRINT : PRINT : FOR J =
   1 TO Z
1300  PRINT J,: IF A(J) = 0
   THEN  PRINT "REST",B(J)"
   ": GOTO 1310
1305  PRINT A(J)"    ",B(J)"
   "
1310  IF  INT (J / 10) = J /
   10 THEN 1330
1320  GOTO 1340
1330  VTAB 23: PRINT "
   PRESS A KEY
   ";: GET Y$: PRINT : VTAB
   13
1340  NEXT J
1345  PRINT "
   "
1350  VTAB 23: PRINT "
   PRESS ANY KEY";: GET Y$:
   RETURN
1360  HOME : PRINT : PRINT :
   PRINT : PRINT
1365  INPUT "SAVE WORDS OR
   TUNE (RETURN TO
   EXIT)?:";WD$
1366  IF WD$ = "" THEN 140
1367  IF WD$ <  > "W" AND
   WD$ <  > "T" THEN 1365
1370  INPUT "SAVE FILE NAME
   :- ";NM$
1375  IF  LEFT$ (WD$,1) =
   "W" THEN NM$ = NM$ +
   ".WD"
1376  IF  LEFT$ (WD$,1) =
   "T" THEN NM$ = NM$ +
   ".TN"
1380  PRINT D$"OPEN ";NM$
1400  PRINT D$"WRITE ";NM$
1410  PRINT Z
1420  FOR J = 1 TO Z
1425  IF  LEFT$ (WD$,1) =
   "W" THEN  PRINT A$(J):
   GOTO 1450
1430  PRINT A(J)
1440  PRINT B(J)
1450  NEXT J
1460  PRINT D$"CLOSE "NM$
1470  RETURN
1480  HOME : PRINT : PRINT :
   PRINT : PRINT
1485  INPUT "READ WORDS OR
   TUNE (RETURN TO
   EXIT)?:";WD$
1486  IF WD$ = "" THEN 140
1487  IF WD$ <  > "W" AND
   WD$ <  > "T" THEN 1485
1489  PRINT : PRINT : PRINT
1490  INPUT " FILE NAME
   :-";NM$
1495  IF  RIGHT$ (NM$,3) <
   > ".WD" AND  LEFT$
   (WD$,1) = "W" THEN NM$ =
   NM$ + ".WD":ZZ$ = "W"
1496  IF  RIGHT$ (NM$,3) <
   > ".TN" AND  LEFT$
   (WD$,1) = "T" THEN NM$ =
   NM$ + ".TN":ZZ$ = "T"
1500 D$ =  CHR$ (4)
1510  PRINT D$"OPEN ";NM$
1520  PRINT D$"READ ";NM$
1530  INPUT Z
1540  FOR J = 1 TO Z
1545  IF  LEFT$ (WD$,1) =
   "W" THEN  INPUT A$(J):
   GOTO 1570
1550  INPUT A(J)
1560  INPUT B(J)
1570  NEXT J
1580  PRINT D$"CLOSE ";NM$
1590  RETURN
1600  HOME : TEXT : HOME :
   PRINT "  GOODBYE (TYPE
   'CONT' TO RESTART).":
   POKE 216,0: END
1610  IF  PEEK (768) +  PEEK
   (769) = 245 THEN  GOTO
   1640
1620  PRINT  CHR$ (4)"BRUN
   AMPERSOUND"
1640  RETURN
1650  HOME : VTAB 5
1655  PRINT : PRINT : PRINT
   "  WORD ADDING SECTION ":
   PRINT
1656  GOTO 1750
1660  PRINT "EACH WORD WILL
   BE PRINTED WITH ITS
   CORRESPONDING NOTE."
