memopad

Memotech Computer User Club Magazine

ISSUE:

ALL ABOUT INTERUPTS

TEACH YOURSELF MACHINE CODE

NEW SERIES - 3D GRAPHICS

MTX OWNERS GUIDE TO SOFTWARE

& MUCH MORE!

EDITORIAL

A HAPPY NEW YEAR ! I would like to start off by welcoming the vast number of new members who have joined since the last publication of the magazine. Also, I would like to apologise for the late publication of this edition. However, you will be pleased to hear that I have now finished my book, and I can spend a little more time on the club beore I start my next contract. I would also like to apologise to the members who have written to me on a specific subject, please be patient, I will answer all the letters as soon as possible, but I am finding that there is just not enough time - we receive over 60 letters a day !

I have just heard some exciting news, although it is only in preparation at the moment, it is likely that we shall soon have a utility that will allow us to load and run **SPECTRUM** software ! I will keep you informed, and as soon as it is definite we shall shout it from the roof tops.

Tri-com have not delivered ! Every day I receive at least three phone calls enquiring about this company, and why they haven't fulfulled their orders. It is a really poor show, and although I have heard that a few people have actually got their goods, the majority haven't. If any members are still awaiting a delivery will you please send me a **postcard** stating when you ordered, what you ordered, and the amount you paid - do not put any other correspondance on the card. I will then see if we can take a collective action to make sure you get the goods.

It is apparent that by then end of this month we shall see an abundance of software released. **Brunning** have now released their **Data File** and details or the tape can be obtained **direct from Brunning only,** and we shall carry a full review in the next magazine. A new taped based word processor, and a home accounts program will be released within the next couple of weeks - the word processor is excellent and will retail to members for **#8.95p - don't let the price put you off there is a 30-odd page booklet to go with it.** Sentient have two exciting utilities to be released shortly, and we shall carry full reviews in the next edition. You will see from this magazine that our friends at **Membrain** have produced an excellent **SPOOLER** which allows you to dump your screens to tape and carry on computing while the printer does the work. All in all, there are too many new pieces of software, almost completed, to mention in this editorial, but we will carry reviews and views as they appear.

Memotech are awaiting the arrival of the actual drives for the new cheapo disc units, but they are ready to spring into action as soon as they have taken delivery. A **Graph Pad** is on the horizon, and the end of February should see this on sale. The **Speech Synth.,** will be available within the next three weeks and this will be announced, and reviewed in the next edition.

Please keep your programs coming in, and don't forget we still have a supply of Peter Goode's "Memotech Program Book" at #5.25 inclusive. I have just heard that the new manual will be released within the next four weeks, so all members who ordered before the difficulties, will receive their copies first.

Many thanks for your support in 1984..... let us go forward in 85 (Memotech please take note !)

GENPAT'S TOP SELLER LIST FOR JANUARY

1.	POTHOLE PETE	Continental.
2.	GRAPHICS	Syntax.
3.	AGROVATOR	Syntax.
4.	UTILITIES	Continental.
5.	LORDS OF TIME	Level 9.
6.	QOGO	Continental
7.	EDASM	Syntax.
8.	SNOWBALL	Level 9.
9.	LITTLE DEVILS	Syntax.
10.	3D TACHYON FIGHTER	Continental.

It's a Winner!

WHATS THIS MAN SAYING WINNER

C 000542 MR.L.R SMITH FROM LOWESTOFT.

His caption read:"**Look, I'm telling you! I did manage to get past the first screen of Pothole Pete!!"**

The software will follow.....

The last competition for a **One Line program** really got you on your toes .... entries a still flooding in. We will judge this competition for the next magazine.

THIS MONTH'S COMPETITION

YOU WANT TO SEND A LETTER TO THE CHAIRMAN OF MEMOTECH BUT YOU DON'T WANT ANYONE ELSE TO READ IT. DESIGN A SMALL PROGRAM TO ENCODE YOUR LETTER AND SEND IT TO GENPAT. WRITE A SMALL NOTE ON HOW TO DECODE IT, AND WE SHALL THEN PUBLISH THE BEST FIVE ENTRIES. WE SHALL THEN CHALLENGE THE REST OF THE MEMBERS TO CRACK THE CODE..... THE ENTRY THAT STILL REMAINS A SECRET WILL WIN THE PRIZE. **Let us a have a really good response to this competition. The replies to last month's were marvellous, and it makes us, here at Genpat, pleased to know that we are not wasting our efforts.**

A second competition for the best idea for a competion - if you see what I mean. If we use your idea, you get a prize..... everyone could be a winner !

WE REQUIRE LOTS AND LOTS OF GAMES PROGRAMS FOR PUBLICATION IN THE MAG. I THINK MOST MTX OWNERS WOULD RATHER PLAY THEM THAN WRITE THEM. SEND YOUR SECRET PROJECT IN TO US - ON TAPE PLEASE - HELP US TO HELP THE OTHER MEMBERS. SO DON'T BE ASHAMED OF YOUR PET PROGRAM....SHARE IT WITH ALL THE OTHER MEMBERS.

EDUCATION *The micro in the classroom is more than a microcosm.*

EDUCATIONAL SOFTWARE - WHAT'S THE USE ? by ALAN STURGESS

To merit its name, 'educational software' must either teach or be both useful and effective in supporting a teaching situation or objective. It must help to induce the learning of some concept, skill or fact. In itself it needs to be designed with some specific objective in mind - there is little to be said for writing an 'educational type' program. The programmer is constrained by the same set of parameters as the teacher:

1. His aim is to induce learning.

2. Within the context of the school, this learning should, as far as possible, be:

a) **Structured** - usually with regard to a coherent scheme of work, an exam syllabus, a given objective or the inherent logical structure of the subject being studied.

b) **Meaningful** - in that it relates to what has already been learned, and is likely to promote new and related learning.

c) **Worthwhile** - in that it can be **justified** by reference to aims, objectives, coherence and efficiency.

d) **Effective** - in that it develops or enhances a pupil's level of skill or understanding.

e) **Capable** at some point of being directly or indirectly assessable.

3. The programmer, like the teacher, has to accept that an inevitable consequence of **any** teaching situation is that something **will** have been learned - the question is: will this be something positive, or will it be something negative (such as a dislike of the activity or a failure to see the point of something)? In teacher-jargon, this is called 'the hidden curriculum'.

In this article I intend to do no more than present a range of opinions and ideas for prospective programmers or reviewers to think about. I hope you find them useful. If you disagree or can add something - you know Keith's address.

