USER Einstein

An independent quarterly magazine Volume 1 Number 1 November 1984 Price 95p

Inside the Einstein

ZEN Assembler review

DR Logo-more than kids' stuff

Languages on the clean machine

Sprite Editor program listing Game Play

A message from Dr T.S. Lin

Chairman, Tatung (U.K.) Limited

designers and manufacturers of the Einstein computer

Since we launched the Einstein personal computer on to the market earlier in 1984, its success has been a tribute to the design and manufacturing expertise of United Kingdom industry combined with the marketing and electronics know-how we have been able to supply from our home base in Taiwan.

We are very pleased that its pre-eminence has been recognised so early in its history by the establishment of a "user magazine" to serve the many thousands of families and small businessmen who are finding its low-priced high technology so appealing.

Though modest in its beginnings, we are sure that the experienced team who are putting the magazine together, who have proved themselves in a number of extremely successful international computer publications, will take it from strength to strength.

For our part, our technical staff are giving as much in-house assistance as we can — tough they are, of course, very fully occupied in developing the potential of the Einstein still further.

As the user base expands, we hope to see it move from quarterly publication to bi-monthly, and then to monthly sale on the news stands of the world.

For already the Einstein has begun to make an impact on world markets – even including the Orient, which in many ways has pioneered the development of consumer electronics – and exports of the Einstein to Taiwan are making a significant contribution to Britain's balance of payments.

I believe there is a British saying about sending coals to Newcastle which may be applied to this situation.

This magazine is not a house organ from Tatung (UK) Limited. All contributors have total freedom to express their views on the Einstein and its associated products. That freedom, of course, applies especially to you, the readers. Let us know what you are thinking, your suggestions, your criticisms, your ideas on how it may be improved.

We already believe the Einstein to be the best personal computer in the world.

With your help, we can make it better than the best.

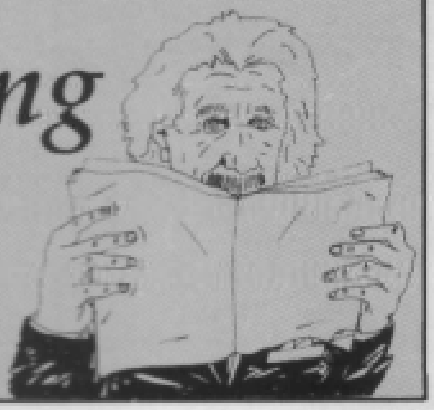
T.S. LIN
Taiwan, R.O.C.

Volume 1, Number 1 an independent quarterly magazine November 1984

Relatively speaking

a news diary

by Bert Stone



The Wired Yule

Are you ready for the electronic Christmas?

Computer games to keep the kids out of our hair as we prepare the Christmas dinner are already part of the established Yuletide pattern, but now that more sophisticated software is becoming available for the home micro, it's now becoming possible to hand all the planning over to the friendly machine blinking its READY sign over there in the corner of the room.

Christmas card addresses? Simple. You can organise them with Kuma's "Database", and if you think £19.95 is too much to pay for that kind of peace of mind, then remember that the same program can help with everything from running the local tennis club to handling the affairs of a small business.

Christmas receipes? No problem. The same program will help.

You could even use "Database" to program in all the radio and TV shows you want to have on over the festivities, and what you want to get on the family VCR for watching later. (In due course, no doubt you'll be able to use your Einstein to actually switch the TV or radio on and off at the appropriate times, control the VCR and so forth, but the hardware's not quite ready as yet.)

Budgeting? well, there's a "Home Budget" program (Kuma, £19.95) that can help you there, though you might find the added power of "The Cracker", Einsoft's superb spreadsheet, worth £57.50 to get the whole sprawling festivities under some kind of financial control.

A spreadsheet, if you're new to computer jargon, is a sort of "what if" program. It's most useful for businesses where the MD might want to work out, for instance, how much profit will be affected if the profit margin is cut to attract more business, how much extra trade is needed to break even ... that sort of thing.

With a spreadsheet, you can check the figures over and over again, and set budgets for yourself so you won't overspend.

If you want to make sure the Einstein earns its keep as an educational tool, you might like to think about the following stocking fillers for the kids in the form of educational software: "Alphabet Quest/Maths Quest", "Mathsmagic/Scoop", "Maths Tank/Counter Blast", "Mighty Writer/Giant Maths/Rocket", and "Mr Fixit/Get Lost" (all Solo, £14.95).

I was particularly struck by "Scoop", which is a business simulation about running a newspaper profitability. Perhaps we could pick up the odd wrinkle there!

Then there are always the games.

Some are reviewed elsewhere in this issue, but others worth noting are "Backgammon/3D Noughts and Crosses" (Solo, £14.95), "Bat Attack" (Kuma, £12.95), "Commando Plain/ Nightmare Park" (Solo, £14.95), "Cursed Chambers/Zrim" (Kuma, £14.95), "Cylon Attack" (Einsoft, £12.95), "Disco Dan" (Einsoft, £12.95), "Galaxoids/Manik Panik" (Solo, £14.95), "Islands of Artuan" (Kuma, £19.95), "Superchess" (Kuma, £19.95), and "Tombs of Karnak/Encounter" (Solo, £14.95).

Coals to Taipei

As Dr Lin says in his message of greeting overleaf, it's a bit like sending coals to Newcastle, but it's true.

Five thousand Einstein micros have been shipped to Taiwan, Tatung's home base, as a first contribution of the Einstein to Britain's export drive.

Prompt deliveries shouldn't be worthy of note, but in an industry that is so long on promises and rather short on delivery on schedule it is unusual, so we have to point out that the Einstein was delivered on time in June, and that they have been rolling off the production line so regularly that over 50,000 will have been delivered to the shops in the computer's first six months of business.

That record compares rather well with some over-hyped micro products from other companies we could mention.

By now, Tatung's £10 million production complex at Telford should have begun to contribute to this output – and also, incidentally, helping to reduce unemployment in the area by 1000 jobs over the next five years.

It's planned to produce 150,000 Einsteins in the 41-acre Telford factory next year.

The Einstein, it's worth pointing

out, is both designed and assembled in the UK.

My picture shows: one of the first Einsteins off the production line put on sale at the nearest dealer outlet to the main Tatung UK factory, Whitfields of Bridgenorth. Tatung rep Roger Ward unwraps the new machine for sales assistants Wendy Childs (left) and Mandy James.

The Enhanced Einstein

I was much impressed by the news at the Personal Computer World Show in September that the awaited 80-column card for the Einstein was now available. For less than £50, therefore, it's possible to upgrade it to a true business machine, capable of running most CP/M software.

To achieve the necessary high resolution, of course, the display has to be monochrome, so at the same time a 12in green screen monitor was released, price £90.85. (If you've already got the colour monitor, you may not have realised that this can be switched into phosphor green monitor mode with a switch on the back, right-hand side.)

There's also news that the long-awaited BBC Basic is now available in a configuration suitable for the Einstein – quite a feat in itself, since the Acorn BBC micro is based upon the 6502 chip, and the Einstein is a Z80 machine, using quite a different chip family in its CPU.

One of the things I like about the machine is the fact that not only is it upgradeable, but it's easy. If you have a single-disk 40-column machine, you can turn it into twin-disk 80 column for only £239.95.

Other nice add-ons are an 80-characters-per-second dot matrix printer (£228.85 plus £11.50 for parallel printer cable) and an external disk drive, if preferred, for £189.99.

More enhancements are on the way, I hear. Watch this space, as they say.

HOW TO UPGRADE

All new purchasers of the Einstein will now receive Digital Research's DR Logo on their system disk.

But what about those early birds who got their machines back in June-September '84?

No sweat.

If they sent back their registration cards, all they'll need to do is to return their system disk with details of their model number, etc, and DR Logo will be added to it.

What's that? You haven't returned your registration card?

Shame on you!

It's essential that buyers return their registration cards, so that the Tatung people know that everyone who contacts them are legitimate Einstein owners.

Then, as improvements are made to their systems software – or any other part of the system – they can be advised how to obtain the upgrade.

Otherwise: no registration, no upgrade.

Incidentally, returning the registration card is the only way you can be sure of getting your free copy of Einstein User.

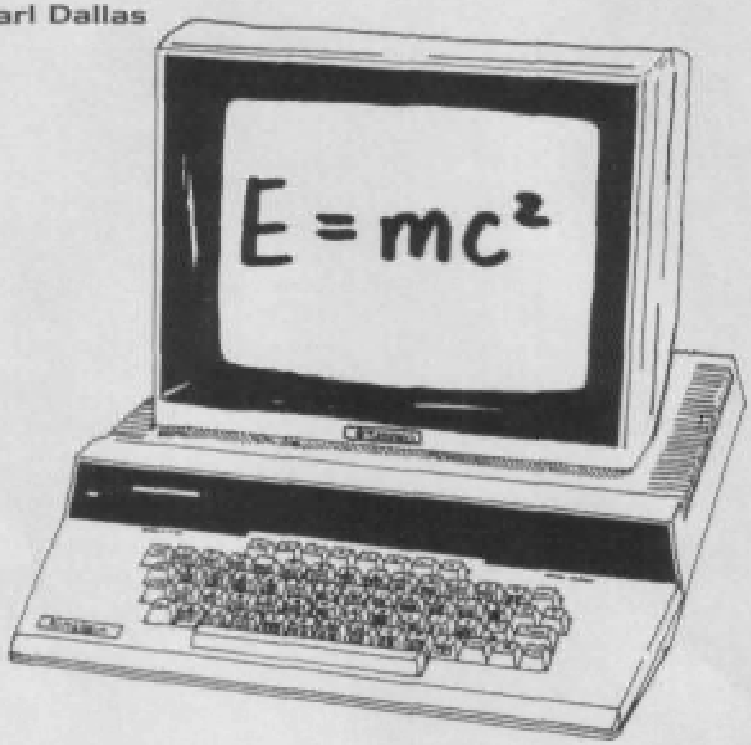
Now that's an offer you really can't refuse!

The Einstein is a clean machine.
Turn it on, and it's at machine-operating system (MOS) level, and the only way you can talk to it is in machine language (ML), the binary notation which is almost impossible for the beginner to understand. But, actually, it's much more versatile than that. The lack of a resident programming language means that you can LOAD in a wide range of languages as they are needed.

Languages

Talking to the clean machine

by Karl Dallas



Basic

Put your system disk into the drive, type

XBAS

and you can talk to it in something like plain English. Type

PRINT 2+2

and it will tell you the answer is 4.

XBAS, as you probably know, is short for **Xtal Basic**, a special application of the language devised at Dartmouth College as a *Beginners' All-purpose Symbolic Code* for students who had difficulty in communicating with computers. Basic was itself developed out of Fortran (*Formula Translator*), still a widely-used scientific programming language.

Basic is a much-maligned language among those who practice the arcane art known as structured programming, and it is true that its most useful command, the GOTO, can tempt you into very untidy programming practices. But as implemented for the Einstein by Crystal Research Ltd of Torquay, it's very powerful, with some extra valuable commands, notably ELSE, which if added to the IF ... THEN conditional test, does encourage more structured constructions, especially if linked with GOSUB rather than GOTO.

BBC Basic

One of the most interesting other Basics is that devised for the BBC micro, and of course the widespread adoption of that machine in schools and other educational establishments means that many have been trained in its use. Fortunately for them, **BBC Basic** is available for the Einstein too, and since it's a clean machine, it can be LOADed into the machine at DOS (disk operating system) level.

However, the implementation of BBC Basic devised for the Einstein by R.T. Russell keeps the nicest features of the original; and adapts them specifically for the new host machine, like COLOUR (spelt the British, not the American way, you'll note, which is more usual with computer speak), which can be used in a similar way to the same command on the BBC machine, but with a range of colours specific to the Einstein. The SOUND command has been modified to make it easier to refer to different ENVELOPE definitions.

There is also a valuable *SPRITE command, which makes the manipulation of graphic figures much easier.

BBC Basic for the Einstein will shortly become available from M-Tec Computer Services, Ollands Road, Reepham, Norfolk NR10 4EL (Tel: 0603-870620).

CIS Cobol

In the 1950s an American organisation called CODASYL (it stands for Committee on Data Systems Languages) devised a language to be known as Cobol, the *Common Business-Orientated Language*. Though it was particularly suited for commercial applications, for some reason the US

Old favourites revisited

Name	*Pakman/ Millipede*
Supplier	*Kuma*
Price	*£14.95*
Category	*Arcade games*
No of Players	*1*
Verdict	*Old favourites for the new machine*

Rating (■■■■■ is maximum)

Ease of loading	■■■
Instructions	■■■
Playability	■■■■
Difficulty	■■■■
Use of graphics	■■■
Use of sound	■
Overall	**■■■**

Is there anything we can say about either of these games except that they are deservedly popular, and here they are?

Both are .COM files which must be LOADed from Xta1DOS by just typing in the program name (no quotes). Beware that the second game is named "Milipede" on the disk.

Both have the same up and down key controls – S/X, and right is L. On Pakman, left is K, but on Millipede it's been moved one over to J, to allow K to be "fire".

The control keys can be changed by the player.

Playing at millionaires

Name	*Tycoon/ Pelmanism*
Supplier	*Kuma*
Price	*£17.95*
Category	*Board/card game adaptions*
No of Players	*2/8 - 2/4*
Verdict	*Good implementations*

Rating (■■■■■ is maximum)

Ease of loading	■■■
Instructions	■■
Playability	■■■■
Difficulty	■■
Use of graphics	■
Use of sound	■
Overall	**■■**

Tycoon is a M*n*p*ly-style game (though with no Chance, Community Chest or Go To Jail) about investing money in six companies, and seeing it either grow or depreciate. There are 11 rather complicated rules spread over four closely-typed screens.

The players can decide how much money they're going to start with, the target sum which has to be achieved by one before a winner is declared, and the length of time each transaction is allowed to take. If a news flash takes place during your turn, then tough: it'll probably eat up most of your ten seconds, or whatever time has been allocated.

Pelmanism is that old game where you have to turn up two cards to see if they're the same. If they're not, they're covered again, and you have to try to remember them when you turn up either of the same elsewhere on the board. Instead of the normal hearts, clubs etc, the cards have various graphics characters on them, which may be harder for some to remember.

There are 108 card-positions (not all taken), so it's a bit harder than the card game.

Two good ones for people who don't like zapping millipedes or adventures.

Never a cross word

Name	*Xanagram/ Quadrax*
Supplier	*Kuma*
Price	*£17.95*
Category	*Word games*
No of Players	*1/1 - 2*
Verdict	*Good training for crossword beginners*

Rating (■■■■■ is maximum)

Ease of loading	■■■
Instructions	■■■■
Playability	■■■■
Difficulty	■■
Use of graphics	■■
Use of sound	■
Overall	**■■**

Xanagrams is a sort of multi-dimensional hangman, with three levels of difficulty (3 is hardest, but there's no indication), and up to five words available to be guessed at a time. Letters available are displayed in alphabetical order on one side of the screen, and vanish as they are guessed successfully. If the player is stumped, pressing 1 will reveal the required letter.

The rather lugubrious bleeps every time you get a letter wrong (including using a cursor to go up a horizontal word, for instance) can be turned off.

Quadrax is another multi-dimensional variation on a well-known game. This time it's noughts and crosses. If you're a single player, your opponent is the computer, who displays an "I'm thinking" message while it sorts out the logic of its best response.