1670 ZZ$ = "W"
1675  PRINT : PRINT : PRINT
   : PRINT
1680  FOR J = 1 TO Z
1685  IF A(J) = 0 THEN
   PRINT "REST": GOTO 1700
1690  INPUT ":";A$(J)
1700  NEXT
1710  VTAB 24: PRINT : PRINT
   : PRINT
1720  PRINT "LAST WORD:ANY
   KEY TO CONTINUE"
1730  GET YY$
1740  GOTO 1650
1750  PRINT
   "START,CHANGE,LIST OR
   RETURN": INPUT "(S,C,L,R)
   :";FF$
1755  PRINT : PRINT : PRINT
1760  IF FF$ = "S" THEN 1660
1770  IF FF$ = "C" THEN 1800
1780  IF FF$ = "R" THEN
   RETURN
1785  IF FF$ = "L" THEN 1870
1790  GOTO 1750
1800  HOME : PRINT : PRINT :
   PRINT : PRINT
1810  INPUT "CHANGE STARTING
   FROM WHERE ?";AS
1820  PRINT : PRINT
   "TYPE'***' TO END": PRINT

1830  FOR J = AS TO Z
1835 SD$ = A$(J)
1840  INPUT ":";A$(J)
1845  IF A$(J) = "***" THEN
   A$(J) = SD$: GOTO 1650
1850  NEXT
1860  GOTO 1650
1870  HOME : PRINT : PRINT :
   PRINT : PRINT
1880  FOR J = 1 TO Z
1890  PRINT J,A$(J)
1900  IF J / 12 =  INT (J /
   12) THEN  GET QQ$
1910  NEXT
1915  GET QQ$
1920  GOTO 1650
1930  HOME : VTAB 7
1940  PRINT "COPY FROM-TO
   (WILL BE ADDED TO END)"
1950  INPUT ":";CF,CT
1955 X1 = Z
1960  IF (CT < CF) OR CT > Z
   THEN  PRINT "WRONG":
   GOTO 1950
1970  FOR CC = CF TO CT
1980 A(Z + CC) = A(CC):B(Z +
   CC) = B(CC):X1 = X1 + 1
1990  NEXT CC
1995 Z = X1
2000  PRINT "COPIED": FOR WT
   = 1 TO 500: NEXT : RETURN

2100 ER =  PEEK (222)
2110  IF ER = 4 THEN  PRINT
   "THIS DISK IS WRITE
   PROTECTED !!": GOTO 2200
2120  IF ER = 5 THEN  PRINT
   "WRONG FILE TYPE (END OF
   DATA)!!": GOTO 2200
2130  IF ER = 6 THEN  PRINT
   "THIS FILE NOT ON THIS
   DISK !!": GOTO 2200
2140  IF ER = 8 THEN  PRINT
   "DRIVE PROBLEM !!. CLOSE
   DOOR .": GOTO 2200
2150  IF ER = 9 THEN  PRINT
   "DISK FULL . START ON NEW
   DISK!!": GOTO 2200
2160  IF ER = 10 THEN  PRINT
   "THIS FILE IS
   LOCKED.WRITE TO NEW
   FILE!!": GOTO 2200
2170  IF ER = 16 THEN  PRINT
   "SYNTAX ERROR AT LINE "
   PEEK (218) + 256 *  PEEK
   (219)".TYPING ERROR!!":
   POKE 216,0: END
2175  IF ER = 254 THEN
   PRINT " BAD
   RESPONSE.PLEASE
   RETYPE!!": GOTO 2200
2176  IF ER = 255 THEN
   PRINT "'CTRL-C' BREAK
   ATTEMPTS DISALLOWED !!":
   GOTO 2200
2180  PRINT " SYSTEM ERROR
   AT LINE " PEEK (218) +
   256 *  PEEK
   (219)".PROBABLE
   TYPING ERROR. ": POKE
   216,0: TEXT : END
2200  FOR A = 1 TO 2000:
   NEXT : IF ER < 16 THEN
   PRINT  CHR$ (4)"CLOSE":
   GOTO 140
2210  RESUME
```

Shaping up to using Shape Editor

GREAT, I thought on seeing your listing for a Shape Editor in the March '85 Apple User – at last an editor which can take shapes from other Shape Table editors and add new shapes.

I have four other shape editor programs – Apple Mechanic, Graphics System, Graphics Magician and Roy E. Myer's program from his excellent Graphics text.

Most of these programs like their own shapes!

Having found time to type in your listing I checked it out by redrawing a number of shapes. All OK.

I then attempted to add a fourth shape to a three shape table drawn and coded manually from Applesoft.

Using Add Shape I correctly entered the shape and was pleased to note its acceptance and correct decimal listing.

I added it to the earlier loaded table and was informed that my table was now a four shape table.

Attempts to display this shape proved negative. Its length indicated on listing table info was 1 address.

I checked the data in the 16384 address and found it to be modified only to the extent that it was a four shape table with the fourth shape a blank.

I repeated this by calling up the shape I had previously drawn in "New Table" and redrew it as an add on shape.

This time it added and was displayed as the second shape in the previous three-shape table I had loaded and subsequently deleted.

I have spent two whole evenings checking and rechecking the listing but cannot locate any errors. No apparent errors are indicated in the running.

Perhaps if you could comment on this problem with the info above I could have another go with your program.