The following questions can be asked of any educational program, or any idea which is being considered as the focus for such a program:

1. What does the program do (or make possible) that everyday teaching methods can't ? How relevant, effective and useful is it to 'computerise' this particular idea ?

2. Which teaching method has generally proved to be most effective ? Is the program at least as effective - and how can this be assessed (can such assessment of pupil's responses and performance become an intrinsic part of the program's structure) ?

3. What features of the computer's potential can be exploited such that
it becomes a truly potent (and unique ?) means whereby to induce the required levels of learning ?

4. How completely does the program cater for the types of mistake or problems that children are likely to encounter whilst 'being taught' ?

5. How complete is the documentation - both on the package and on the screen. Imagine that you have only the cassette. If you RUN the program, is it self-explanatory ? Does it tell you what to do when you go wrong ? Is it always clear what you have to do - and how ? Is the language applicable to the age range at which the program is targetted ? Can you as programmer (or reviewer) analyse and state the educational objective both precisely and clearly ? (Or is it all a bit vague and generalised ?)

6. Is the scoring and reward system appropriate ? Is the child given sufficient time in which to read and absorb comments and assessments ? (I'll have more to say about rewards in a moment).

7. If sound is used, can it be turned off ? (not really a problem for the MTX or RS). Just try using a noisy program in a classroom where the majority of children are **not** using the computer! Is the use of sound meant as a stimulus, a comment or is it an implicit part of the activity ? Does it enhance, or is it superfluous ?

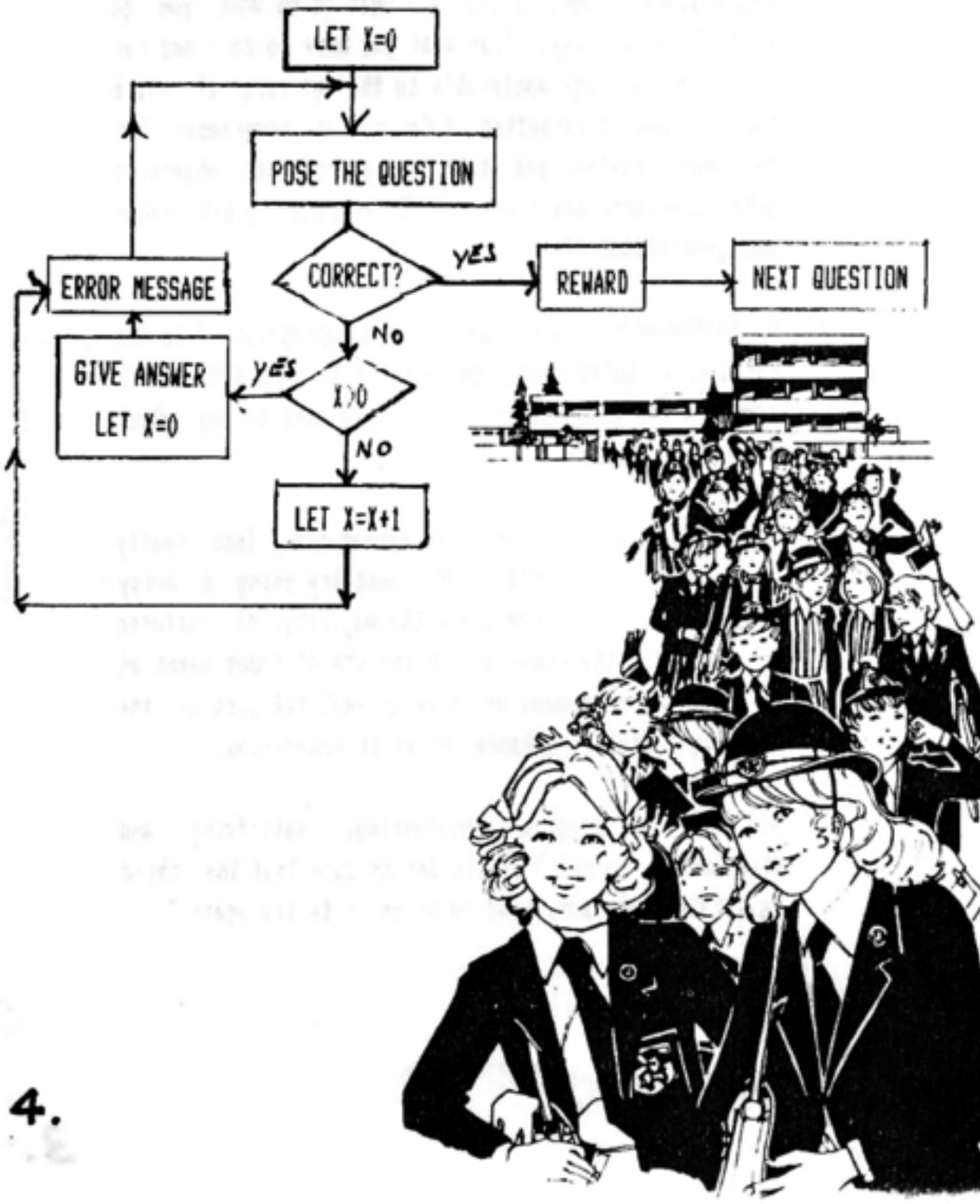
8. Is the program interesting, satisfying and stimulating to use ? Is its design such that the child is likely to be motivated to go on or to try again ?

9. Does the program stimulate **active** participation, or is the user essentially a passive observer for much of the time ? Does the term 'interactive' actually mean anything ?

10. Is it likely that the program will lead to furtherance of **understanding** ? Is its style of presentation and content such that it effectively illustrates or characterises the objective-in-mind ? If some new approach has been used, or if the idea has been transformed into a game, are such changes and additions based upon educational grounds, or are they gimmicks ? Do they add to or detract from the meaningfulness of the task ?

There are two further considerations which are worthy of whole articles in themselves:

11. **FEEDBACK** - Many software packages overlook the fact that children - especially young children - need to be kept up-to-date with their progress. They need constant help, clues, or to be told to ask for help. They easily lose confidence or concentration. If they make a mistake they need immediate and clear advice. When treating errors, far too many programs rely upon the closed loop. In very crude terms such loops follow this line of reasoning:



There is little to be said for such an approach. There are at least two good alternatives:

(a) After the initial loop, error message and provision of the correct answer, respond to further failures by giving a **similar** problem - not the same one again and again and again.

(b) After 2 or 3 such varied failures, halt the program and advise the child to ask for help - or return to a page which gives clear instructions and an explanation of the principle which underpins the questions.