You can't decide exactly what square to pick, only the column, so winning is not so easy as you might expect.

A simple game, but fun. A good starter for those who haven't yet developed the necessary hand-to-eye co-ordination, or logical skills, for the really tough games.

Software In Depth

ZEN Z80 Assembly Language Programming System

If the Einstein is your first computer and you're just getting into programming, then you'll probably be familiar with the Basic language. But it may surprise you to know that the Einstein doesn't understand Basic.

The language your computer understands is machine language (ML), a collection of figures in hexadecimal notation that can be very daunting for the beginner. (To see what it looks like, turn to page 174 of the Einstein manual.) Inside the computer is an *interpreter* which translates your Basic into machine language. Every Basic instruction has to be interpreted into ML before it can be executed, which is why Basic programs run slower than those written in ML.

You can enter ML from the machine operating system (MOS) – this is the mode you'll get by typing MOS from Basic or DOS, or if you power up without a disk in the drive. Then type M plus a number (try &0100) and you'll see whatever you have in memory. If you've just turned on, there won't be much there and you can type in a program in ML. They are often published in computer magazines. Press ‹ENTER› at the end of each line, and a full stop + ‹ENTER› after the last line. To RUN the program you have to type G plus the number you typed at first (the START address) and ‹ENTER›.

The problem is that you probably will have made a typing error, and the chances are you'll "hang" the machine. (Don't worry: though it's alarming, a "hang" won't damage the machine, but you'll have to reset it to start again.)

And anyway, typing in all those numbers and letters without the faintest idea what you're doing can be deadly boring.

The solution is an "assembler", which is a sort of in-between programming language, not so easy on the eye as Basic, but not so dense as ML. It uses fairly easy-to-understand mnemonics, like LD for "load", to create a text file of what's known as source code. Then it translates this into what's known as object code, which is actually our new friend ML. The ML can be SAVEd, LOADed, RUN like any other program.

The Art of Zen

Kuma's ZEN is a fine example of what can be done with an assembler. It's well documented, as far as it goes, and works well. It's not much help to the absolute beginner, however, because there are no examples or exercises to work through.

ZEN is best LOADed from DOS command level with the simple command

ZEN‹ENTER›

after which the program will be LOADed into the transient program area and control transferred to it.

The opening screen tells you one of the nicest things about ZEN: that it uses easily memorable single-character commands, which barely need any explanation:

A – Assemble	O – Out
B – Bye	P – Print
C – Copy	Q – Query
D – Down	R – Read
E – Enter	S – Sort
F – Fill	T – Target
G – Goto	U – Up
H – Howbig	V – Verify
I – In	W – Write
K – Kill	X – eXamine
L – Locate	Z – Zap
M – Modify	d – disassemble
N – New	u – unscramble

Mistypings can be corrected with the ‹DEL› key and ‹CTRL›+A will perform a screen dump to the printer. If ZEN doesn't understand a command, it gives you the friendly error message:

HUH?

Pressing ‹ENTER› on its own will clear the screen.

The "disassemble" instruction is interesting, because it allows you to examine the way programs work – yes, including ZEN itself, and the Einstein's Basic commands. They can be LISTed to the screen or

Government adopted it for naval use, and it's still a US Federal requirement that any Government-purchased computer must be capable of running Cobol.

A British company, Micro Focus, devised a special version called **CIS Cobol**, and it is this which is shortly to become available on the Einstein. The price will not be cheap – some £375 – but when you see the size of the manuals alone, you'll realise that this is a really powerful programming tool.

Forth

All these languages require you to learn a vocabulary, not all of which is as close to English as they would like to claim.

Commands like ADVAL or GCOL are easy to understand, once you know what they mean (*ADvise VALue* and *Graphics COLour*, since you ask), but wouldn't it be nice if you could devise your own vocabulary?

Of course, you'd have to explain what all the words meant, which might be a chore, but not in **Forth**, another language shortly to become available for the Einstein.

Because of the fact that apart from a Basic vocabulary, all the other words used are defined by the programmer, it's a transportable language, and any Forth program can be typed into any computer capable of running the language. This, of course, is very valuable: anyone who's had any experience of typing in Basic programs from books or magazines will tell you of the frustration awaiting you if your computer just doesn't understand what you're talking about.

Another big advantage of Forth is that it's *F-A-S-T*! A Basic program will take several seconds to count up to ten thousand – indeed, a counting loop is often put into a program for that reason, to hold a display on the screen so you can read it before it's cleared with the CLS clear screen command, for instance. In contrast, Forth will count about eight times as quickly.

Two versions of Forth are to be available for the Einstein.

Kuma, the well-known software company, has an implementation selling at £46, while Einsoft, Tatung's own software branch, will be offering what's called Superforth. We'll be comparing the two in future issues.

Pascal

Another very powerful language for mainframe computers was (is) Algol (*ALGOrithmic Language*) devised as a sort of European response to Cobol, and in the 1960s Nicklaus Wirth at the Zurich Technical Institute drew upon Algol as the inspiration for **Pascal**, the language he named in honour of Blaise Pascal, the 17th Century theologian and inventor who devised the world's first mechanical calculator in 1647.

Pascal and Forth programs may look similar to the beginner, but they couldn't be more different.

Pascal is what is known as a *compiled* language, which means that it operates much faster than an *interpreted* language like Basic. But it's supposed to be fairly hard to learn, because of its complex vocabulary.

It encourages structured programming, however, which is A Good Thing.

Pascal for the Einstein has been devised by a very eminent British company called Hisoft, and distributed by Kuma.

DR Logo

According to Setmour Papert, who invented it, **Logo** is more than just a programming language; it's a whole "philosophy of education". But, as the article elsewhere in this issue explains, it also comes closest to giving the computer artificial intelligence.

Yet it's particularly suitable for children, not merely for teaching them how to handle computers, but also for helping them to acquire logical ways of thinking.

The news that Digital Research's version (the DR in the name stands for the company; it is not short for doctor) is being "bundled" with Einstein computers is perhaps the most exciting language announcement of all.

Other languages

The versatility of the Einstein means that Dr Roy Clarke's Bradford laboratory has already been able to adapt other languages, like the UNIX-orientated C language and Fortran, to run on the machine, though at the time of writing no commercial implementations are available.

Though not, strictly speaking, a language at all but an operating system, CP/M (which stands for Control Program/Monitor) has built up an incredible power in the market-place since it was devised in 1976, again by Digital Research.

The Einstein can handle CP/M programs, but since many CP/M programs are written for a screen 80 characters wide, it may be necessary to fit the new 80-column screen card before you can run them. They will, of course, have to be obtained on 3in disks.

Machine code & assembler

At the beginning of this review of available languages for the Einstein, I referred briefly to machine language. It is possible to write programs in this from MOS level, and there are instructions on how to do this in the DOS manual.

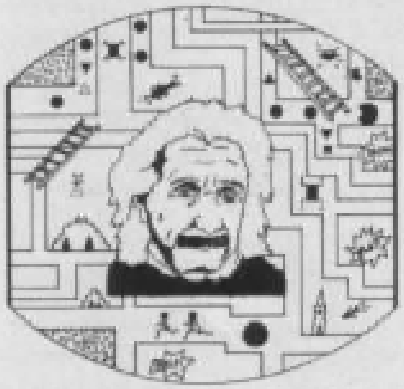
However, as I've said, it's difficult, and if you make the tiniest mistake, not only will the program not run, but detecting what you did wrong is virtually impossible without hours of checking. Since the machine will probably hang up, you may have to turn off without finding out what you did (so it's even more important to SAVE what you've done with ML before you attempt to RUN it than with Basic programs).

However, all is not lost, because you can use an assembler, which uses something closer to English than the ones and zeroes of ML.

There's a review of an assembler for the Einstein on page 10 of this issue.

THE OFF-DUTY EINSTEIN

Franklyn Davies' Gameplay



The Einstein is a serious computer. But that doesn't mean you can't have fun with it.

In fact, alongside the utilities and the languages and the word processors and spreadsheets, there are a remarkable number of games for the off-duty Einstein: nearly 20 at the last count, and still rising. The speed of the CPU, the excellence of the graphics and the sophistication of the sound commands available mean that games on the Einstein should outplay the versions for more primitive machines.

One thing that was notable from the first batch we had in for review was that few programmers are making full use of the Einstein's sound capabilities, but doubtless that will change as programmers move from adapting old favourites like "Pakman" to the new configuration and start using its capacity more creativity.

In reviewing the games, we have marked them out of five for the following characteristics:

Ease of loading
Instructions
Playability
Difficulty
Use of graphics
Use of sound

from which we've been able to owrk out an overall average.

Where graphics and sound are not relevant, for instance in a text adventure, the overall figure may be higher than seems reasonable. But then you will have to decide whether you're into that kind of game or not.

How easily do they LOAD? Can they boot themselves from switch-on, or do you have to LOAD XBAS first? It's a nuisance to have to find a system disk when all you want to do is to play a game or two.

Is the music irritating and distracting; if so, it is possible to turn it off without hitting the volume control?

Another area where some improvement is possible is the allocation of control keys. These should either be mnemonics (U=up, D=down, L=left, R=right, F=fire), or grouped in some logical pattern that falls naturally under the fingers:

```
  W                 I
A-S-D      or     J-K-L
  Z                 M
```

(depending on whether you're left-handed or right-handed), or grouped under the left or right hand so that W/S is up/down and O/P is left/right, for instance.

There should be a means of reconfiguring the keys so you can adopt the pattern you're happiest with.

If so, can you use the non-alphabetical keys (eg < and >)?

Instructions should be clear and concise, and in legible colours. All-capitals is harder to read than upper and lower-case (capitals and small letters, in the vernacular).

Games shouldn't be too easy – unless they're obviously aimed at the very young – but they shouldn't be bloody impossible, either.

In the end, however, what really matters is: are they *FUN*? We guarantee that all these are worth playing. We don't intend to waste precious space tearing the duds to pieces.

How to get addicted

Name	*Castle Quest/ Quest*
Supplier	*Kuma*
Price	*£14.95*
Category	*Adventure*
No of Players	*1*
Verdict	*Good introduction for beginner adventurers*

Rating ■■■■■ is maximum)

Ease of loading	■■■
Instructions	■■■
Playability	■■■
Difficulty	■■■
Use of graphics	*N/A*
Use of sound	*N/A*
Overall	■■■

When you've tired of zapping down insects and ancient Egyptians, and if you've got a lifetime or so to spare,

Pakman

you might like to try getting into adventures. But beware, because they're addictive.

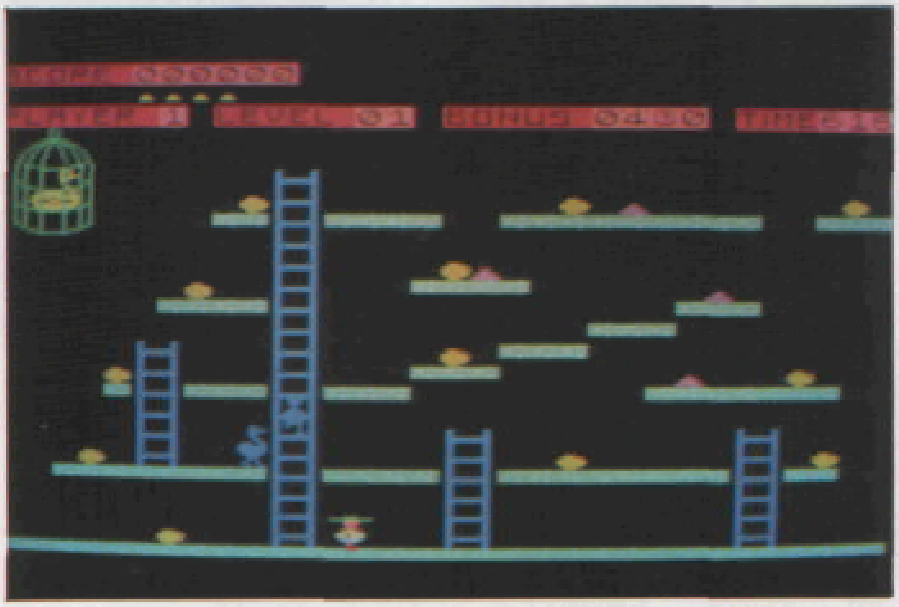
These two games are good introductions for the tyro adventurer, being neither too easy to entertain nor too hard to entice your interest. Experienced adventurers who are used to the thousands of locations possible in the more sophisticated games, which were originally written for mainframes, may scorn the humble 210 rooms in "Castle Quest", but they are a start.

Their complexity may be gauged by the fact that they occupy over 35K of memory each, without using graphics or sound (purist adventurers think graphics and sound use up valuable memory which should be dedicated to the greater bafflement of the players, but data compression techniques have begun to produce some truly staggering effects).

The game is LOADed after XBAS, with the command

RUN"QUEST" ‹ENTER› or RUN"CQUEST"‹ENTER›.

Chuckie Egg



Oh! Mummy

Then come 2½ screens or so of instructions (in tasteful purple on black, surprisingly easy to read), which explain what you have to do. In "Quest", you have to find the Treasure of Borgan and get it over the fiery crevasse, aptly named the Everlasting Fires of Hell, which you have to quench by throwing in the Eye of Morpheus (when you've found it, natch).

In "Castle Quest", you have four treasures to find in the seven levels of the Castle (30 rooms to a level), and the Magic Pass which allows you to rise up from one level to the next, and then back again once you've got the treasure together.

At the outset you are faced with the mesmerising (and eventually totally infuriating) prompt:

WHAT NEXT?

and you have to type in a response, usually in a two-word command (eg "go north"), to which the computer may reply "I don't understand what you mean" if the words aren't in its vocabulary. (Since it's in Basic, and LISTable, you could cheat by checking the DATA statements for possible words.) It may also give a somewhat gnomic response.

For instance, wanting to end the game, I typed "exit" instead of the correct "quit", and got the reply "I don't see anyone to hit", or some such.

The perceptive will notice that there's a program called "Help" on the disk, but to save you the trouble, all its says is to write to the software author, John Wolstencroft, at Sagar Fold, Bleasdale, Preston PR3 1UU, if you get really stuck.

They couldn't be described as classics (like the horribly difficult "Hobbit" or the near-impossible "Hareraiser") but they're certainly a good beginning.

(We shall be returning to the

subject of adventure games in greater detail in future issues. Meanwhile, if any readers have problems with "Quest" or "Castle Quest", or hints on how to proceed, please let us have them and we'll publish them. – ED)

Them chicks is ducks

Name	*Chuckie Egg*
Supplier	*Einsoft*
Price	*£12.95*
Category	*Farmyard arcade game*
No of Players	*1 - 4*
Verdict	*Brilliant*

Rating (■ is maximum)	
Ease of loading	■■■■■
Instructions	■■■
Playability	■■■■
Difficulty	■■■
Use of graphics	■■■■
Use of sound	■
Overall	**■■■**

Game of the month

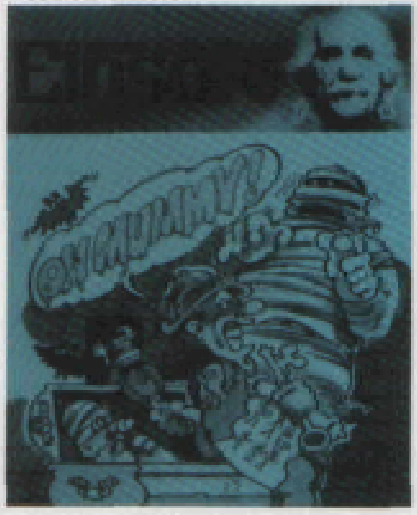
If you're into arcade games – and even if you're not – this is a must, because it is almost perfect, from its ease of LOADing (just reset with the disk in, and it boots itself) to the graphics and general scenario of the game. Even the non-violent will like it, because the hero, Hen House Harry (that's you), has no weapons to defend himself against the aggressive fowl who are trying to peck him to death.