At this time I feel I have wasted quite a lot of terminal hours. – **Ian D. Entwistle, Sittingbourne, Kent.**

● I'm sorry you are having such trouble with the shape table program. I assure you that the editor itself works fine and many of the tables I've used in past columns were created and edited with it.

As a general word of warning, not all commercial packages create Applesoft shape tables. Some, like Penguin's Graphics Magician or the Tool Kit character generator, produce "bit mapped" shapes which work on a totally different principle.

However, since you say you can display shapes already in a table this obviously isn't the case here.

Unfortunately, without a listing or a disc containing your program it's rather difficult to suggest what might be wrong.

I have sent you an original listing which is a bit easier to read than the reproduction in *Apple User*.

Points to watch out for are mixing up 0 and O or I and 1.

From the description of the problems you have encountered it would appear that the shape is constructed correctly but that it isn't getting POKEd into memory.

I suggest putting PRINT statements before each of the POKEs in the routine at 1200-1310 to see what is being put into memory – and where.

You should see the whole table being put into memory if you do this. – **Peter Gorry**

Koala connection

AS an Apple user for some time, I recently decided to update my II+ to a IIc.

I encountered a few problems with connecting a Koala pad to my IIc, as the pad is designed to fit in the Game I/O socket, inside the IIe and II+. So I hope the following will be of use to other IIc users.

If you cut the plug off the Koala pad, you will find seven wires, namely yellow, sky, green, brown, orange, black and red. They should be connected to a nine-way "D" available at most electronic shops, as in Figure I.

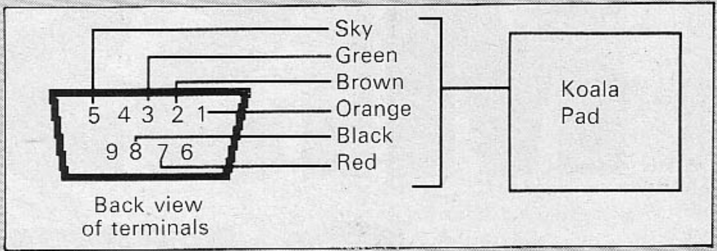


Figure I: Connection to nine way D plug

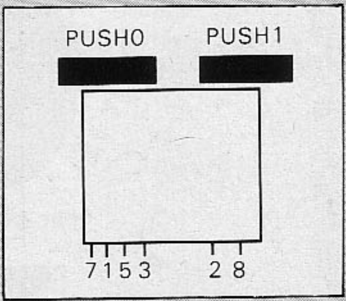


Figure II: Koala Pad circuit board

If you don't find the colours match, then the connections are as Figure II. (The numbers in Figure II correspond to the pin numbers of the nine-way "D".) The yellow is not connected.

Be warned, though, this conversion will void your warranty on your Koala pad.

The connection of a II+, IIe joystick to a IIc is as follows:

16 way	9 way "D"
Pin 1	Pin 2
Pin 2	Pin 7
Pin 3	Pin 1
Pin 4	No connection
Pin 5	No connection
Pin 6	Pin 5
Pin 7	No connection
Pin 8	Pin 3
Pin 9	No connection
Pin 10	Pin 8
Pin 11	No connection
Pin 12	No connection
Pin 13	No connection
Pin 14	No connection
Pin 15	No connection
Pin 16	No connection

Note: The pin numbers on the 16-way connector are on the pin side of the plug.

I am at present compiling a list of programs and different versions of programs that are not compatible with the IIc, so if you will print my name and address I would be pleased to hear from other users with any info. – **David Palmer, 4 Ludlow Grove, Bispham, Blackpool.**

Minotaur's lair

IN the February Feedback column you mentioned that TSR had stopped distributing Theseus and the Minotaur. This is certainly true, but your readers may like to know that we still have a few copies left in stock.

The general point is that it's always worth making a few phone calls before deciding that a game is unavailable. – **Robin A. Hood, Woodland Software, London.**

Want to write to Feedback? The address is: Feedback, Apple User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Pascal's no great fun with only one drive

I HAVE just started to learn Apple Pascal and have come up against the problem of not being able to load the system using an Apple IIe with single disc drive.

I thought Gordon Findlay's Pascal tutorial in November's issue of Apple User was going to be my saviour. Unfortunately all the information he gave was that a modified loading procedure with a modified set of discs was required.

I have spoken to P&P Micro Distributors, from whom we bought the program some time ago, and Apple UK but have had no joy from either.