Central to any teaching activity is the process of making assessments and diagnoses. Of inestimable value is a facility which allows the teacher to access an analysis of a child's performance. Where programs begin by asking for a name, the possibilities of personalised record-keeping become self-evident. Where a school has a printer, the possibilities for permanent record-keeping are greatly enhanced.

Remember : educational software should not only be an aid to learning, it should also be an aid to teaching. It needs to help and support both pupil and teacher.

12. **REWARDS** : Commonly, a child is rewarded with a sound effect or a comment or updated graphics. There are many other possibilties. eg.

(a) A game based upon the concepts or skills being taught in the program.

(b) The chance to take a simple test and thereby move to a higher level of work (the program would need to make occasional returns to previous levels of activity to ensure that a child does not race too far ahead too soon).

(c) The provision of clues to help solve some form of related puzzle or problem (a similar idea to the one used in TV's 'A Question Of Sport' wherein elements of a large photograph are revealed bits at a time).

(d) Displaying a bar chart which plots correct and incorrect responses. This could be made into a competitive game where two or more children use a program simultaneously or one after the other - but this may not always be a desirable practice.

(e) A crossword (or something similar using numbers) which can be attempted for a few minutes after the child has got a high enough score. The rules for these games would be tied to the principles which underlie the main program.

(f) A simple 'animated sequence'. (eg. begin by showing a seed. The next time you see rootlets, then the the main root, then a first shoot...etc).

Lastly, a few more generalised comments. It would be a mistake to suggest or believe that educational software has to be complete in itself. For one thing it is almost certainly going to be used only as an aid in the development of some larger investigation. It would, for example, be quite possible to write a program which explains the Fibonacci sequence. But this could also be achieved in the traditional way. However, once an appreciation of the sequence has been acquired, and after children have had some experience of creating and exploring the properties of such sequences, a software package could be used to provide raw data for further investigations. In essence, such a program would simply seek to remove the drudgery of repeated calculations. It could also allow the child to ask 'What if...?' questions and to test the outcome almost immediately.

And must the child always work **at** the computer? If games can be written so that they can be SAVEd in the middle of a run - thus preserving current data, scores etc. - then why not the same idea for educational packages ? The child could be introduced to a task (say measuring, discovering dates or data, interpreting information, etc.). These tasks could well take several 'sessions' to complete and could be SAVEd after each return to the computer (which could then set a new task based upon the child's input). The beauty of this idea is that the child could gradually build up a complete record of his/her own activities and researches. Once again, the value of a printer becomes self-evident.

Neither must teachers or programmers overlook the fact that there is much to be gained in school by using word processors or adventure games. The former can be used or adapted to help with writing stories, reports, letters, poetry..... The latter can be used as an aid (and stimulus) to teaching reading, logical thinking, creativity, transforming ideas into diagrams.....

The possibilities may not literally be endless, but they're not far short. What **is** required, however, is a commitment from teachers and writers to become more aware of the potential, and critical of the limitations of computers in the classroom. Many teachers are still unsure or suspicious. They have not as yet been convinced of the worth of this new activity. How far, I wonder, is this because the majority of currently available educational software is of little or limited use ? ●

SOFTWARE REVIEW

AGROVATOR (another view) by B. MOSS

This super game must rate along side **Pothole Pete** as one of the best games available for the **Memotech. Agrovator** is a game with lots of very clever mazes and 20 different objects to aggregate, some quite canny.

There are a couple of surprises....if you read last month's you will know all about these.

Agrovator makes a sound not unlike a girder being hit by a hammer and is different unlike the dull gulp, gulp, gulp sounds some make.

The graphics are splendid and as a matter of fact I find this game hard to criticise in any way. If this game is one of the first from **SYNTAX** then I can't wait for more. ●

Competition

One-Liners

HERE IS THE FIRST ENTRY TO BE PUBLISHED OUT OF THE HUNDREDS SUBMITTED. THIS ONE IS BY NEIL EMPSEN OF WILTSHIRE.
Line 8 is the competition line.Also NOTE that lines must be entered in the abbreviated form.

```
2 GOTO 100
4 VS 4: COLOUR 2,1
6 LET X=128: LET Y=95
8 CLS : CIRCLE X,Y,70: FOR N=1 TO 200:
COLOUR 3,INT(RND*13)+2: PLOT X,Y: ANGLE
 INT(RND*N/2): DRAW INT(RND*65)+5: COLO
UR 4,INT(RND*13): NEXT : FOR M=72 TO 90
: CIRCLE X,Y,M: COLOUR 3,INT(RND*15): N
EXT : PAUSE M^2: GOTO 8
100 LET K=2
101 FOR J=1 TO 3
102 READ A$
104 CSR 20-LEN (A$)/2,K: PRINT A$
106 LET K=K+2
108 NEXT
109 LET K=K+3
110 FOR J=1 TO 6
112 READ A$
114 CSR 20-LEN (A$)/2,K; PRINT A$
116 LET K=K+2
118 NEXT
190 PAUSE 10000
199 GOTO 4
200 DATA DEMONSTRATION PROGRAM,COMPUTER
 : MEMOTECH MTX512,OBJECT : PACK A SING
LELINE
210 DATA WRITTEN BY,NEIL EMSEN,7 ARGOSY
 ROAD,LYNEHAM,CHIPPENHAM,WILTSHIRE
```