The zoologically pedantic may worry about why a henhouse is full of killer *ducklings* (and an exterminating Mother Duck), but these things happen in computer games!

The music is that sort of one-finger synthesiser you often get in games, but it's easy to turn it off.

You can abandon the game whenever you like, with ‹ESC›+A, "hold" (freeze) it with ‹ESC›+H, restart with S (you are not told this), SAVE the high score with ‹F8› and LOAD it back on a future date with ‹F1›. The control keys are W/S up/down, O/P left/right, and ‹SPACE› jump, but they can be reconfigured any way you like, as long as you stick to the alphabetic keys.

A new classic!



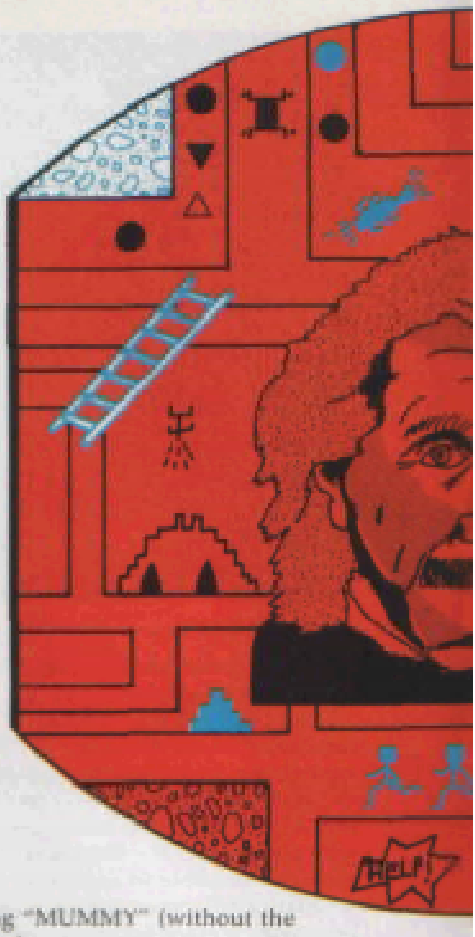
Raiders of the Mummy's Tomb

Name	*Oh! Mummy!*
Supplier	*Einsoft*
Price	*£12.95*
Category	*Ancient Egyptian arcade game*
No of Players	*1*
Verdict	*Mum's no bum*

Rating (■■■■■ is maximum)	
Ease of loading	■■■
Instructions	■
Playability	■■■
Difficulty	■■■
Use of graphics	■■■
Use of sound	■
Overall	**■■■**

We've given this a slightly better overall score than the average of its individual ratings because while the sound (that Wilson, Keppel and Betty da-da-dah-dah-dah sand-dance tune) is a bit naff, and the instructions drone on and on for *nine screens*, it's quite good once it's under way.

There are five difficulty options, and five speed options, and you can turn the music off (but not unless you wade through the instructions). It LOADs from Xta1DOS by just typing "MUMMY" (without the quotes).

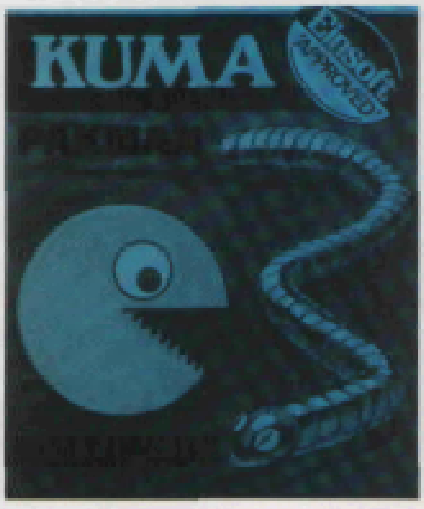


It's a complicated story of uncovering Royal mummies (mummys?) from treasure boxes in five levels of each pyramid. The boxes may contain a Guardian who can kill one of your five-person team of archaeologists unless you have also found a Magic Scroll, which kills him. There's also a sleeping Guardian, however, who may pursue (or, as the instructions have it, persue you) to another level, but not to another pyramid. To get to a new level, you need a key.

The Guardians get harder to evade as you proceed from pyramid to pyramid.

Controls are Q/A up/down and O/P left/right.

Good for the kids, of whatever age.



to other external device like a printer.

The "unscramble" is similar, but only looks at eight Z80 instructions. For instance, if you want to know what the start of Einstein's main loop looks like, type:

u325H‹ENTER›

(the lower case "u" is essential, to avoid confusion with Up).

Though it comes first alphabetically, "Assemble" is actually the final command you'll use, after you've finished putting together your "source code", a text file consisting of the Z80 mnemonics, and want to produce the "object code" in machine language, which is what you actually SAVE (though in this assembler you need to use the command Write). As in "disassemble", you can have it display what it's doing either on the screen, printer or other device. As you might expect, this generates reams of stuff, which you don't have to see if you don't want, because there's a "null" output option. This speeds up the operation considerably.

"Bye" is the next command you'll need, returning you to MOS when everything's done. If you do it in error, don't worry, because you can get back to where you were with GO, and any files or data will still be in memory.

"Copy" moves a block of memory. You have to define the START and STOP parameters and the DESTINATION you want it moved to: the value may be decimal, hexadecimal (with an "H" at the end) or octal ("O" at the end). No suffix assumes a decimal figure.

Some assemblers use line numbers for help in finding your way around (though line numbers really have no meaning away from Basic) but while ZEN uses them, they don't have a great deal of significance to you. Instead, commands like "Up" and "Down" move the appropriate number of lines in the appropriate direction, so U30 will display the line 30 lines above the previous position.

You start typing in your instructions with the command "Enter", which allows you to type in lines of text, each terminated with the ‹ENTER› key, until a full stop is typed in as first character of a line. "Enter" can be used at any time, and will move the previous lines down to make room for the new lines.

Basic-like, but not

"Howbig" lets you know where your ZEN text file starts and ends, and where is the top of memory. "Kill" erases the text file and is the equivalent of NEW in Basic, but "New" in assembler doesn't mean the same thing: it lets you edit the current line.

Likewise, "Locate" has nothing to do with the cursor-positioning command used in some Basics (though not in the Einstein): it is really a "find" to look for the first occurance of a string. (Presumably, they'd have used "Find" if they hadn't needed "Fill" to fill a block of data with a given DATA value.)

"Modify" is like "New", but allows you to define the start address in memory that you want to examine and edit. "Print" displays the next line(s) in memory on the screen, depending how many lines are specified. One line is the default. "Query" is similar, but it displays 64 bytes of memory in HEX and ASCII, from the current line if no parameter is specified.

"Read" is the assembler equivalent of LOAD. "RS" will read an SRC (source) file and append it to the end of any text already in memory. "RC" will read a COM file and the user is prompted for a LOAD address. "RH" will read an Intel Hex file as used in CP/M.

Similarly, "Write" is the equivalent of SAVE, with WS writing an SRC file, WC a COM file, and WH will write an Intel Hex file. Start and stop addresses will be requested in the latter two cases, and also a LOAD address for the Hex file. If ‹ENTER› is pressed, the LOAD address will be the actual address in memory. Finally, an EXECUTION address will be required for a Hex file.

‹CTRL› + ‹STOP› will abort "Read", "Write" and other I/O operations, such as "Out" and "In", which output or input data to or from a specified I/O port.

ZEN is well documented, containing the full Z80 instruction set, an assembly-language listing of ZEN itself, plus an alphabetical index of where all its instructions can be found, and a clear, concise guide to each of its special instructions.

Tool for the skilled

ZEN is a tool, and a good tool, but just as you wouldn't let a kid loose in a Formula 1 racer down the M1, it needs care and understanding if you're to get the full use out of it. The lack of any worked examples may hamper the absolute beginner who wants fast access to some of the more sophisticated facilities of the Einstein, but then that never was its intention.

Some indication of how to drive the sound and graphics firmware of the Einstein via the Z80 might have been useful, however, since the increased speed of ML games is one of the best reasons many users will have for getting into assembler.

One good thing is that you can experiment as much as you like, but as long as you don't Write over the program disk any modifications you might try to ZEN, you can't do any damage either to the assembler or your computer, though you might hang it up, and would have to turn everything off and on to get back to square one.

If you are serious about wanting to get into assembly language, as well as this program you'll need a good guide to the Z80 chip and its instruction set, either Rodney Zaks' 624-page "Programming the Z80" (Sybex, £13.95) or the almost equally-massive, but slightly less forbidding "Z80 Assembly Language Programming", by Lance A. Leventhal (Osborne/McGraw Hill, £13.95).

Conclusions

ZEN Z80 Assembly Language Programming System
(Kuma, £19.95 inc VAT)
Supplied on disk
Manual: 93pp A5, spiral bound

If you're a serious programmer, then ZEN will be an essential utility for you. If you're a beginner, it may be a bit off-putting, but worth the trouble.

Suitable for Intermediate Users

Effectiveness	■■■■
Documentation	■■■
Value for money	■■■■
Overall rating	■■■■

Wonderful	■■■■■
Excellent	■■■■
Very good	■■■
Good	■■
So-so	■

The birth of a new machine

The Roy Clarke interview

Just over two years ago, in May '81," recalls Roy Clarke, "we were asked to develop a home computer by Dr T.S. Lin, who you may know is the chairman of the whole group. And that was it, really. The brief was let's develop a home computer, about as wide as one can get.

"It took us about six to nine months to get a functioning prototype. It was all done here. None of it was done in Taiwan.

"The fact that we've got a Taiwanese parent doesn't mean that we have to buy from Taiwan. Like any other computer, the parts are sourced from anywhere you can get them.

"But I can't say that a particular Einstein will have a particular percentage of British-made or European-made components. We have to get them from wherever we can, because there's a world-wide shortage of semi-conductors right now.

The prototype

"The working prototype, of course, was just a board, not very cosmetic. You have to prove the design. Then you have to lay it out on to a board and see if there are any snags in that layout. The hardware had to be designed to optimise the software. So obviously things were altered.

"A number of prototypes were made, and then we started to tote these around the software houses. They were also shown at the main trade show this year. This time was also usefully spent in ironing out any bugs. Not that we found any major problems, nonetheless anything we found had to be cured.

"One of the biggest difficulties we had was getting manuals produced. Even allowing for the great deal of effort that went into those three manuals, there's probably still some room for improvement."

Dr Clarke with the prototype Einstein board

As you talk to Roy Clarke, it becomes obvious that not only is he proud of his new baby with the enormous IQ, but that he has very pronounced ideas about the way it should develop. But a lot of those ideas arose from the way he was personally involved in the market research that lies behind the Einstein concept.

Typically, he found a unique way of getting some of his data: the radio ham networks.

Enter the hams

"Why does a radio ham not have a computer? It might sound silly, but that's a very valid question, because they're quite used to spending large sums of money. A thousand pounds isn't unusual, and typically £400 to £500. Why weren't they buying them? One thing was they didn't do what they wanted them to do. They wanted them to be peripheral-orientated, and by the time you've bought a cheapie and its peripherals it's more expensive than perhaps it needs to be.

If anyone doubts the English ancestry of the Einstein they need only to talk for a few minutes to Dr Roy Clarke, the ebullient, Bristol-born and Cheshire-raised technical manager of the computer division of Tatung (UK), operating out of the laboratory in Bradford that produced it, and he'll soon disabuse them of any idea that it might be an Oriental wolf in good Yorkshire tweeds.

"The other point was that tape is a pain. We considered things like having intelligent tape handling. You could have a logic-controlled cassette mechanism so you could wind forwards and wind backwards, you'd get search facilities, like a version of the Microdrive.

"And while we were thinking about this, the cost of the mechanisms and so on, we heard about the 3in Hungarian disk drive, which was very competitive with the logic-controlled tape system we'd been looking at. We also got involved in the very front end of compact floppy disk technology, and in the end we went for 3in.

"One reason we couldn't use 3½in was because we could not even now be guaranteed the volume of 3½ drives we would want. Certainly none of the 3½in disk manufacturers would commit themselves to any delivery for the middle of this year in the volumes that we were talking about.

"But the biggest reason is that the 3½in drive is much more expensive than the 3in drive.

"So we built in a 3in drive, and that obviated the need for tape. We've been asked why we don't put a tape interface in and we could have a cheap software base. But file handling can't be realised on a tape, so that's one set of software you couldn't do.

"And secondly, you've still got the reliability problem, and it would add to the cost of the machine to put in a tape interface that people don't really need.

All users

"What you see was the result of asking people what they wanted, of being users ourselves. One of our team had a BBC, another had a *Newbrain*, and I had an Apple and a Sharp, so we've all suffered under tape-based systems, and we all hate them. The result was the specification for Einstein, which was broadly as you now see it."

What isn't mentioned there, though, is the effect of Dr Clarke's own background, which tends to be towards communications rather than computers as such. He describes his education as being "through the Open University and industry".

"I guess most of us about here had done it the hard way, because there just wasn't anything around. Of course it's my hobby. That's probably the biggest single factor. I bought one of the early Apples and it's in a fair old state now."

Hard software

Obtaining the software for the Einstein was as hard as designing the hardware, he found.

"Being a Sharp user, I'd already consigned Sharp Basic to the bin as being not worthy of serious consideration and bought Xtal Basic II, from Crystal Research of Torquay, which is quite a good language. It's got most of the common core of Microsoft Basic and a lot of extra features which are good for hackers and enthusiasts.

"We drew up the enhancements to that language that we would require to drive all our display and all our machine-dependent peripherals. We approached them, and they said they'd already done a lot of the enhancements to the Basic core in a language called Xtal Basic III. We used it as a basis and asked them to write all the enhancements for the graphics, the sound, the I/O handling and everything that we would need to make Einstein what it is today.

"We've now got a language which I believe is probably the most powerful Basic around. It's certainly a lot easier to drive than BBC Basic in many respects.

A friendly system

"We also looked for a disk operating system. We approached various operating system houses, but most were unhelpful. We also looked at CP/M look-alikes, and at Crystal Research's XtalDOS, which was in its early infancy at the time, I suppose. The subsequent product was very much a collaborative effort between Crystal Research and Tatung and the result was a

CP/M-compatible disk operating system, Tatung/XtalDOS.

"But the thought of giving CP/M to the first-time user didn't appeal to me at all: 'BDOS ERROR ON A'! So we did the error-trapping a little bit better than CP/M, we made it friendly."

Dr Clarke is justifiably proud of the fact that Einstein enhancements, like the newly-announced 80-column monochrome card, are on course.

"We've already announced that we will do an 80-column *colour* card. Of course it will have its own RAM, it won't take the main RAM. It will be a very powerful beastie, and it allows you to do lots and lots of goodies.

"One of the limitations of the Texas chip we now use is that you can only have text in two colours, one foreground and one background. That's what the MSX machines do. We are able to have multi-coloured 40-column text and mix it with graphics, something that MSX can't do. We've found a way of doing it."