Would it please be possible to tell me how to modify Apple Pascal 1.1 to work under my circumstances. Please bear in mind that my knowledge of Pascal is extremely limited but I understand DOS 3.3 and Assembly language fairly well. – **J.P. Holden, Greenwich, London.**

● The trouble with operating Pascal with only one drive is that it's difficult, there's not much space, and it takes all the fun out of it.

Running Pascal is the greatest spur I know to acquiring a second drive!

If you want to go ahead, I think the easiest way is to make two copies of APPLE1: using COPYA and then put the original safely away.

Boot one of the copies using PR#6 and press F to enter the filer. Press R and when the prompt appears type SYSTEM.CHARSET and then Y in response to the second prompt.

In the same way remove SYSTEM.LINKER and SYSTEM.FILER. Then press K to crunch the disc and Y in response to the prompt.

Replace the disc with APPLE2: and press T. In answer to the prompt type APPLE2: SYSTEM.COMPILER and in response to the second prompt type APPLE1:$.

If you follow the instructions you should end up with a disc with 30 free blocks on it. Test this by typing E and then type APPLE1: in reply to the prompt.

This disc can then be used for creating simple Pascal programs and compiling them. The drawback is that programs must be small, but when you're learning programs are generally short.

Also every now and then you will have to take this disc out, put the other APPLE1: in and enter the filer by pressing F, then replace the disc with your working APPLE1: and crunch it as before.

This is to make sure that the disc's space is being optimised. I hope this is of some use to you while you ponder the purchase of another drive.

Max Parrott

SpeeDemon and Corvus

FURTHER to Max Parrott's review of the SpeeDemon card, (Apple User, April 1985) I can confirm that the card works perfectly well with my Corvus hard disc.

Incidentally, I also have an Accelerator card which performs as it should with the disc. – **Trevor Neil, Speldhurst, Kent.**

Delighted with discs

I'VE noticed recently that you have taken to omitting long listings and instead offering the programs on disc.

I'd just like to say I think this is a great idea – keep it up! – **Colin Henderson, Erith, Kent.**

Pascal posers

I HAVE been an Apple II+ owner for seven years. Most of my programming has been in Basic with a little in machine language.

About 12 months ago I decided the time had come to branch out and learn Pascal so I purchased a Digitek 16k RAM card, a Vision 80 eighty column card and the Pascal operating system complete with the Language and Operating System Manuals.

After being completely lost for the first few months I have now learnt enough to be able to write quite complex programs. (Although I still cannot find any reference to Variant Records in any of the published literature, and my brother-in-law insists that they are the elegant solution to many problems!)

I am however completely baffled by one problem.

It is a common feature of many of my hybrid Basic and machine language programs that I use both the Vision 80 screen and the Apple 40 column screens for output.

For example the Apple screen can hold a menu and the 80 column screen the data output.

Toggling between the two is easily achieved by the use of the commands Ctrl-Z 4 and Ctrl-Z 8 as follows:

10 PRINT CHR$(26); "4"

(To switch from Vision 80 to Apple screen)

and

20 PRINT CHR$(26); "8"

(To switch from Apple to Vision 80 screen)

after the initial PR#3 command to initialise the Vision 80 card in slot 3.

Imagine my surprise therefore when similar commands in Pascal did not work!

The command Ctrl-Z "T" works all right to switch from Vision 80 to Apple hi-res screen ready for the Initturtle Pascal hi-res plotting command.

Also if I use a procedure to implement a POKE command in Pascal, I can then switch from the hi-res page to the Apple 40 column screen by use of a POKE (–16303,0) reversing this with a POKE (–16304,0).

But any Pascal WRITELN command still appears on the Vision 80 screen as it is only the video display that has been switched and not the output vectors.

I know this is the case as I can get text to appear on the Apple 40 column screen by POKEing it there, but this is very cumbersome and not the way an elegant language like Pascal should be used.

I have tried to discover where the Pascal output vectors are located and have made some progress in this direction with the following:

D772	Console keyboard check routine.
D898	Console init routine.
D950	Console write routine.
BF27-BF2E	Slot types table.

Also it appears that the COUT vector at $0036 and $0037 (normally set to $FDED with DOS active) is still used as it points to $C305 when the Vision 80 in Slot 3 is active, $C105 when the printer in Slot 1 is receiving output and $FDFO when I remove the Vision 80 card and revert to the Apple 40 column screen for all output.