●

UTILITY

DOUBLE HEIGHT by E. ROY

This program shows how text can be drawn in double height format , with or without multi-coloured lettering. The lettering used is entered into the data statements so those examples shown in the pogram could be changed and others used.

```
100 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
110 REM ‡‡‡ DOUBLE HEIGHT HEADINGS ‡‡‡
120 REM ‡‡‡  ON GRAPHICS SCREENS.  ‡‡‡
130 REM ‡‡‡        by E. Roy       ‡‡‡
140 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
150 REM
160 LET TRUE=0: LET FALSE=-1
170 VS 4: COLOUR 0,1: COLOUR 2,1
180 COLOUR 3,15: COLOUR 4,1: CLS
190 LET MULTICOLOUR=FALSE
200 READ NUMBER
210 FOR DOUBLE=1 TO NUMBER
220 IF DOUBLE>3 THEN  LET MULTICOLOUR=TRUE
230 READ DHI,DHY,DH$
240 GOSUB 300
250 NEXT
260 GOTO 260
270 REM-----------------------------
280 REM    PRINT  DH$ DOUBLE HEIGHT
290 REM-----------------------------
300 CSR DHX,DHY+1
310 COLOUR 1,0
320 PRINT DH$
330 LET PLX=DHX‡8-1: LET PLY=191-(DHY‡8)
340 LET LY=PLY
350 FOR SY=PLY-8 TO PLY-15 STEP -1
360 IF MULTICOLOUR=TRUE THEN  COLOUR 3,RND‡13+2
370 FOR SX=PLX TO LEN (DH$)‡8+PLX
380 LET POINT=ASC(GR$(SX,SY,1))
390 IF POINT=1 THEN  ATTR 2,1: PLOT SX,SY:
    ATTR 2,0: PLOT SX,LY: PLOT SX,LY-1
400 NEXT SX: LET LY=LY-2: NEXT SY
410 RETURN
420 REM-----------------------------
430 REM Data for Double Height Strings
440 REM-----------------------------
450 DATA 5
460 DATA 3,0,--- DOUBLE HEIGHT DEMO ---
470 DATA 3,4,Could be speeded up using
480 DATA 9,6,assembly code.
490 DATA 3,10,--- MULTICOLOURED DEMO  ---
500 DATA 9,14,Give it a try.
```

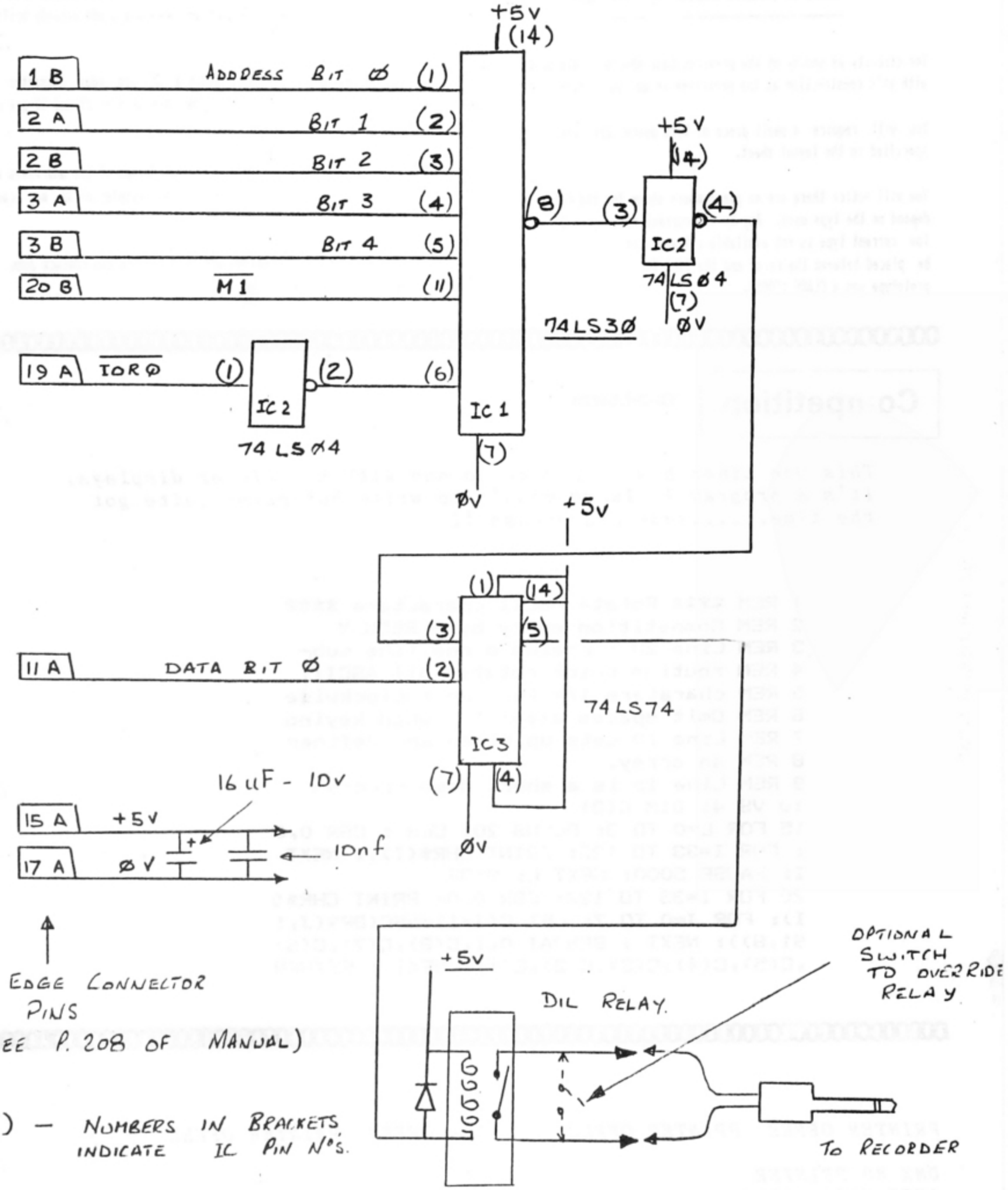
THE MISSING WILLIE SHOWS HIS FACE ...

HERE'S THE MISSING NODDY PAGES FROM LAST MONTH'S PROGRAM " WILLIE WORM"

```
INSTRUCTIONS
‡‡‡‡‡‡‡‡‡‡‡‡
''''''''''''

‡D INSTRUCTIONS. ‡E ‡5 IN2. ‡E ‡R

AVOID HITTING THE WALL OR YOUR TAIL.

WILLIE WILL GET LONGER THE MORE HE

EATS BUT DO NOT EAT THE GREEN BUGS

IF YOU HAVE NO STRENGH POINTS SHOWN

IN THE BOTTOM LEFT HAND CORNER.

STRENGH POINTS ARE GAINED BY EATING

BLUE FLOWERS.

      PRESS RETURN

IN2
MOVE LEFT AND THE 'X' KEY WILL MAKE

HIM MOVE RIGHT.KEEPING YOUR FINGER ON

EITHER KEY WILL MAKE HIM TURN IN

CIRCLES.REMEMBER THE MORE HE EATS THE

LONGER HE WILL GROW.

    PRESS RETURN TO START GAME.
```

HARDWARE

JOHN HUDSON SHOWS YOU HOW TO AUTOMATE YOUR CASSETTE



AUTOMATIC CASSETTE SYSTEM

JOHN HUDSON

AUTOMATED CASSETTE HANDLER By John Hudson

The circuit, as shown, on the previous page should create no problems with it's construction as the prototype is up and running.