Man-machine interface

Interestingly, Roy Clarke doesn't approve of the unnecessary use of high resolution, possibly because of his communications background. He points out that the Einstein display is optimised to get the best out of a domestic TV and says that there is no point in producing a display resolution on the computer which the TV screen is incapable of displaying.

At the time of the beginnings of Prestel, he was investigating man-machine interfacing, trying to make people's lives easier. "In fact," he says, "specifications for Teletext and Viewdata were designed to get the best text display on a domestic TV set.

"You know, the problem still is that people are going around devising very nice systems, but they're not very friendly. The sort of problems one finds are how to present information, use of colour – people say you need high resolution, but as soon as you do that you may put the cost beyond people's reach, because the technology is very expensive.

"What do people actually want? Prestel is quite a good way of presenting information. Alright, so the British Isles on Prestel is not a factual representation of Britain, but we're not trying to send out manufacturing drawings to make the British Isles. The information's still there, you recognise it, therefore you've got the information across. That's all that it needed.

The cost of information

"A lot of people are driving it the wrong way. Look at the Ameri-

Continued on page 21

Inside the Einstein

by Dr Roy Clarke

In many respects one home computer is very much like another. Its component parts are similar insofar as there is a need for a central processing unit, (CPU), a memory, storage capacity, and input and output facilities. What differentiates one machine from the other is the way in which the package of component parts is selected and assembled in order to give the user the best value for money.

Einstein was conceived and developed for a specific sector of the market embracing the enlightened home computer user wanting to make greater use of his new-found interest, the educationalist, and the small business user.

Every parameter in the equation was carefully considered and evaluated in relation to this key objective with the aim of providing a complete package containing every ingredient crucial to the requirement, at minimum cost.

CPU/memory

The heart of the computer is its processor and we chose the popular 8-bit Z80A because it is well-proven, well-liked and well-understood. Additionally there is a wide range of books about the Z80, and there is a considerable wealth of machine code programs for users to draw from.

Thus, the potential user can derive maximum benefit from Einstein. This is of particular advantage to the hobbyist, professional programmer and educationalist alike.

Although there are faster microprocessors on the market, we felt that the additional speed was not warranted, the resulting savings being better invested elsewhere in the design.

There are two common limitations with home computer memory. One is its small size, and the other is that the display takes up some of that already small memory. Einstein's advanced display generator has its own 16k of memory, so that full use can be made of the extensive colour and graphics, while allowing full access to the 64k RAM for programs.

A unique technique for the machine's operating system, stored in 8k Bytes of ROM, also allows full use of the 64k RAM. Provision is made internally to allow the ROM to be expanded to 32k Bytes.

Text and graphics

Einstein can generate 56 alphanumeric characters and 160 graphics symbols, all of which can be reprogrammed by the user.

We chose 40 columns and 24 rows for the text, which is the standard chosen by broadcasting and telecommunication authorities for Teletext and Viewdata. This standard was established to get the best text display on a normal domestic colour TV tube.

Text can also be displayed in 32 columns and 16 colours.

Since graphics are becoming an indispensible aid to the small business user and achieving increasing popularity with the home user, we decided that sophisticated graphics and a colour display were an essential part of the new microcomputer.

Einstein uses the Texas Instruments display processor chip which currently leads the field in providing 33 different display planes on which to program up to 32 sprites, groups of picture elements which improve the speed and ease of animation.

The graphics display has a resolution of 256 pixels horizontally and 192 pixels vertically, with 16 colours.

Full powerful text, sprite and graphics handling is available from Tatung/Xtal Basic 4.

Keyboard

The aim was to provide a professional, typewriter-style keyboard so that the typist would feel equally at home at Einstein's console as at the office machine. Experienced users and newcomers alike will find the Einstein keyboard a welcome change from the rubbery feel often experienced with less expensive computers.

Disk Operating System (DOS) Commands/Utilities

DIR – directory	GO – jump to 0100H	PSW – password
DISP – display	LOAD	REN – rename
DRIVE – select drive	LOCK	SAVE
ERA – erase	MOS – go to MOS	UNLOCK

Machine Operating System (MOS) Commands

A –	arithmetic command – displays sum, difference and relative jumps.
B –	selects the baud rate for the RS232-C between 75 and 9600 bauds.
C –	copies a block of memory from one location to another.
D –	converts from hexadecimal to decimal notation.
E –	execute command
F –	fill a block of memory
G –	go to specified location and execute the program there
H –	converts from decimal to hexadecimal notation
M –	modify memory contents
R –	read a block of data from disk
T –	tabulate memory
W –	write to disk
X –	"cold" start to program
Y –	"warm" start to program
Z –	inspect Z80 registers

The Einstein keyboard has 67 keys, comprising graphics keys, eight user-programmable special function keys giving 16 functions for the user's own specific applications, and 11 "control" and cursor movement keys.

Program storage

Many lower-priced home computers use the least expensive means of storing programs – the tape cassette. Putting aside the possible inconsistency in quality between different makes of cassette tape recorder – not necessarily related to price – the more serious limitation is the slow and cumbersome way of locating data.

While a certain amount of delay in loading programs might, initially, not concern home users, they soon realise the limitations of tape and, from the outset, tape would certainly not satisfy the small business user to whom random access filing is normal practice.

By far the best method of program storage is the disk drive where the data, carried in concentric tracks on a flat circular disk, can be instantly accessed. While disk drive is much faster than tape, the advantage carries a cost premium.

Consequently, most manufacturers of the lower priced microcomputers who opt for disk drive offer these as peripheral equipment in their own containers with their own power supply. The additional charge for this 'extra' converts a relatively inexpensive machine into an expensive one.

Our choice was a built-in disk drive, thereby avoiding the extra cost of a separate unit with its own power supply.

Although we believe that the average home user will need only one disk drive, there is simple provision in Einstein for a second to be fitted inside the machine.

We have chosen the Teac super-silent 3-inch compact floppy disk drives. The disks are housed in rugged 'damage proof' hard plastic covers. They are small, easy to store and need not be as carefully handled as the conventional floppy disk. The construction of the disk housing makes it virtually foolproof.

These 3-inch compact floppy disks are already widely available from leading High Street software/computer stockists but, in order to avoid any possibility of the Einstein user buying sub-standard products, certified diskettes under the Tatung label will be made available from all Einstein stockists.

The disks have a capacity of 250k bytes per side, and can be flipped over, giving a total capacity of 500k ▶

bytes per disk. Einstein compact floppy disks probably represent the lowest cost-per-bit when compared with other disk and tape loop systems and we believe that the 3-inch compact floppy disk will become the standard for the home/hobby and small business user.

Operating system/language

Einstein uses a disc operating system specially developed by Crystal Research Ltd known as Tatung/Xtal DOS. Unlike many other disc operating systems, Tatung/Xtal DOS has been developed with the first-time user in mind, providing powerful facilities and utilities, but having clear, easy to understand commands and meaningful error messages.

For the business user, Tatung/Xtal DOS already has the ability to run CP/M programs. Tatung/Xtal DOS provides for fast sequential and random access file handling and will handle compressed, secret and ASCII files. It also allows full interaction between all files and input/output ports.

Einstein's Basic is Tatung/Xtal Basic 4. A powerful, enhanced version of the widely-accepted Xtal Basic 3, developed by Crystal Research, it provides over 190 powerful commands and functions, many of which provide for powerful graphics, sprite and sound handling ability.

The interpreter has been specifically designed to allow the user to easily extend the reserved word and error message list feature. We believe this to be unique to Xtal Basic.

Additionally the Tatung/Xtal Basic probably has the most extensive screen editor found in any home micro.

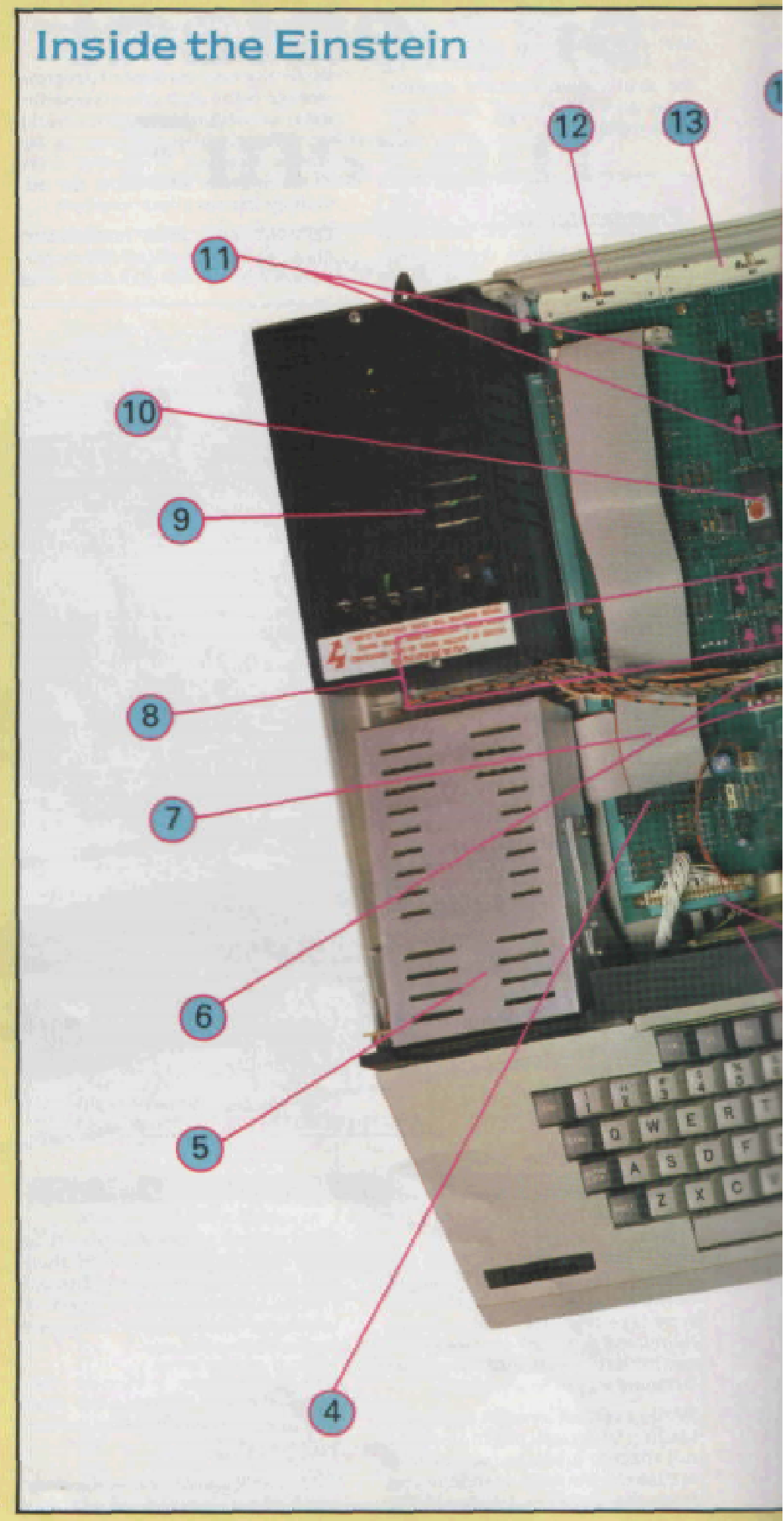
Input-output

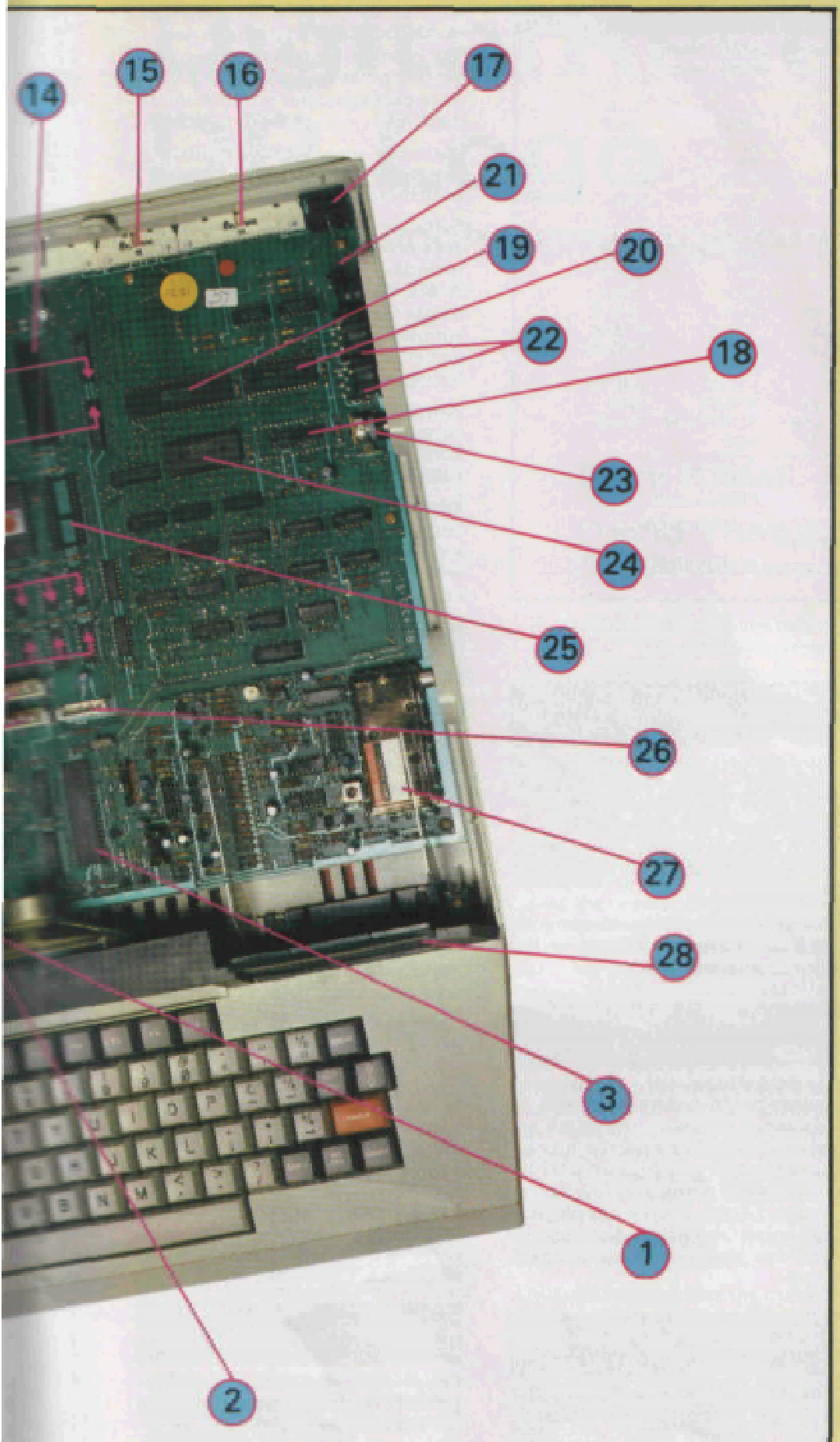
Another very important feature of Einstein is its wide range of integral input-output facilities.