I have tried changing these parameters but with zero success.

I know it must be possible to re-vector the output to the Apple screen from within a Pascal program as this is done when the PRINTER: is selected to receive output, but I am at a loss to know how. – **John Avery, Newbury, Berks.**

● Starting with variant records: see Stuart Bell's Pascal Tutorial in this issue. The Pascal User Manual and Report by Jensen and Wirth (sold with the Apple Pascal system) also covers variants as do two books which I would recommend:

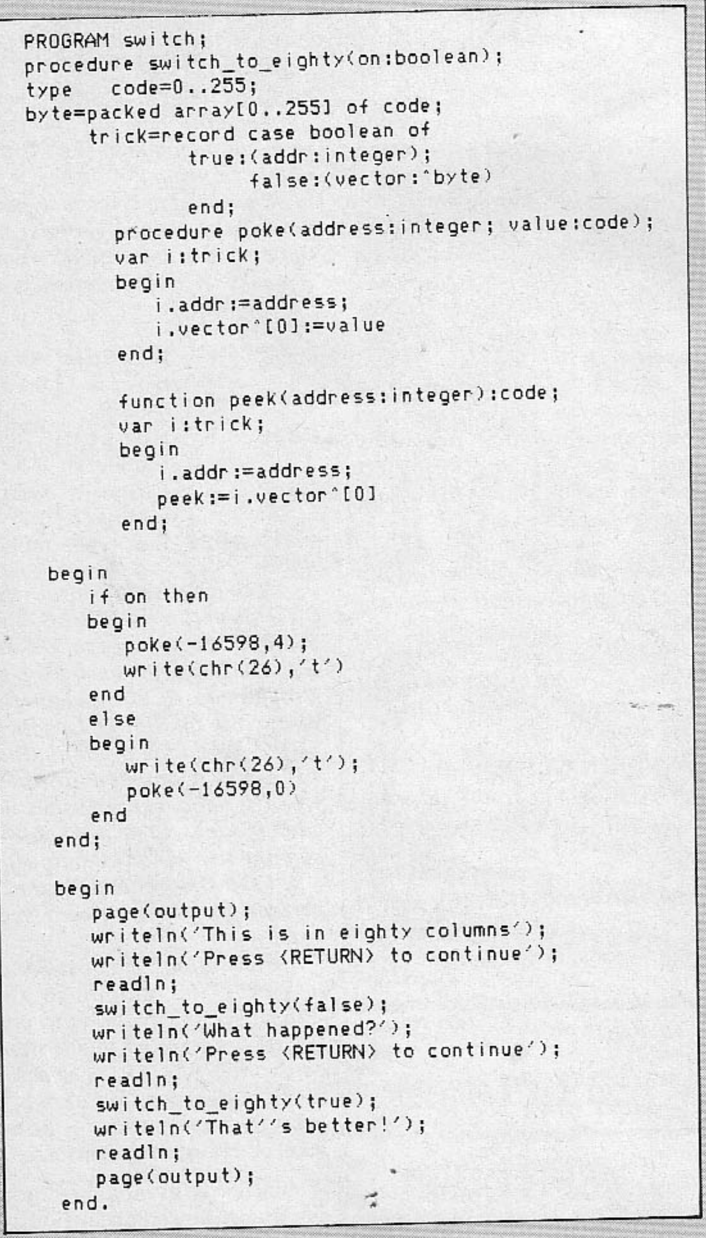
An Introduction to Programming and Problem Solving with Pascal, by G.M. Schneider, S.W. Weingart and D.M. Perlman, John Wiley and Sons, New

〉

York, 1982 (2nd Ed.), ISBN 0-471-08216-3.

And a cheaper but no-less-good book – Practical Pascal for Microcomputers, by R. Graham, Sigma Technical Press, Wilmslow, Cheshire, 1983, ISBN 0-905104-17-X.

Now, with regard to the 80/40 column switching. I have found a way to do what you want, BUT it is by no means perfect. The following program will demonstrate the method in action.