You will require a small piece of Vero board and the components specified in the layout sheet.

You will notice there are no pin numbers shown for the **relay** as these depend on the type used. Any 5v low current relay will do but if the low current type is not available a transistor drive circuit should be placed between the relay and the 74LS 74. The relay used on the prototype was a CLARE 15005B.

The diode shown in the circuit is a IN4148 but in some cases this may not be required as numerous DIL relays have the diode already built in the component.

If it proves difficult to obtain a 30 pin edge connector an alternative 28 pin Spectrum tpe may be used as pins 29 and 30 are not required.

NOTE: The circuit only performs a partial decode of the addresses and any peripheral device with an address in multiples of **31** may cause false operation of the circuit.

Members who experience problems with the circuit can contact John by telephone on 09367-3058.

Competition

One-Liners

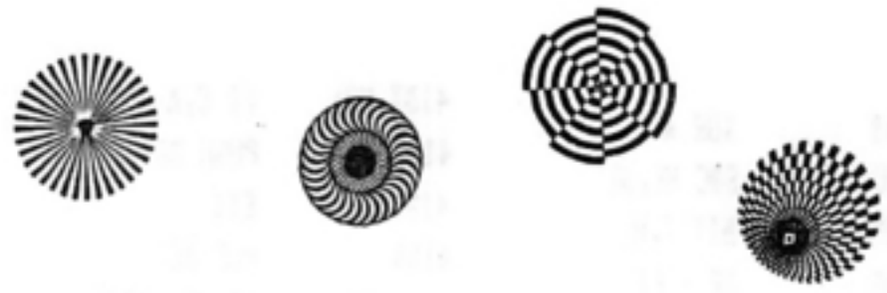
This One Liner has helped me no end with my printer displays. It's a program I always mean't to write but never quite got the time.....same old excuses!!

```
1 REM **** Rotate ASCII characters ****
2 REM Competition entry by H REDWAY
3 REM Line 20 contains a one line sub-
4 REM routine which rotates all ASCII
5 REM charaters 1/4 turn anti-clockwise
6 REM Omit spaces after ":" when keying
7 REM Line 10 sets up sceen and defines
8 REM an array.
9 REM Line 15 is a short demo program.
10 VS 4: DIM C(8)
15 FOR L=0 TO 3: GOSUB 20: CLS : CSR 0,0
: FOR I=33 TO 122: PRINT CHR$(I),: NEXT
I: PAUSE 5000: NEXT L: STOP
20 FOR I=33 TO 122: CSR 0,0: PRINT CHR$(
I): FOR J=0 TO 7: LET C(J+1)=ASC(GR$(J,1
91,8)): NEXT : GENPAT 0,I,C(8),C(7),C(6)
,C(5),C(4),C(3),C(2),C(1): NEXT : RETURN
```