For two-way general data communication, a serial RS232-C port, often an extra on other microcomputers, allows connection to other computers or connection via a modem to a telephone link for Viewdata.

Einstein also includes a high speed ▶

Inside the Einstein





Picture: Jerry Mason

1. Keyboard connector
2. Loudspeaker
3. Video display generator
4. Sound generator
5. 3in compact floppy disk drive
6. Power connectors
7. Disk drive ribbon cable
8. 64K RAM
9. Switched mode power supply
10. System firmware ROM
11. Buffers
12. External disk connector
13. Tatung 'Pipe'
14. Z80 microprocessor
15. User port
16. Printer port
17. Monitor output socket
18. Analogue to digital converter
19. Parallel interface chip
20. Serial interface chip
21. RS232-C serial port
22. Analogue inputs
23. Volume control
24. Counter/timer
25. Spare ROM position
26. Power connector for second disk drive
27. UHF modulator
28. Position for second internal disk drive

Main Basic keywords

ABS	DIM	INCH	MUSIC	READ	SWAP
ADC	DIR	INCH$	NEW	REM	TAB
AND	DOKE	INP	NEXT	REN	TAN
APPEND	DOS	INPUT	NOT	RENUM	TCOL
ASC	DRAW	INPUT#	NULL	RESTORE	TEMPO
ATN	DRIVE	INT	OFF	RETURN	THEN
AUTO	ELLIPSE	IOM	ON	RIGHT$	TI$
BCOL	ELSE	KBD	OPEN	RND	TO
BEEP	END	KBD$	OR	RST	UNLOCK
BIN$	EOF	KEY	ORIGIN	RUN	UNPLOT
BTN	ERA	LEFT$	OUT	SAVE	VAL
CALL	ERR	LEN	PEEK	SCRN$	VDEEK
CHAIN	ERL	LET	PI	SEP	VDOKE
CHR$	EVAL	LIST	PLOT	SGN	VERIFY
CLEAR	EXP	LISTP	POINT	SHAPE	VOICE
CLOSE	FILL	LN	POKE	SIN	VPEEK
CLS	FMT	LOAD	POLY	SIZE	VPOKE
CONT	FN	LOCK	POP	SPC	WAIT
COS	FOR	LOG	POS	SPEED	WIDTH
CREATE	GCOL	MAG	PRINT	SPRITE	XOR
DATA	GOSUB	MGE	PRINT#	SPRITE OFF	ZONE
DEEK	GOTO	MID$	PROT	SQR	
DEF	HEX$	MOD	PSG	STEP	
DEG	HOLD	MON	PTR	STOP	
DEL	IF	MUL$	RAD	STR$	

Note: The list does not take into account combinations of keywords to generate other commands. PRINT @ for example.

There are over 162 Basic commands/functions and 30 MOS/DOS commands.

4-channel, 8-bit resolution analogue-to-digital converter, so that continuously variable commands, as for example from a pair of joysticks, strain gauges or a pyrometer, can be handled. Barcode wands and graphics tablets can also be connected.

A parallel output to Centronics interface standard permits other ancillaries, such as a printer, to be connected. There is also an 8-bit user port, with two "control" signals.

There are two outputs to the display unit. The main output is an analogue video YUV signal chosen in preference to the more conventional analogue RGB signal because it permits adjustment of colour saturation. The YUV signal can, if required, be changed to RGB plus separate sync by means of internal links.

A UHF output is also provided, allowing Einstein to be connected to a domestic television receiver.

All the Z80A signals are buffered to TTL levels and wired to a 60-way connector. We call this the Tatung Pipe – a high speed, powerful and universal concept in microcomputer expansion, which allows for the addition of a wide range of peripherals.

We consider that Einstein has the most comprehensive range of built-in input-output facilties. Many computers only provide these facilities by means of expensive add-on peripherals. Provision is also made for the addition of 2 external disk drives.

Sound

Einstein uses the popular GI sound generator which has 3 channels, plus noise. Each channel can be independently programmed in both pitch and amplitude, providing a wide variety of music and sound effects.

Powerful, easy-to-use-commands in Basic allow full control of the sound generator.

Power supply

Einstein uses the well-proven switch-mode technology used in our range of colour television receivers. This has ensured a cool-running unit with high reliability. Extensive filtering and care in the design of the power supply has minimised the risk of data corruption due to mains-borne interference.

Software

The true merit of any computer must be proportional to its software support. The computer maker has the option of commissioning specially written software packages, but it is far better if the software suppliers themselves provide unsolicited programs which they are perfectly prepared to do if, first, the computer looks like being a bestseller or, second, the computer accepts existing software packages after minimal modification.

We have ensured that software support will come as a result of both these incentives.

Software houses that are already supporting Einstein with software include: Crystal Research; Kuma; A & F Software; Leasalink Viewdata; Gemsoftware and Microsimplex. The range covers home games, education, small business, hobby and specialised uses.

A wide range of languages can be supported by Einstein, such as Forth, DR Logo, Pascal, CBasic, CIS Cobol, Fortran and Assembly.

We believe that Einstein is being launched with the greatest software support ever, support that will rapidly be extended in all areas.

Colour display unit

Our 14-inch colour display unit is designed to suit a variety of computers in addition to Einstein. Besides YUV, it can accept RGB plus sync, switching automatically between the two. These linear systems anticipate the future use of 256 colours. In addition it has an RGB/TTL input.

For analogue inputs the screen can be switched to green for the pleasing traditional data display made popular by green monochrome tubes.

The display unit can be seated on the top of the computer, in which a shallow well provides two positions, a normal front position and second rear position in which the screen is placed at the optimum viewing distance for long term use of the keyboard.

Considerable thought has gone into the appearance and design of these two matching units with an aesthetically attractive result.

Peripherals

Peripheral equipment includes printers, joysticks, and an external twin-disk drive unit giving, together with two internal drives, 2M Bytes total capacity.

To these will be added other items such as Prestel operation and 80-column colour display.

Digital Research's DR Logo

A user-friendly language comes of age

DR Logo – shortly to be made available to the Einstein user – incorporates the list processing capabilities of LISP with a syntax that can be learned by children. And Logo and LISP share other powerful features, too.

Gary Kildall
Digital Research Inc.
David Thornburg
Innovision

Logo for personal computers has been heralded by some as the beginning of a revolution in computer languages that promises to be as far-reaching as the introduction of the personal computer itself. Yet many people think that Logo is not much more than a graphics language for children. Adding to this confusion is the fact that some commercial implementations of Logo are weak (somewhat akin to a version of English that contained no adjectives).

Because of the confusion surrounding Logo itself, the appearance of a sophisticated version of this language on a professional microcomputer such as the IBM Personal Computer might be expected to raise some eyebrows. The development of a powerful Logo for 16-bit computers such as the IBM PC can change our way of thinking about programming.

In this article we will show what makes Logo truly powerful, what it can be used for, and how Digital Research's DR Logo, with its powerful language, large workspace, and complete program-development environment, sets a new benchmark by which to measure the properties of useful computer languages.

To help you understand the power of Logo, we'll give you some background about the earlier language, LISP. LISP, developed more than 20 years ago by John McCarthy, is overwhelmingly the language of choice for researchers in the field of artificial intelligence.

Unlike many other languages, LISP lets users perform operations on several data types, including numbers, words, and lists. A list can consist of a collection of words, numbers, or lists themselves. Because the names of LISP primitives or procedures are also words, one can write LISP programs that automatically generate other LISP programs. It is the ability to manipulate this type of data that gives LISP its name (LISt Processing).

LISP has been used to explore topics as diverse as image processing, the analysis of natural language, the computer solution of certain types of 'intelligence' tests, and theorem proving. More mundane programs in LISP (such as word processors) have also been created. Viewed from any angle, it is a powerhouse of a language.

DR Logo incorporates the list-processing capabilities of LISP with a syntax that can be learned by children. More than the utility (and beauty and simplicity) of turtle graphics, it is this list-processing capacity that gives it so much power.

Other important characteristics are shared by Logo and LISP. Among these is the ability to extend the language through the creation of procedures that are treated just as if they were part of the language itself. As with some Forth devotees, many Logo enthusiasts see themselves as not writing programs, but as creating new 'words' in Logo tailored to the solution of their particular programming task. While this may appear to be a subtle distinction, it has a tremendous effect on programming style. This style affected the design of Digital Research's Logo in several ways, especially in the debugging and procedure-management tools.

The Power of DR Logo

Before showing what Logo procedures look like, we will list a few of the characteristics of DR Logo. To provide maximum power to the user, we designed the first implementation of DR Logo for the 16-bit IBM Personal Computer. The use of a 16-bit processor greatly increased the amount of workspace available to the user and also yielded a modest speed improvement over 8-bit versions of the language. A DR Logo user with 192K bytes of RAM (random-access read/write memory) has about 10,000 nodes available for use. (See the text box). For comparison, an Apple II user running Apple Logo has only about 2800 free nodes to work with. It goes without saying that sophisticated applications require comparably more workspace

want to see what is bound to the variable rather than the variable name itself. If we had entered

print "friends

we would have seen

friends

on the screen instead.

You can take lists apart in Logo with commands such as first, butfirst, last, and butlast. For example, if we enter

print first :friends

the screen will show

Pam

The command

print butfirst :friends

prints

Roy Pat George

Now that we know a little about lists, let's explore Logo's extensibility by creating a new command in the language.

Suppose you did a lot of work with lists and you found that you would like to rotate a list by moving its first element to the rear end and pushing everything else up front. We can create a word (e.g., rotate) to do this for us. If we had such a procedure, we could make a rotated version of friends by entering

make "neworder rotate :friends

Because Logo doesn't have a primitive called rotate, we can create a procedure with this name that looks like the following:

```
to rotate : list
output sentence butfirst :list first :list
end
```

This procedure accepts a list (denoted by the local variable name :list) and makes a new list starting with all but the first word and then appending the first word to the end of the list. The sentence primitive (or native instruction) is used to assemble a list from two parts. The output command passes the new list back out of the procedure to any procedure that used rotate, or to the command level.

Once defined, Logo procedures are treated just as if they were part of the language's native vocabulary. For example, if you were to enter

print rotate :friends

the list

Roy Pat George Pam

would appear on the screen.

Logo's ability to manipulate lists by taking them apart, adding to them, examining their contents, and altering their order is central to the use of Logo in the creation of knowledge-based programs. For an excellent introduction to the use of lists in the creation of a knowledge "tree" that "sprouts" new nodes as the program gets "smarter", you should read Harold Abelson's discussion of the program animals in his book, "Apple Logo".

In addition to the ability to perform list processing and arithmetic, DR Logo also supports an excellent turtle graphics environment. While much has been written about turtle graphics, especially on its use with children, it is important to understand that turtle graphics are of tremendous value to expert programmers as well.

The power of this graphics environment comes through its description of the shape of an object as a series of incremental steps that create it. Once a procedure describing an object has been written, the object can be displayed at any screen location, orientation, and size without having to tamper with the basic description. For example, the procedure

```
to square :size
repeat 4 [forward :size right 90]
end
```

can be used to create a square at any screen position, angular orientation, or size. To draw a square at a given place, you first instruct the turtle (a cursor that has both position and orientation) to move to a specific x-y coordinate and heading (angle). Next you type square 50, for instance, to create a square with sides 50 units long. This property of turtle graphics procedures, coupled with Logo's capacity to run recursive programs, has allowed the easy exploration of geometrical shapes and their properties.

Programming Tools

DR Logo provides many tools to assist the programmer. While smaller Logo systems can adequately survive with a rudimentary procedure editor, larger Logo environments benefit from some of the extra tools that make program analysis and debugging less tedious. DR Logo's procedure editor allows the use of both uppercase and lowercase letters for programs and data. Two primitives, uppercase and lowercase, allow the conversion of a word from one case to the other.

Also, procedure listings can be indented to make decision branches and nesting easier to see. While not essential to the creation of good programs, such formatted listings are easier to read.

While Logo's syntax generally makes procedures easy to read, it is valuable to have comments appended to certain program lines. This ability is provided in DR Logo, along with the ability to strip these comments from procedures with the noformat primitive if more workspace is needed. If the name or syntax of a Logo primitive or editing command is forgotten, online help is available.

Once procedures are created, DR Logo has several primitives that help show how procedures interact with each other. This is especially important for those Logo enthusiasts who experiment with several coexisting versions of procedures before settling on the final choices.

Most versions of Logo will print the names of resident procedures on receiving the pots command (print out titles). If, in DR Logo, you enter potl, the workspace will be examined for all top-level procedures (those not called by other procedures) and their names will be displayed on the screen.

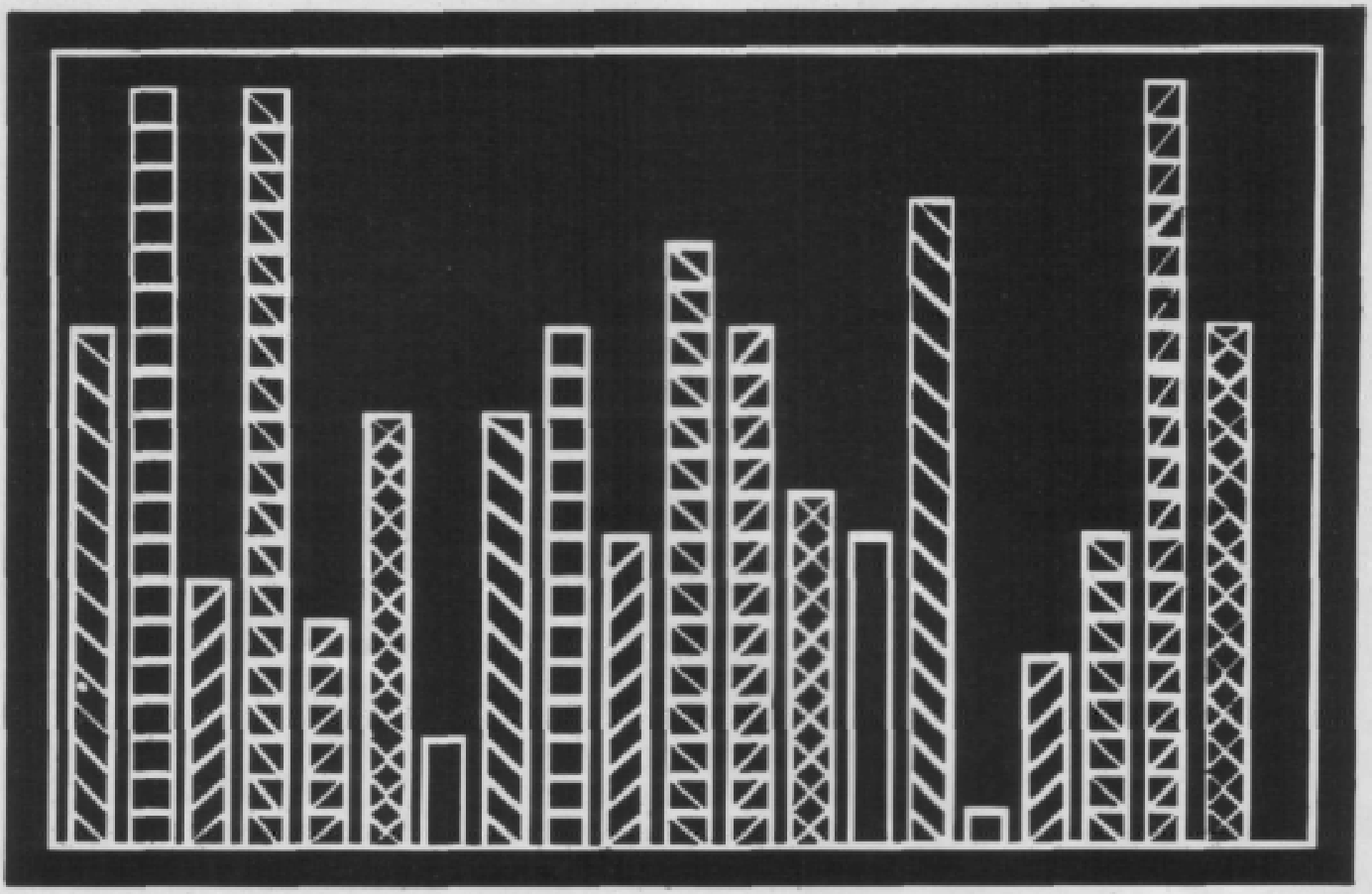
If you enter pocall followed by the name of a procedure, DR Logo will examine the calling structure of the named procedure and print the names of the procedures used by the one mentioned, as well as the names of the procedures used by these secondary procedures, and so on until the calling sequence is complete. This gives a great deal of information on the internal organization of the Logo workspace. If, on the other hand, you enter poref followed by a procedure name, all the procedures that reference this name will be found and displayed.