```
PROGRAM switch;
procedure switch_to_eighty(on:boolean);
type   code=0..255;
byte=packed array[0..255] of code;
     trick=record case boolean of
             true:(addr:integer);
                  false:(vector:^byte)
             end;
      procedure poke(address:integer; value:code);
      var i:trick;
      begin
         i.addr:=address;
         i.vector^[0]:=value
      end;

      function peek(address:integer):code;
      var i:trick;
      begin
         i.addr:=address;
         peek:=i.vector^[0]
      end;

  begin
     if on then
     begin
        poke(-16598,4);
        write(chr(26),'t')
     end
     else
     begin
        write(chr(26),'t');
        poke(-16598,0)
     end
  end;

  begin
     page(output);
     writeln('This is in eighty columns');
     writeln('Press <RETURN> to continue');
     readln;
     switch_to_eighty(false);
     writeln('What happened?');
     writeln('Press <RETURN> to continue');
     readln;
     switch_to_eighty(true);
     writeln('That''s better!');
     readln;
     page(output);
  end.
```

When the Pascal system boots, the slots are searched for cards and, as you point out, the results of the search are stored in locations $BF27—BF2E.

Thus for slot 3 ($BF2A, –16598) the value 4 is stored for a Vision-80 card as it is treated as a serial card. It occurred to me that if a zero were stored in this location the system would probably ignore the card and write directly to the screen.

I started checking the BIOS (bank 2, $D000 – DFFF) at $D950 as you pointed out and found that location $BF2A is indeed checked and my program thus works.

However, if an attempt is made to write outside the screen limits, for example with a "page(output)" or a "write (chr(12)", the system crashes and is reinitialised.

I searched further and found that when the Vision-80 is active the system claims most if not all of the pages $800-$C00 as workspace.

For example, without an 80 column card the output file information block appears to exist at $C7E onwards but with the card this block exists at $87E onwards.

I take this to mean that only the first page of 40-column text can be used and great care has to be taken not to encroach on the second text page.

All in all I think you will be better off not using your page switching scheme (although it does seem a waste of 1k of memory) because your resulting programs will be very machine specific.

An Apple IIe for example with its 80-column card will only work in alternate columns and I suspect that the new Pascal 1.2 will not function as you want either, and of course other 80-column cards in a II+ will not allow the easy switching of video output.

Maybe some other readers have tackled this problem and can suggest alternative solutions. – **Max Parrott.**

A race apart . . .

HAVING recently pondered on purchasing a Macintosh computer, it has occurred to me that most of the programs I had previously acquired for the Apple IIs won't behave as they are supposed to due to difference in size of the display screen (pixel-wise, that is).

Since I haven't had the chance to actually test this out myself could you mention the problems that might arise?

So many of the advertisements I've come across in Apple User and other unmentionable sources are mostly designed for the Apple II range of computers. I'm sure that other Mac users have come across this little fault in which the computer isn't fully compatible with the rest of the Apples and I think it would be extremely handy if something is developed to remedy this (that's unless there isn't something on the market already, if so could you please let me know).

Also, is it possible for the Keyport 717 (September 1984 issue) to be fitted to the Mac, as it would be a great boon to program construction?

Lastly, can the Mac, or the Apple IIs for that matter, operate a colour printer or plotter because it would give a document/chart etc, that extra something a black-on-white copy can't? And can the Mac be used on a colour TV/monitor as it would give startling 3D effects? – **J. Kresin, Queensland, Australia.**

● What a wonderful sense of the under-statement you have! The Mac is just about completely incompatible with the Apple II range, being based on the 16 or 32 bit (depending on how many fingers you've got) 68000 chip rather than the 8 bit 6502.

From a programming point of view, anything but the simplest Basic program will be difficult to transport. Also running such programs on the Mac would be a bit like keeping your Porsche permanently in reverse.

If by programs you mean commercial software, then Apple II software will certainly be no good if you buy a Mac. Not the least of the problems is that Mac software comes on 3½in discs.

Also, since the two operating systems are completely incompatible your chances of getting a commercial program up and running are somewhat less than favourable.

Hence if you buy a Mac you'll have to start from scratch as far as software is concerned.

Many software houses are writing "special" versions of their best-sellers which take advantage of the Mac's features – the mouse, pull-down menus, etc.

This means that you might have to do a bit of re-learning even with standard packages. However devotees of the Mac would tell you it's worth it.

With regard to a colour Mac, see the story on page 6 of the May issue of *Apple User*.

It is also possible to drive a colour printer, and hence obtain coloured output. Mind you, it's also possible to do this from an Apple II.

Some printer interface cards cater for colour printing – Digitek's Super Printmaster III and Thirdware's Fingerprint spring to mind, but there are probably others.

As far as we are aware, the Keyport 717 is not Mac-compatible. However a call to the manufacturers would give you the up-to-date position regarding their intentions.