PROGRAM LISTING

3D ROTATOR

BY MEL GROVER BEDFORD.



```
10 GOTO 40
20 CODE

400F          JP START
4012          RET
4013 PERSP:   LD A,#80
4015          LD (FLAG),A
4018          LD DE,#0100
401B          LD HL,(Z)
401E          CALL CALC
4021          LD (Z),HL
4024          LD (VARST),HL
4027          XOR A
4028          LD (FLAG),A
402B          LD DE,#0000
402E          LD HL,(X)
4031          EX DE,HL
4032          CALL CALC
4035          LD DE,#0080
4038          CALL FLADD
403B          LD (X),HL
403E          LD DE,96
4041          LD HL,(Y)
4044          PUSH DE
4045          CALL CALC
4048          POP DE
4049          CALL FLADD
404C          LD (Y),HL
404F          RET
4050 CALC:    XOR A
4051          EX DE,HL
4052          SBC HL,DE
4054 CHECK:   BIT 7,H
4056          CALL NZ,FLSUB
4059          LD B,H
405A          LD C,L
405B          LD A,(FLAG)
405E          ADC A,A
405F          JR C,A1
4061          LD BC,(VARST)
4065 A1:      PUSH HL
4066          PUSH BC
4067          EX DE,HL
4068          LD HL,#0000
406B          CALL CHE
406E          JR NC,A2
4070          XOR A
4071          POP HL
4072          POP DE
4073          CP H
4074          JR NZ,A3
4076          EX DE,HL
4077 A3:      CALL DIV16
407A          LD B,H
407B          LD C,L
407C          LD HL,#0000
407F          CALL CHE
4082          CALL DIV16
4085          RET
4086 A2:      LD B,#08
4088          CALL DIVB
408B          POP DE
408C          POP DE
408D          RET
408E DIV16:   LD B,#04
4090 DIVB:    SRL H
4092          RR L
4094          DJNZ DIVB
4096          RET
4097 CHE:     SRL B
4099          RR C
409B          JR NC,B1
409D          ADD HL,DE
409E          RET C
409F B1:      JR Z,B2
40A1 B3:      EX DE,HL
40A2          ADD HL,HL
40A3          RET C
40A4          EX DE,HL
40A5          JR CHE
40A7 B2:      LD A,B
40A8          CP #00
40AA          RET Z
40AB          JR B3
40AD FLADD:   EX DE,HL
40AE          LD A,(FLAG)
40B1          SRL A
40B3          JR NC,B4
40B5          ADD HL,DE
40B6          LD (FLAG),A
40B9          RET
40BA B4:      SBC HL,DE
40BC          RET
40BD FLSUB:   PUSH DE
40BE          EX DE,HL
40BF          XOR A
40C0          LD H,A
40C1          LD L,A
40C2          SBC HL,DE
40C4          LD A,(FLAG)
40C7          INC A
40C8          LD (FLAG),A
40CB          POP DE
40CC          RET
40CD SINCOS:  RRA
40CE          JR NC,C1
40D0          LD BC,#005A
40D3          ADD HL,BC
40D4          PUSH HL
40D5          LD BC,#0168
40D8          SBC HL,BC
40DA          JR C,C2
40DC          POP BC
40DD          PUSH HL
40DE C2:      POP HL
40DF C1:      PUSH HL
40E0          XOR A
40E1          LD BC,#00B4
40E4          SBC HL,BC
40E6          JR C,D1
40E8          LD A,#01
40EA          POP BC
40EB          PUSH HL
40EC D1:      POP HL
40ED          LD (FLAG),A
40F0          LD BC,SINTAB
40F3          ADD HL,BC
40F4          LD C,(HL)
40F5          LD B,#00
40F7          LD (VARST),BC
40FB          EX DE,HL
40FC          CALL CHECK
40FF          RET
4100 MULT:    PUSH BC
4101          PUSH HL
4102          PUSH AF
4103          CALL SINCOS
4106          LD A,(FLAG)
4109          RRA
410A          CALL C,FLSUB
410D          POP AF
410E          INC A
410F          POP DE
4110          POP BC
4111          PUSH AF
4112          PUSH HL
4113          EX DE,HL
4114          LD D,B
4115          LD E,C
4116          CALL SINCOS
4119          LD A,(FLAG)
411C          RRA
411D          CALL C,FLSUB
4120          POP DE
4121          POP AF
4122          RRA
4123          JR NC,E1
4125          ADD HL,DE
4126          RET
4127 E1:      EX DE,HL
4128          SBC HL,DE
412A          RET
412B ROTATE:  LD HL,(THETA)
412E          LD DE,(X)
4132          LD BC,(Z)
4136          PUSH HL
4137          PUSH DE
4138          PUSH BC
4139          LD A,#01
413B          CALL MULT
413E          LD (X),HL
4141          POP BC
4142          POP DE
4143          POP HL
4144          XOR A
4145          CALL MULT
4148          PUSH HL
4149          POP BC
414A          LD HL,(PHI)
414D          LD DE,(Y)
4151          PUSH HL
4152          PUSH DE
4153          PUSH BC
4154          XOR A
4155          CALL MULT
4158          LD (Z),HL
415B          POP BC
415C          POP DE
415D          POP HL
415E          LD A,#01
4160          CALL MULT
4163          PUSH HL
4164          POP BC
4165          LD HL,(PSI)
4168          LD DE,(X)
416C          PUSH HL
416D          PUSH DE
416E          PUSH BC
416F          LD A,#01
4171          CALL MULT
4174          LD (X),HL
4177          POP BC
4178          POP DE
4179          POP HL
417A          XOR A
417B          CALL MULT
417E          LD (Y),HL
4181          RET
4182 LIPLT:   LD HL,(X)
4185          LD DE,(PCX)
4189          LD C,#01
```

```
418B        XOR A
418C        SBC HL,DE
418E        BIT 7,H
4190        JR Z,F1
4192        LD H,#00
4194        SUB L
4195        LD L,A
4196        DEC C
4197        DEC C
4198 F1:    LD B,L
4199        PUSH BC
419A        LD HL,(Y)
419D        LD DE,(PCY)
41A1        LD B,#01
41A3        XOR A
41A4        SBC HL,DE
41A6        BIT 7,H
41A8        JR Z,F2
41AA        SUB L
41AB        LD L,A
41AC        DEC B
41AD        DEC B
41AE F2:    POP DE
41AF        LD C,D
41B0        LD D,B
41B1        LD B,L
41B2        EXX
41B3        PUSH HL
41B4        EXX
41B5        LD A,C
41B6        CP B
41B7        JR NC,G1
41B9        LD L,C
41BA        PUSH DE
41BB        XOR A
41BC        LD E,A
41BD        JR G2
41BF G1:    OR C
41C0        JR Z,G3
41C2        LD L,B
41C3        LD B,C
41C4        PUSH DE
41C5        LD D,#00
41C7 G2:    LD H,B
41C8        LD A,B
41C9        RRA
41CA J2:    ADD A,L
41CB        JR C,H1
41CD        CP H
41CE        JR C,H2
41D0 H1:    SUB H
41D1        LD C,A
41D2        EXX
41D3        POP BC
41D4        PUSH BC
41D5        JR H3
41D7 H2:    LD C,A
41D8        PUSH DE
41D9        EXX
41DA        POP BC
41DB H3:    LD HL,(PCY)
41DE        LD A,C
41DF        LD C,B
41E0        DEC B
41E1        JR Z,H4
41E3        INC B
41E4 H4:    ADD HL,BC
41E5        LD (PCY),HL
41E8        LD HL,(PCX)
41EB        LD C,A
41EC        LD B,A
41ED        DEC B
41EE        JR Z,J1
41F0        INC B
41F1 J1:    ADD HL,BC
41F2        LD (PCX),HL
41F5        CALL PTPLT
41F8        EXX
41F9        LD A,C
41FA        DJNZ J2
41FC        POP DE
41FD G3:    EXX
41FE        POP HL
41FF        EXX
4200        RET
4201 PTPLT: XOR A
4202        LD HL,(PCX)
4205        CP H
4206        RET NZ
4207        LD C,L
4208        LD HL,(PCY)
420B        CP H
420C        RET NZ
420D        LD A,191
420F        SUB L
4210        RET C
4211        LD B,A
4212        CALL LOCRAM
4215        LD B,A
4216        INC B
4217        LD A,#01
4219 J3:    RRCA
421A        DJNZ J3
421C        LD B,(HL)
421D        OR B
421E        LD (HL),A
421F        RET
4220 START: CALL SCSET
4223        LD A,(MODE)
4226        RRA
4227        JR NC,INIT ;DRAW ON TOP
4229        RRA
422A        LD BC,#1800
422D        LD DE,SCREE
4230        JR NC,CLS  ;CLEAR MEMORY
4232        LD HL,(SPSCR)
4235 LOADS: LDIR