Many Logo programmers create procedures in a haphazard sequence. Because a listing of multi-

ple procedures follows the sequence in which they were entered, large listings can be hard to assimilate. By using the DR Logo follow command, procedures can be resequenced in any order, thus allowing large listings to be more easily scanned.

Once you are ready to try a Logo program, DR Logo provides additional tools to assist in debugging. One of these tools allows the text screen to be split into windows corresponding to the command level, a user I/O (input/output) port, and the debugger. The trace command traces the procedure and displays what is happening and at what level the procedure is relative to the top (command) level. Because a single recursive procedure (that calls a copy of itself) may oscillate through many levels, knowing the level at which an error occurs is helpful when fixing the fault. The command watch allows single-step execution of a procedure with the ability to change values and see the effect of each statement.

The use of multiple text windows in debugging is only one application for this powerful tool. The development of good window-management tools can, by itself, increase the simplicity, flexibility, and power of this programming environment.



An example of business graphics possible with DR Logo. The main program that produces this picture is in listing 1.

Applying DR Logo in Education

Perhaps because of its historic use as a discovery tool for children (and because of the typically small workspace found with most implementations) Logo is not generally perceived as an applications language. It is anticipated that DR Logo will prove to be an exception in this regard.

The educational applications for Logo have typically focused on the use of turtle graphics. The beauty of turtle graphics is that children simultaneously acquire skills in programming, geometry, and art. Many children who are "turned off" by maths have discovered it to be an exciting field through their exploration with turtle graphics. Furthermore, it has been found that once a child uses Logo to discover new ways of thinking about mathematics, this new way of thinking continues to produce beneficial results – even if the child is no longer exposed to Logo.

In the physical sciences, Logo can be used to construct microworlds in which bodies obey different natural laws, such as gravitation. By exploring these artificial microworlds, children can develop better intuitions about the properties of their own corner of the universe.

DR Logo in language teaching

Given Logo's powerful list-processing capability, one would ex-

pect it to be of value in the language arts as well. To pick one simple example, suppose a child created several lists called nouns, verbs, adjectives, articles, etc., and assigned appropriate words to each list. The word order in each list can be randomized with the shuffle command, and a random sentence can be constructed by assembling words from each list in a syntactically valid order. Legitimate nonsense sentences can be automatically generated in this fashion (e.g., No yellow toad smells tall people.) while bringing the child to look at and solve the structure of English.

The educational value of this program can be seen on several levels. First, if the child creates the lists of words, a misplaced word will show up as a misplaced part of speech. Having a verb appear when a noun is expected results in an obviously invalid sentence structure. The result is a self-reinforcing mechanism for learning the parts of speech. Second, the student can learn to identify valid sentence forms without sample words (a reversal of the traditional parsing process). This helps to cement sentence structure concepts as well. Finally, the student learns some of the challenges awaiting those who want to create natural-language interfaces between people and computers.

DR Logo in Business

While Logo is not usually thought of as a language for business applications, DR Logo has several characteristics that may change this perception. The creation of an interactive illustration generator using an inexpensive graphics tablet is quite easy in DR Logo.

In addition to business graphics, the list-processing capability of DR Logo makes it suitable for database management. In fact, one might envisage incorporating some of the results of research in natural-language understanding to generate a query system that responds to questions such as: "If we increase our salaries by 10 percent this year and increase our sales by 20 percent next month, what will our profit be in the fourth quarter?"

There is no question that many business applications will be found for DR Logo, but it is premature to set limits on the scope of these applications.

DR Logo in Artificial Intelligence

There has been much talk lately about knowledge-based or "expert" systems. The noble efforts of personal computer software experts notwithstanding, sophisticated microcomputer programs that can adapt to various queries are few and far between. The major reason for this is the inadequacy of most computer languages for dealing with the types of data and operations natural to adaptive systems. Because of DR Logo's close connection with LISP, we expect to see artificial-intelligence techniques appearing in personal computer software rather than being limited to university and large industrial research centres as they have in the past.

This movement is valuable for several reasons. First, it will help to demystify artificial-intelligence research. Second, it will result in the application of advances in artificial intelligence to the development of practical programs. To pick one example, suppose you had a computer program (called car repair) that allowed the following dialogue:

User: I hear noises when I steer the car.

Computer: Do you think the problem is in your steering mechanism?

User: Yes, I think so.

Computer: Do you have power steering?

User: Yes.

Computer: Is the noise loudest when you turn the steering wheel?

User: Yes, but I hear it when the car is idling too.

Computer: You should check the level of your steering fluid before proceed-

Logo, Turtles and Kids

Anyone who has watched the personal computer industry for the past few years has probably seen the evolution of certain myths regarding computer languages. Many devotees of Basic, for example, claim that it is the optimal choice for the home user because of its nearly universal adoption as the default language for personal computers. The fact the Basic was the only high-level language that was readily available in compact form in the late 1970s is not considered to be relevant by many users.

Fortunately, the recent availability of other languages on personal computers (Logo, Pascal, Forth, and Pilot, to name but a few) has afforded programmers other choices. But some of these languages have myths of their own.

In the case of Logo, the common myth is that it is a turtle graphics language designed to be used exclusively by children. As evidence in support of this myth, one is pointed to Seymour Papert's book, Mindstorms. It is true that Papert devotes the bulk of his book to the use of turtle graphics as a powerful programming and discovery tool for children, and that he stresses the accessibility of Logo to the young and inexperienced.

The problem with the Logo myth is that it suggests that Logo is exclusively for children's use. As with many myths, the reality of the situation is quite different. First, it is true that Logo supports turtle graphics. In this regard it is similar to some versions of Pascal, Pilot, and Forth. Note also that, while turtle graphics are accessible to children, they also have applications of value to advanced programmers as well. Anyone who doubts this would benefit from reading Turtle Geometry by Abelson and diSessa or Discovering Apple Logo by Thornberg.

The point is that Logo is no more a "kid's" language than is English. Yes, English is the language of "Mary had a little lamb", but it is also the language of Moby Dick and Shakespeare's sonnets.

At its base, Logo is a symbol-manipulation language in the finest sense of the word. Rooted in the artificial-intelligence language LISP, Logo allows the user to extend its vocabulary, to use recursion, and to manipulate various types of data in ways that are nearly impossible with languages like Basic.

It would be a shame if the myth of Logo kept serious programmers from exploring a language whose foundation goes to the heart of computer science itself.

ing. Do you know how to do that?

User: Yes.

Computer: Fine. Check the fluid level. If it is low, fill the reservoir and see if the problem is fixed, otherwise we will continue to explore other causes.

Programs that allow this type of interaction can be used for many diagnostic applications and might be far more valuable applications for home computers than checkbook balancers or recipe files.

Domestic applications for artificial intelligence represent a sleeping giant. The list-processing capability and large workspace of DR Logo will allow this giant to tbe awakened and will enable the creation of a whole new class of applications software.

DR Logo is the first of a new family of languages that promises not only to change our programming style, but to alter the way we think about computing itself. ■

```
to graphics
; A sample business graphics program for bar graphs
make "screen.height 198
make "yfactor .25
make "xfactor .575
make "xdeg 22.5
make "xmin -139
make "xmax 139
make "ymin -79
make "ymax 119
make "return char 13
get.request
end
to get.request
(local "reply "h.or.v "s.or.o "2.or.3)
cleartext
make "reply prompt [Horizontal or vertical bars (h or v)] "char
if :reply = "h
    [make "h.or.v "h]
    [make "h.or.v "v]
if :reply = :return
    [stop]
make "reply prompt [Solid or open bars (s or o)] "char
if :reply = "s
    [make "s.or.o "s]
    [make "s.or.o "o]
if :reply = :return
    [stop]
make "reply prompt [2 or 3 dimensional (2 or 3)] "char
if :reply = 2
    [make "2.or.3 2]
    [make "2.or.3 3]
if :reply = :return
    [stop]
make :reply prompt [Values to be graphed] "list
if "reply = []
    [stop]
bar.graph :h.or.v :s.or.o :2.or.3 :reply
get.request
end

to prompt :text :type
local "reply
(type :text ": char 32)
if :type = "char
    [make "reply readchar print :reply output :reply]
    [output readlist]
end

to bar.graph :h.or.v :s.or.o :2.or.3 :values
cleartext
(local "max.value "min.value "origin "width "depth "axis "reply
    "graph.width "graph.height "proc "spacing)
if emptyp :values
    [stop]
make "max.value 0
make "min.value 999999999
if :h.or.v = "h
    [make "origin list :xmin :ymax make "graph.height :screen.width
    make "graph.width :screen.height make "axis 90]
if :h.or.v = "v
    [make "origin list :xmin :ymin make "graph.height :screen.height
    make "graph.width :screen.width make "axis 0]
if :2.or.3 = 2
    [make "spacing (:graph.width / count :values) * :yfactor]
    [make "spacing (:graph.width / count :values) * :xfactor]
if :2.or.3 = 2
    [make ":width (:graph.width / count :values) * (1 - :yfactor)]
    [make ":width (:graph.width / count :values) * (1 - :xfactor)]
make "depth :width * :xfactor
minmax :values
make "values scale :values :graph.height * .8 / :max.value
cleanup
penup setpos :origin pendown
if :h.or.v = "h
    [line [] list :screen.width your]
    [line [] list xcor :screen.height]
penup setpos :origin pendown
draw.bars :axis :width :spacing :2.or.3 :values
splitscreen
setcursor [0 23]
type [Return to continue]
make "reply readchar
end
```

References

Logo:

1. Abelson, Harold. Apple Logo. Highstown, NJ: BYTE/McGraw-Hill, 1982.
2. Abelson, Harold and Andrea diSessa. Turtle Geometry: The Computer as a Medium for Exploring Mathematics. Cambridge, MA:MIT Press, 1981.
3. BYTE. August 1982.
4. Papert, Seymour. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.
5. Thornburg, David. Discovering Apple Logo: An Invitation to the Art and Pattern of Nature. Reading, MA: Addison-Wesley, 1983.

Artificial Intelligence:

1. Bundy, A., ed. Artificial Intelligence: An Introductory Course. New York: North Holland, 1978.
2. Winston, Patrick. Artificial Intelligence. Reading, MA: Addison-Wesley,1977.

LISP:

1. BYTE: August 1979.
2. McCarthy, John et al. LISP 1.5 Programmers Manual. Cambridge, MA: MIT Press, 1965.
3. Winston, Patrick and Berthold Horn. LISP. Reading, MA: Addison-Wesley, 1981.

```
to minmax :list
if emptyp :list
    [stop]
if first :list > :max.value
    [make "max.value first :list]
if first :list < :min.value
    [make "min.value first :list]
minmax butfirst :list
end

to scale :list :factor
if emptyp :list
    [output []]
output sentence (:factor * first :list) scale butfirst :list :factor
end

to cleanup
hideturtle
setbg 6
penup
home
clean
pendown
end

to draw.bars :axis :width :spacing :2.or.3 :values
if emptyp :values
    [stop]
setheading axis
draw.1.bar :s.or.o :2.or.3 first :values :width :depth :zdeg
setheading :axis + 90
forward :spacing + :width
draw.bars :axis :width :spacing :2.or.3 butfirst :values
end

to draw.1.bar :s.or.o :2.or.3 :height :width :depth :zdeg
(local "origin "direction)
make "origin pos
make "direction heading
if :s.or.o = "o
    [make "proc "open.bar]
    [make "proc "solid.bar]
run (list :proc :height :width)
if 2.or.3 = 2
    [stop]
forward :height
right 90 - :zdeg
forward :depth
right :zdeg
forward :width
right 180 - :zdeg
forward :depth
back :depth
left 90 - :zdeg
forward :height
right 90 - :zdeg
forward :depth
penup setpos :origin pendown
setheading :direction
end

to open.bar :height :width
repeat 2 [forward :height right 90 forward :width right 90]
end
to line :pos1 :pos2
if not emptyp :pos1
    [penup setpos :pos1 pendown]
make "pos1 pos
setheading towards :pos2
forward sqrt sum
    sq ((first :pos1) - (first :pos2))
    sq ((last :pos1) - (last :pos2))
end

to sq :num
output :num * :num
end

to solid.bar :height :width
(local "course "origin)
make "course heading
make "origin pos
repeat :width / 2 [forward :height right 90 forward 1 right 90
    forward :height left 90 penup forward 1 pendown left 90]
if remainder :width 2 = 1
    [forward :height]
penup setpos :origin pendown
setheading :course
end
```

Dr Roy Clarke *Continued from page 13*

cans. They've driven Prestel to beyond its present idea to picture drawing, almost. It's fine. But who cares whether you want to recognise Albert Einstein's left dimple and check if it's in the right place? You don't need that information.

"For example, people say they want 80 columns word processing, but you don't necessarily. You don't need it for a spreadsheet – you window it. It's only a way of thinking. There are very few people who actually need very detailed screens and when you get very detailed screens it actually gets harder to read, you know. You're actually working on the limitations of the human physiology and there's not much point in exceeding that.

"The Einstein was thought out that way. In truth most people who are going to buy it, even small business people, will use it on the telly.

"There's no point in having 80 columns on a TV. Some people say you can. Well, the TV system is inadequate to support an 80-column display, so therefore you're bound to miss information, resulting in a very poor, fuzzy display. That's why you need a dedicated monitor for 80-columns.

"In fact, since you know what the information is, most of the time your brain is putting it in. Back to my philosophy: you don't need to transmit data that is not actually required in the first place. Visual information in the real world contains a lot of data that is not necessary to convey meaning or understanding. This is known as redundancy. So why transmit information that's not needed, since information equals bandwidth equals cost.

"There's no way you can avoid that equation. You've got to get the information rate down, either by transmitting information slowly or by not transmitting redundant information. In reality, you go for a compromise, and then information comes cheap." ■

A SIMPLE SHAPE/ SPRITE EDITOR

by Kevin Salt

If you are anything like me you will be impressed by Einstein's ability to handle sprite graphics, particularly as the commands are so simple! However, the stumbling block always comes when one has to define the shapes one wishes to use.

No matter how quick it is to program sprites, the inevitable hassle of drawing graph-paper type squares, filling in the illuminated pixels, and then converting to hex is still a time-consuming pain in the proverbial.

During the creation of a set of demo programs for a trade exhibition I needed to produce about 200 individual sprite definitions so I decided to write a program to save me a lot of the effort involved.

The program below is a commented version of that program. I apologise for the relatively sloppy programming involved but it was written in about two hours and has not been altered since then (it does its job!). Using the program is easy as long as you remember to write down the displayed definitions before terminating the program.

All the instructions are printed on the screen and the sprite is displayed as it is being built up.

Remember that if you are defining a sprite for use in MAG 0 or MAG 1, you must restrict yourself to the top left box only. If you are defining a printable character for use in 40-column mode you can only use the left hand 6 pixel-columns of the top left hand box.

Any suggestions as to improvements to the basic program (maybe filing the definitions on disk) will be gratefully received and may well be included in forthcoming editions of the magazine.

```
 10 REM ******************************
 20 REM ** A SIMPLE SPRITE EDITOR **
 25 REM ** ====================== **
 30 REM ** WRITTEN BY - K.J.SALT  **
 40 REM ** WRITTEN    - MAY 1984  **
 50 REM ******************************
 90 REM -- always leave room for subroutines
 91 REM -- at the beginning of basic programs
 92 REM
100 GOTO 20000
101 REM
103 REM -- INITIALISATION ROUTINE
104 REM -- ======================
105 REM
106 REM -- set up sprite for display cursor
107 REM
110 SHAPE 255,"0018187E7E181800"
111 REM
112 REM -- blank out normal cursor
113 REM
115 SHAPE 127,"0000000000000000"
116 REM
117 REM -- blank out the characters used for displaying the redefined sprite
118 REM
120 FOR I=0 TO 3:SH$(I)="0000000000000000":SHAPE 144+I,SH$(I):NEXT
121 REM
122 REM -- draw the 'enlarged' sprite boxes
123 REM
130 DRAW 10,56 TO 10,56+&80 TO 10+&80,56+&80 TO 10+&80,56 TO 10,56
140 DRAW 10,56+&40 TO 10+&80,56+&40:DRAW 10+&40,56 TO 10+&40,56+&80
141 REM
142 REM -- draw in the pixel divisions
```

```
143 REM
150 FOR X=0 TO &70 STEP 8
155 IF X=&40 THEN GOTO 180
160 DRAW 10,56+X TO 10+&80,56+X,2
170 DRAW 10+X,56 TO 10+X,56+&80,2
180 NEXT
181 REM
182 REM -- print up the instructions
183 REM
190 PRINT §2,18;"Shape Definitions :-"
200 FOR I=0 TO 3
210 PRINT §2,20+I;"SHAPE X+";I;",";CHR$(34);SH$(I);CHR$(34);
220 NEXT
225 PRINT §24,0;"COMMANDS"
226 PRINT §24,1;"========"
230 PRINT §24,3;"CURSOR MOVEMENT"
240 PRINT §24,4;"GREY ARROW KEYS"
250 PRINT §24,6;"PIXEL ON/OFF"
260 PRINT §24,7;"USE SPACE BAR"
270 PRINT §24,9;"TO EXIT PROGRAM"
280 PRINT §24,10;"HIT <ESC> KEY"
290 PRINT §33,18;"SHAPES"
291 REM
292 REM -- draw in the black box where the actual sized sprite is displayed
293 REM
300 GCOL 1:DRAW 210,10 TO 210,25 TO 225,25 TO 225,10 TO 210,10:FILL 212,12:
    GCOL 6
301 REM
302 REM -- put the actual sprites in the black box
303 REM
310 SPRITE 1,210,26,15,144
320 SPRITE 2,210,10,15,145
330 SPRITE 3,218,26,15,146
340 SPRITE 4,218,10,15,147
341 REM
342 REM -- initialise the edit cursor x & y positions to the top left corner
344 REM
350 X=0:Y=0:RETURN
390 REM ========== END OF INITIALISE ROUTINE ==============================
391 REM
392 REM -- LOCATE THE EDIT CURSOR
393 REM -- ====================
394 REM -- at position x,y on the enlarged grid
395 REM
400 SPRITE 0,10+X*8,56+&80-Y*8,1,255
410 RETURN
411 REM ========== END OF CURSOR LOCATE ROUTINE ============================
412 REM
414 REM -- SPACE BAR HAS BEEN PRESSED !!
415 REM -- ============================
416 REM
417 REM -- first set or reset the 'bit' in SP%() corresponding to
    position x,y
418 REM
450 IF SP%(X,Y)=1 THEN SP%(X,Y)=0:ELSE SP%(X,Y)=1
451 REM
452 REM -- next fill in the corresponding enlarged pixel
453 REM -- (if it is already filled then EINSTEIN automatically 'unfills' it)
454 REM
460 DRAW 12+X*8,54+&80-Y*8 TO 16+X*8,54+&80-Y*8 TO 16+X*8,50+&80-Y*8
    TO 12+X*8,50+&80-Y*8 TO 12+X*8,54+&80-Y*8
465 FILL 14+X*8,52+&80-Y*8
466 REM
467 REM -- find which hex digit we are dealing with (must be one of four)
468 REM
470 J=INT(X/4):REM -- J is the digit position
471 REM
472 REM -- now calculate new value for the digit (i.e. four 'bits')
473 REM
474 DIGIT=0:FOR K=0 TO 3
```

```
  475 IF SP%(K+4*J,Y)=0 THEN 490
  480 DIGIT=DIGIT+2^(3-K)
  490 NEXT
  491 REM
  492 REM -- now convert the decimal value of the digit into a hex number
  493 REM
  500 HX$(Y,J)=HEX$(DIGIT,1)
  501 REM
  502 REM -- find out which part has been affected by this redefinition (YP)
  503 REM -- and then blank out the string containing that sprite definition
  504 REM
  511 YP=0:IF Y>7 THEN YP=YP+1
  512 IF X>7 THEN YP=YP+2:REM SH$()pos
  513 SH$(YP)=""
  514 REM
  515 REM -- unfold the pixel grid (16 * 2) bytes of data into the actual shape
  516 REM -- definitions (8 * 4) bytes of data (you work out how I did it !!)
  518 REM
  519 REM
  520 IF YP>1 THEN INC=2:ELSE INC=0
  530 IF YP MOD 2 <> 0 THEN INX=8:ELSE INX=0
  540 FOR I=0 TO 7
  550 SH$(YP)=SH$(YP)+HX$(I+INX,INC)+HX$(I+INX,INC+1)
  560 NEXT
  561 REM
  562 REM -- now redefine the displayed sprite (the one in the black box)
  563 REM
  565 SHAPE 144+YP,SH$(YP)
  566 REM
  567 REM -- then rewrite the new definition of the sprite for the user
  568 REM
  570 PRINT $15,20+YP;SH$(YP);:RETURN
  571 REM
  572 REM ============ END OF SPACE BAR PRESSED ROUTINE ======================
  573 REM
  600 REM **
  601 REM ** END OF SUBROUTINES **
  602 REM **
19000 REM
19001 REM ************************************************************************
19002 REM **          MAIN DRIVER PROGRAM STARTS AT 20000                       **
19003 REM ************************************************************************
19004 REM
19900 REM -- first set up the colours to be used
19901 REM
20000 BCOL 14:TCOL 6,0:GCOL 6,0:CLS
20001 REM
20002 REM -- set up the 'bit map' (SP%( , )) and the array containing its hex
20003 REM -- equivalent (HX$( , )) and clear them ready to start
20004 REM
20005 DIM SP%(15,15),HX$(15,4)
20006 FOR I=0 TO 15:FOR J=0 TO 15:SP%(I,J)=0:NEXT:NEXT
20007 FOR I=0 TO 15:FOR J=0 TO 4:HX$(I,J)="0":NEXT:NEXT
20010 GOSUB 110:REM -- INITIALISE SYSTEM
20020 REM **** MAIN CONTROL LOOP STARTS HERE ****
20030 GOSUB 400:REM -- LOCATE CURSOR
20040 A=INCH:REM -- GET INSTRUCTION
20050 IF A=4 THEN X=X+1:IF X>15 THEN X=15:REM -- right arrow pressed
20060 IF A=8 THEN X=X-1:IF X<0  THEN X=0:REM -- lileft arrow pressed
20070 IF A=10 THEN Y=Y+1:IF Y>15 THEN Y=15:REM -- down arrow pressed
20080 IF A=11 THEN Y=Y-1:IF Y<0  THEN Y=0:REM -- up arrow pressed
20090 IF A=32 THEN GOSUB 450:REM -- space bar pressed
20100 IF A=27 THEN CLS:GOTO 30000:REM -- escape key pressed
20101 REM
20102 REM -- any other key pressed is invalid so ---
20110 GOTO 20030
20111 REM **** END OF MAIN CONTROL LOOP
29999 REM
29999 REM -- redefine normal cursor before leaving
30000 SHAPE 127,"FFFFFFFFFFFFFFFF":SPRITE OFF:END
```

Everyone knows that the top people shop at Harrods, the world-famous Knightsbridge store. And now, in the newly-refurbished third floor computer, TV and video department, the top micro for top people is the Einstein.

Your Einstein Dealer

Top people like the best

Al Brown, Harrods

Al Brown – everyone calls him "Al" – has been buyer in the department for 12 years, but it's only now that the store has gone into computers in a serious way.

"We fell into it as a sort of carry-on from the TV game era," he said, snatching a brief lunchtime respite from the crowded new department. "We saw the writing on the wall towards the middle of 1983 and transferred all video games to the toy department and started selling micro-computers in a small way towards the end of the year.

"Initially, we had the 16K and 48K versions of the Spectrum, the Commodore Vic-20 and 64, and shortly afterwards the BBC. We made a poor fist of it at first, largely because we were waiting for the whole department to be rebuilt and refurbished, which has now happened."

The department is now spacious, exciting and elegant, with hi-fi, TVs and videoa around the walls, and the micro display in a flurry of activity on an island site in the middle. The computers are spaced far enough apart to allow potential customers to have a hands-on experience, and there was no shortage of triers while I was there.

Full software support is also displayed.

It comes as something of a surprise to learn that Harrods almost didn't join the micro revolution.

"Because the profit margins on computers are small we have to be very careful in how we order," says Al Brown. "We don't look at this just technically, because you can get into a lot of trouble that way, so we decided to go for the big sellers in the market place. Then with the new department, we wanted to go a bit up-market, so we looked at the next stage up from the BBC."

Which was when they started stocking the Einstein?

"Precisely. There's a certain type of customer who comes in wanting the very best, and for them, the Einstein has everything to recommend it. But for the more economically-minded, it's also very appealing as a package product, in my opinion – though remember, I'm not a technically-minded person. I am on the same level as the more naive kind of customer.

"To me, it seems a more professional-looking design than most, and the best value for money around. The price is certainly very attractive.

"I'm expecting a very big demand for the Einstein business package, with the 80-column screen and the green monitor. It appears that its capability and expandability is rather better than the BBC's, and the customer ends up with a much better package for his money."

Despite the rather derogatory way Al Brown refers to his own expertise – and, personally, I don't believe a word of it! – he's certainly a believer in having trained staff to service the needs of his customers. Two of his salespersons have been trained by Tatung, and shortly he is planning to add a third, specially-trained sales assistant to look after the Einstein customers full-time, especially on the business-user front.

"You do need dedicated staff," he exclaims, and who can contradict him?

As we spoke, he had to break off to deal with a delivery from another computer manufacturer, who seemingly was delivering goods completely at variance with what was on the official order.

"It happens all the time," he said, "but not with Tatung. They keep their promises, delivering on time, and looking after us properly. Not like some."

With a range that extends upwards in price to the 16-bit Apricot, and includes Sir Clive Sinclair's new wonder-child, the QL, Al Brown is in a unique position to be able to make comparisons, but he was unwilling to do so.

"The QL is certainly selling well," he said, "but that may be largely because it was advertised so heavily, and then supply kept short, artificially, I suspect. It remains to be seen how well it will do in the long run – though the Spectrum has been a very good, consistent seller.

"Tatung's approach is quite different, and I must say I like it. They announced when it would be available, and they had it out on time, the product performs well, and I expect it, too, to do very well.

"Perhaps better in the long run."

Tell us about your local computer dealer. Let us have your bouquets – and your brickbats – for service. We'll publish the most interesting.

Hints and Tips

by Julian Rangeley

In this first article, which is planned to be a regular feature, we will outline ideas to give you "food for thought" and also try to cover items which are not included in the user manuals.

1. *Function Keys*

The BASIC reference manual refers to embedding a "carriage return" in a statement associated with a function key. This is achieved by holding down the ‹GRAPH› key and then pressing the ‹ENTER› key. This produces a "C R" on the screen at the current cursor position.

All the screen control codes used by Tatung Xtal Basic can be entered into function key statements in the same way. These are summarised in the following table.

As an example, to program function key 1 to clear the screen, type KEY 1, a double quote, then press ‹GRAPH› and ‹CTRL› keys and press and release ‹L› key, followed by double quote. This will appear on the screen as

KEY 1, "F F" where F F indicates Form Feed.

ASCII Code	Function	GRAPH + key(s)	Symbol
0			
1	Screen dump to printer	CTRL–A	S_D
2			
3			
4	Cursor right	← →	→
5			
6	Delete character at cursor	CTRL–F	→/
7	Beep	CTRL–G	b_L
8	Cursor left	SHIFT– ← →	←
9	Horizontal Tab	CTRL–I	h_T
10	Cursor down	↑ ↓	↓
11	Cursor up	SHIFT–↑ ↓	↑
12	Cursor Home & Clear Screen	CTRL	F_F
13	Carriage Return	ENTER	C_R
14	Clear Screen 40 column	CTRL–N	4_0
15	Clear Screen 32 column	CTRL–O	3_2
16			
17	Cursor on	CTRL–Q	C_O
18	Printer on	CTRL–R	P_O
19	Printer off	CTRL–S	P_F
20	Cursor off	CTRL–T	C_F
21	Delete to end of line	CTRL–U	D_L
22	Delete to end of screen	CTRL–V	D_S
23			
24	Delete row	CTRL–X	D_R
25	Delete to left	DEL	→|
26	Insert	SHIFT–INS	■↓■
27	Escape	ESC	E_C
28			
29			
30	Cursor home	CTRL–↑	↑
31			

2. *Ellipse Command*

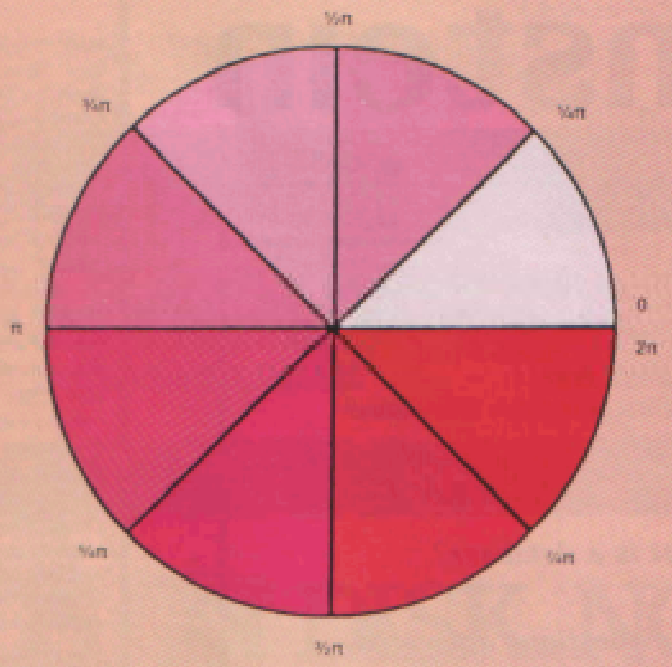
The syntax of the ELLIPSE command is given as ELLIPSE x,y,R,T,z in the Basic reference manual. There are two additional parameters which give the full syntax: ,

ELLIPSE x,y,R,T,z,a,b.