4237        JR INIT
4239 CLS:   LD H,D
423A        LD L,E
423B        DEC HL
423C        LD (HL),0
423E        JR LOADS
4240 INIT:  LD BC,(NUMB)     ;NO OF COORDS
4244        LD HL,(ADDR)     ;DATA HERE
4247 NXTPT: PUSH BC
4248        XOR A
4249        LD D,A
424A        LD E,(HL)
424B        BIT 7,E
424D        JR Z,PLUSA
424F        DEC D
4250 PLUSA: LD (X),DE
4254        LD D,A
4255        INC HL
4256        LD E,(HL)
4257        BIT 7,E
4259        JR Z,PLUSB
425B        DEC D
425C PLUSB: LD (Y),DE
4260        LD D,A
4261        INC HL
4262        LD E,(HL)
4263        BIT 7,E
4265        JR Z,PLUSC
4267        DEC D
4268 PLUSC: LD (Z),DE
426C        INC HL
426D        PUSH HL
426E        CALL ROTATE
4271        LD DE,(PX) ;TRANSLATION & SCALING
4275        LD HL,(X)
4278        ADD HL,HL
4279        ADD HL,DE
427A        LD (X),HL
427D        LD DE,(PY)
4281        LD HL,(Y)
4284        ADD HL,HL
4285        ADD HL,DE
4286        LD (Y),HL
4289        LD DE,(PZ)
428D        LD HL,(Z)
4290        ADD HL,DE
4291        LD (Z),HL
4294        CALL PERSP
4297        POP HL
4298        LD A,(HL)
4299        INC HL
429A        PUSH HL
429B        RRA
429C        JR C,JPA
429E        LD HL,(X)
```

```
42A1        LD (PCX),HL
42A4        LD HL,(Y)
42A7        LD (PCY),HL
42AA        CALL PTPLT
42AD        JR M1
42AF JPA:   CALL LIPLT
42B2 M1:    POP HL
42B3        POP BC
42B4        DJNZ NXTPT
42B6        JP VRAMBK
42B9        RET
42BA NUMB:  DS 2
42BC X:     DS 2
42BE Y:     DS 2
42C0 Z:     DS 2
42C2 PX:    DS 2
42C4 PY:    DS 2
42C6 PZ:    DS 2
42C8 PHI:   DS 2
42CA THETA: DS 2
42CC PSI:   DS 2
42CE VARST: DS 2
42D0 FLAG:  DS 1
42D1 MODE:  DS 1
42D2 SPSCR: DS 2
42D4 ADDR:  DS 2
42D6 PCX:   DS 2
42D8 PCY:   DS 2
42DA SINTAB: DS 182
4390        DS 200      ;MAX NO OF POINTS
4458        DS 200      ;IS 100 POINTS
4520        RET
4521 LOCRAM: LD A,191
4523        SUB B
4524        SRL A
4526        SRL A
4528        SRL A
452A        LD H,A
452B        LD A,C
452C        AND #F8
452E        LD L,A
452F        LD A,191
4531        SUB B
4532        AND #07
4534        ADD A,L
4535        LD L,A
4536        LD A,C
4537        AND #07
4539        LD DE,SCREE
453C        ADD HL,DE
453D        RET
453E SCSET: LD DE,15
4541        LD HL,#FF5D
4544        LD A,(NUMBS)
4547        CP 0
4549        JP Z,SCRENO
454C        LD B,A
454D LOOPA: ADD HL,DE
454E        DJNZ LOOPA
4550 SCRENO: LD DE,3
4553        ADD HL,DE
4554        LD A,(HL)
4555        LD (XST),A
4558        INC HL
4559        LD A,(HL)
455A        LD (YST),A
455D        INC HL
455E        LD A,(HL)
455F        LD (XCT),A
4562        INC HL
4563        LD A,(HL)
4564        LD (YCT),A
4567        RET
4568 NUMBS: DS 1
4569 XST:   DS 1
456A YST:   DS 1
456B XCT:   DS 1
456C YCT:   DS 1
456D PVBK:  DS 2
456F STY:   DS 1
4570 CTY:   DS 1
4571 BUFF:  DS 1
4572        RET
4573 VRAMBK: DI
4574        LD A,(YCT)
4577        LD (CTY),A
457A        LD DE,(PVBK)
457E LOOPE: CALL VRAM
4581        PUSH DE
4582        POP HL
4583        LD BC,SCREE
4586        ADD HL,BC
4587        LD A,(BUFF)
458A        LD B,A
458B LOOPD: LD C,(HL)
458C        INC HL
458D        CALL DATA
4590        DJNZ LOOPD
4592        INC D
4593        LD A,(CTY)
4596        DEC A
4597        LD (CTY),A
459A        JP NZ,LOOPE
459D        EI
459E        RET
459F VRAM:  PUSH AF
45A0        PUSH BC
45A1        LD A,E
45A2        OUT (2),A
45A4        LD A,D
45A5        ADD A,#40
45A7        OUT (2),A
45A9        POP BC
45AA        POP AF
45AB        RET
45AC        RET
45AD DATA:  PUSH AF
45AE        LD A,C
45AF        OUT (1),A
45B1        POP AF
45B2        RET
45B3        DS 1
4584 SCREE: DS 192
4674        DS 192
4734        DS 192
47F4        DS 192
48B4        DS 192
4974        DS 192
4A34        DS 192
4AF4        DS 192
4BB4        DS 192
4C74        DS 192
4D34        DS 192
4DF4        DS 192
4EB4        DS 192
4F74        DS 192
5034        DS 192
50F4        DS 192
51B4        DS 192
5274        DS 192
5334        DS 192
53F4        DS 192
54B4        DS 192
5574        DS 192
5634        DS 192
56F4        DS 192
57B4        DS 192
5874        DS 192
5934        DS 192
59F4        DS 192
5AB4        DS 192
5B74        DS 192
5C34        DS 192
5CF4        DS 193
5DB5        RET
```

```
30 RETURN
40 LET POK=592+PEEK(64170)+256*PEEK(64171)
50 LET FLAG=0: LET B$=""
60 LET PX=10: LET PY=40: LET PZ=50
70 LET PHY=0: LET PSI=0: LET THETA=0
80 VS 4: CLS
90 GOSUB 270
100 GOSUB 530
110 GOTO 300
120 DATA 20,-20,20,0,20,-20,-20,1,-20,-20,-20,1,-20,-20,20,1,20,-20,20,1,20,20,20,1,20,20,-20,1,-20,20,-20,1
130 DATA -20,20,20,1,20,20,20,1,20,-20,-20,0,20,20,-20,1,-20,-20,-20,0,-20,20,-20,1,-20,-20,20,0,-20,20,20,1
140 IF V<0 THEN  LET V=V+65536
150 LET A=INT(V/256): LET B=MOD(V,256)
160 POKE D,B: POKE D+1,A
170 LET D=D+2
180 RETURN
190 DATA -7,5,-26,0,-7,0,-18,1,-7,0,9,1,-7,4,12,1,-7,8,12,1,-7,8,-10,1,-7,8,-13,1,-7,5,-26,1
200 DATA 7,5,-26,1,7,0,-18,1,7,0,9,1,7,4,12,1,7,8,12,1,7,8,-10,1,7,8,-13,1,7,5,-26,1
210 DATA -7,0,-18,0,7,0,-18,1,-7,0,9,0,7,0,9,1,-7,8,-13,0,7,8,-13,1,-7,8,-10,0,7,8,-10,1,5,13,-2,1,-5,13,-2,1,-7,8,-10,1
220 DATA -7,8,12,0,7,8,12,1,5,13,4,1,-5,13,4,1,-7,8,12,1,-5,13,-2,0,-5,13,4,1,5,13,-2,0,5,13,4,1
230 DATA 2,12,-25,0,-2,12,-25,1,-2,10,-25,1,2,10,-25,1,2,12,-25,1,2,12,-4,1,-2,12,-4,1,-2,10,-7,1,2,10,-7,1,2,12,-4,1,-2,12,-25,0,-2,
240 DATA -2,10,-7,1,2,10,-25,0,2,10,-7,1                                                                    12,-4,1,-2,10,-25,0
250 RESTORE 190: LET V=52
260 GOTO 280
270 RESTORE 120: LET V=16
280 GOSUB 610
290 RETURN
300 IF FLAG=1 THEN  GOTO 340 ELSE  IF INKEY$="" THEN  GOTO 300
310 IF INKEY$="t" THEN  GOSUB 250
320 LET A$=INKEY$
330 IF A$="r" AND B$<>"s" THEN  LET FLAG=1: LET A$=B$
340 LET PX=PX-10*(A$=CHR$(8))+10*(A$=CHR$(25))
350 IF INKEY$="s" THEN  LET FLAG=0
360 LET B$=A$
370 LET PY=PY+10*(A$="u")-10*(A$="d")
380 LET PZ=PZ-10*(A$=CHR$(10))+10*(A$=CHR$(11))
390 LET THETA=THETA-10*(A$="y")+10*(A$="Y")
400 LET PSI=PSI-10*(A$="z")+10*(A$="Z")
410 LET PHY=PHY-10*(A$="x")+10*(A$="X")
420 IF THETA>360 THEN  LET THETA=THETA-360
430 IF PSI>360 THEN  LET PSI=PSI-360
440 IF PHY>360 THEN  LET PHY=PHY-360
450 IF THETA<0 THEN  LET THETA=THETA+360
460 IF PSI<0 THEN  LET PSI=PSI+360
470 IF PHY<0 THEN  LET PHY=PHY+360
480 IF A$="1" OR A$="2" THEN  POKE POK+129,VAL(A$)
490 IF PZ>256 THEN  LET PZ=256
500 GOSUB 530
510 GOTO 300
520 STOP
530 LET V=PX: LET D=POK+114: GOSUB 140
540 LET V=PY: GOSUB 140
550 LET V=PZ: GOSUB 140
560 LET V=PHY: LET D=POK+120: GOSUB 140
570 LET V=THETA: GOSUB 140
580 LET V=PSI: GOSUB 140
590 GOSUB 20
600 RETURN
610 CSR 5,5: PRINT "Please wait a moment.": FOR F=1 TO V: READ X,Y,Z,P
620 IF X<0 THEN  LET X=X+255
630 IF Y<0 THEN  LET Y=Y+255
640 IF Z<0 THEN  LET Z=Z+255
650 POKE POK+316+F*4,X: POKE POK+317+F*4,Y: POKE POK+318+F*4,Z:
    POKE POK+319+F*4,P: NEXT F
660 POKE POK+107,V
670 CLS
680 RETURN
690 REM INITIALISE LOOKUP SINE TABLE
700 LET POK=592+PEEK(64170)+256*PEEK(64171)
710 FOR F=0 TO 180
720 LET C=INT(256*SIN(PI*F/180))
730 IF C=256 THEN  LET C=255
740 POKE F+POK+138,C
750 NEXT F
```