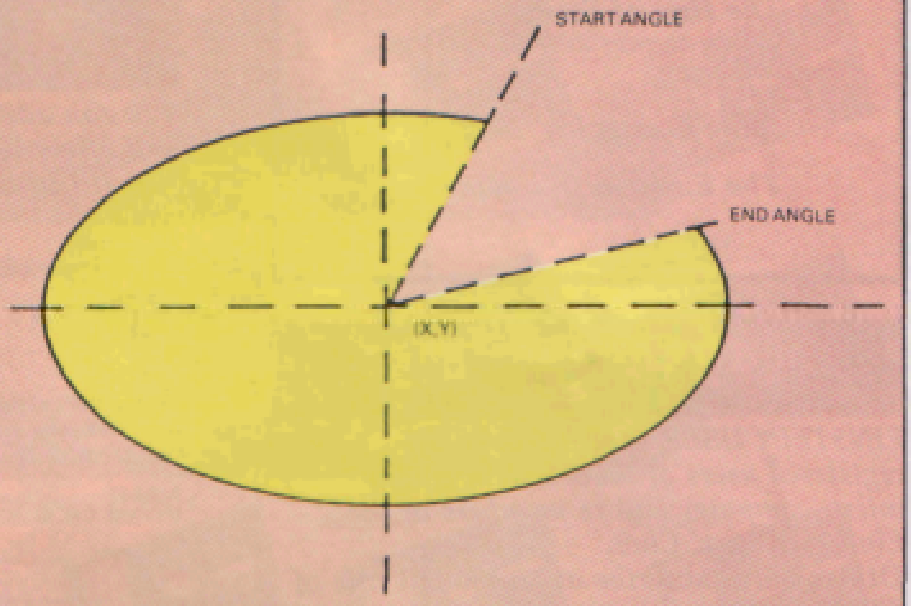
The optional parameters a and b are the start and end angles which indicate where the drawing of the ellipse should begin and end.

The angles are specified in radians and are numbered in an anti-clockwise direction from 0 at the right hand horizontal axis up to 2Pi radians for one complete revolution. The values may be specified as numbers or an expression in terms of Pi.

Specifying a start angle of 1 and an end angle of 0.25, for example, would result in an ellipse similar to the one shown here.

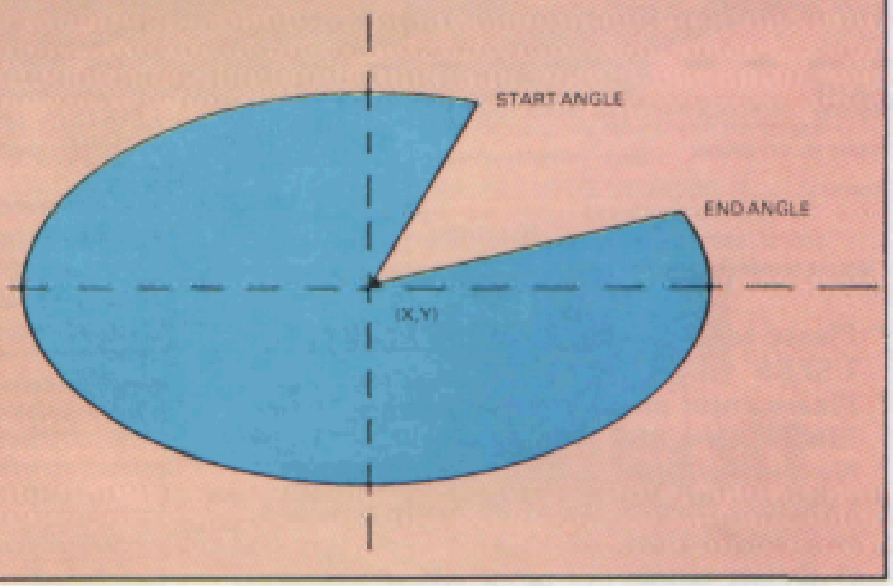


Ellipse 100,100,60,1,0.25



If a or b are given as negative values the start and end points will be joined to the 'x,y' point (centre) with a line.

Ellipse 100,100,60,1,0,-1,0.25



Similar parameters may be added to the POLY command:- more in the next issue!

EINSTEIN USER

Next Issue

It's a cracker
Einstein's superlative spreadsheet reviewed

Fitting a second disk drive

What is a database?

Your questions answered
by Dr Clarke (send 'em in!)

The first of Forth
Getting to grips with the new language

More Gameplay

The Einstein in business

SUBSCRIPTIONS

EINSTEIN USER is sent free to all registered users of the Einstein colour micro-computer and to members of the Einstein Users' Club.
Others may subscribe at an annual rate of £4.80.
Fill in the form below to receive your copy.

NAME .....
(PLEASE USE BLOCK CAPITALS THROUGHOUT)

NAME OF COMPANY .....

ADDRESS .....

..... TELEPHONE .....

* SERIAL NUMBER OF COMPUTER ..... (if applicable)

* Please send me details of the EINSTEIN USERS' CLUB
* Please enrol me as a subscriber to EINSTEIN USER from the
* November/February/May/August issue.
* I enclose cheque for £4.80 made payable to Einstein User.
(* Delete words not required)

About our Contributors

Karl Dallas is a professional journalist who worked for 25 years as a music critic before teaching himself programming during a serious illness in 1979. He has written for Personal Computer News, Personal Computer Games, TV Gamer, Soft, Corporate Computing, MSX User, Commodore User, and Computer and Software Retailing. He has been editor of Commodore User and is software editor of A-Z of Personal Computing. He is author of forthcoming books on MSX, the Amstrad CPC 464 and the Commodore Plus/4 computers.

Gary Kildall is the president of Digital Research Inc. He is active in research and was developer of CP/M, Digital Research's version of PL/1 and DR Logo.

Julian Rangeley is senior development engineer, computer division, at the Tatung (UK) laboratory, Bradford.

Kevin Salt is the North-East Area sales development manager of Tatung (UK). He deals with technical problems.

Bert Stone was born in Germany. He was a slow pupil and started writing on science while working as a Customs officer. His ambition is to be a watchmaker.

David Thornburg is an author and lecturer who has been actively involved in the development and support of user-friendly programming environments. His most recent book, "Discovering Apple Logo", shows how Logo can be used to explore the art and patterns of nature.

FREE MARKET

You can advertise your computer wants and offers FREE!
Just fill in the form below and send it to:

FREE MARKET
Einstein User
28 Gordon Mansions, Torrington Place,
London WC1E 7HF

by December 1 1984

Please note classified advertisements from commercial organisations must be pre-paid at the rate of 10p per word. Make cheques payable to Einstein User.

NAME .....
(PLEASE USE BLOCK CAPITALS THROUGHOUT)

ADDRESS .....

..... TELEPHONE .....

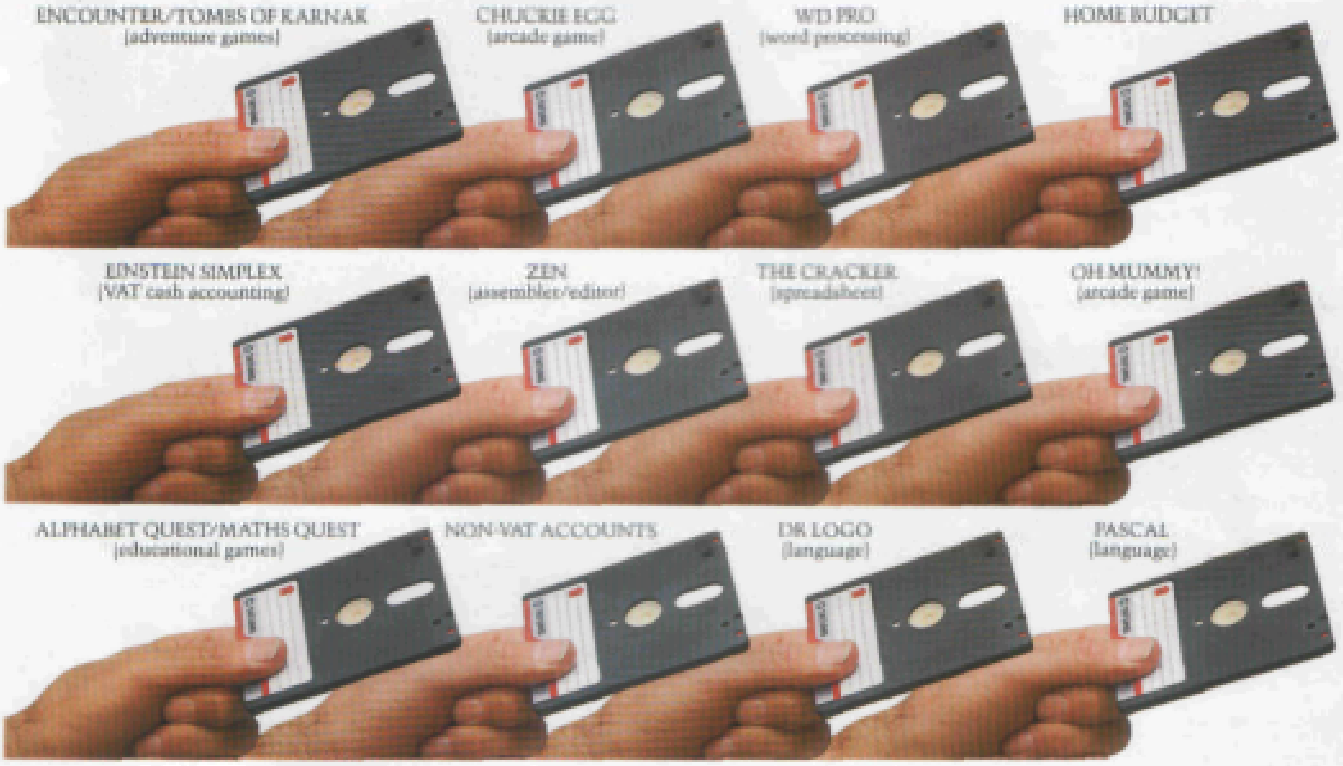
* SERIAL NUMBER OF COMPUTER .....

*Please insert my FREE MARKET advertisement in the *November/February/May/August issue of EINSTEIN USER.*
(Delete words not required)*

If a commercial company, please tick here ☐ *and enclose cheque @ 10p per word.*

TEXT OF ADVERTISEMENT

ALL HANDS ON DISK. NOW.



Einstein from TATUNG. Sheer genius from around £499. The complete colour micro with no hidden extras. And with no hidden software, unlike some other new computers.

Right now you can get your hands on both Einsoft and Einsoft Approved Software. An immediately available range that's every bit as versatile as Einstein itself. "Einsoft" is our own software, designed and prepared for Einstein by TATUNG.

Einsoft 'APPROVED'

Kuma, Crystal Research, Digital Research and Solo are some of the country's leading software houses now writing superb software packages for Einstein, and TATUNG has complete confidence in recommending those packages which bear the "Einsoft Approved" seal.

So from today you can explore the genius of Einstein with software covering a tremendous range of applications. From word processing, database and home economics to education, languages and games. And there's plenty more coming all the time.

Einstein from TATUNG. Sheer genius at home. At work. At school.

And now at your local stockist.

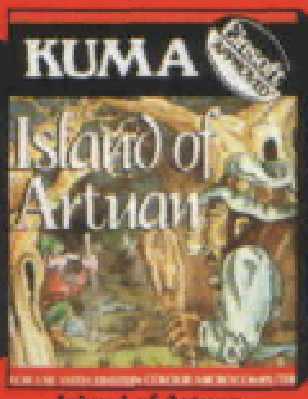
TATUNG Einsoft

DIAL 100 AND ASK FOR FREEFONE EINSTEIN FOR YOUR NEAREST STOCKIST.

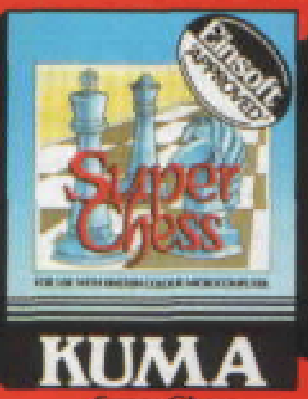
the only choice

Kuma Einstein

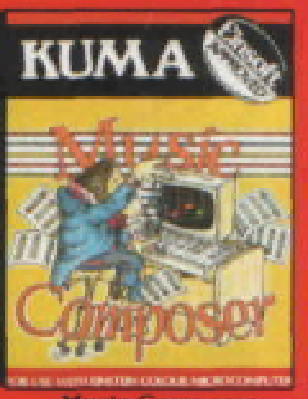
Quest Castle Quest



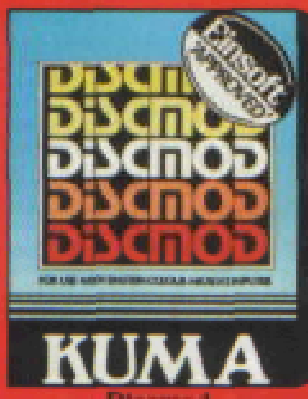
Island of Artuan



Super Chess



Music Composer



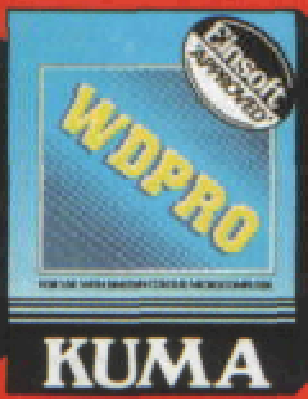
Discmod

Pascal

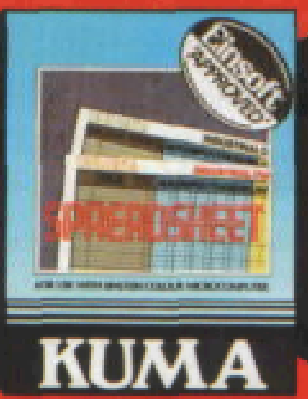
ZEN Assembler

FORTH

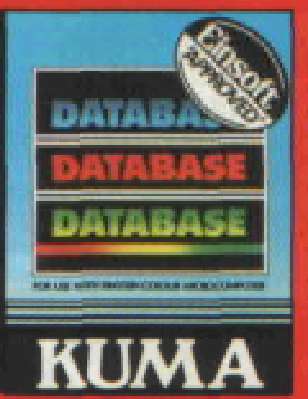
Eastvat



Wordprocessor



Spreadsheet



Database

plus much more.....

Phone 07462-5721 for more details

or contact

TATUNG (U.K.) Ltd., Bridgnorth, Salop, WV15 6BQ

or Kuma Computers Ltd., 12 Horseshoe Park, Horseshoe Road, Pangbourne, Berks. RG8 7JW