3D ROTATOR.

Each figure can be rotated in all axes and moved away from or toward you with full perspective. The figure may also be moved UP,DOWN,LEFT,and RIGHT.

Each location is specified by four items of data: X,Y,Z,A. If A = 0 then the point is plotted. If A = 1 then a line is plotted from the last point to the present plot. If you examine the data for the cube, the method should become obvious.

Alterations to the figure can be made by the following keys:
d .. Move Down u .. Move Up y .. rotate positive axis
Y .. rotate negative axis x .. rotate positive x axis
X .. rotate negative x axis z .. rotate positive z axis
Z .. rotate negative z axis UP arrow .. movetoward Down arrow .. move away . Left arrow .. move left Right arrow .. move right.
r repeats last instruction until s is pressed.
1 CLS
2 Overplots figures.
t changes cube to tank.

ENTERING THE PROGRAM

Type in the code and Basic lines. Note that the address mode should contain a 1 to start with. Initialise sin table by Goto 690 then make sure the Alpha Lock is off. Type "RUN" (RET) DON'T ADD ANY CODE BEFORE THE CODE .....IT WILL UPSET THE POKES.

This is an excellent program that will allow you to rotate an object about the X or Y axis and moved towards you or away from you. This is a fine example of what can be achieved on the MTX when a little thought is given to the subject.....What with the SCROLL ROUTINE and now this... it looks like all the brains are down in Bedford ..??

RAFFLE WINNER

Member Mr. G. W. Walton, Southampton B·001167
Mr. Watson receives RS232 interface, Pascal Rom & 3 pieces of software.
We shall no longer run the raffle.... the results are too disappointing. I thought this would be an excellent way to get members the add-on they require, but the first time it was run the club ended up donating to the prize money, and this time the results are even more disappointing

---
PLEASE READ CAREFULLY ....

One of the biggest moans we hear from members is the lack of publications revolving around the MTX computer. The Club has the opportunity of aquiring the publishing rights of an excellent book called "ADVANCED PROGRAMMING WITH THE MTX". No, it's not the orginal book that was advertised, this is a completely new publication. However, before the Club can commit itself, we need to know how many people will be interested in buying the book. The published price will be £6.95 including postage and packing.
SEND NO MONEY NOW BUT FILL IN THE FORM IF YOU WISH TO PURCHASE THE PUBLICATION. Distribution will be on a first come first served basis. Expected delivery within 6 weeks.

Please put my name on the list to receive "THE ADVANCED PROGRAMMING WITH THE MTX"BOOK.

NAME _____

ADDRESS _____

_____

POSTCODE MEMBERSHIP No A/B/C --------

ASSEMBLY LANGUAGE

.....continued from last month.

If HL contained $3C00_H$ after **PUSH HL** the stack would hold the value $3C00_H$ and the same value would **still** be contained in the HL pair. Since the stack operates on a last in - first out basis, to restore registers to the orginal values, after carrying out other operations ,they must be pooped of the stack in the **reverse** order.

```
LD HL,#3C00
LD BC,#1234
LD DE,#00AE
PUSH BC
PUSH DE
PUSH HL .....at this point all registers still contain their original
LD HL,#2000  ....values
ADD HL,DE
ADC HL,BC
LD (STORE),HL
POP HL ..... The value in HL will now be restored to #3C00
POP DE
POP BC
```

This ideal method of saving data allows you to use the registers for other calculations, and then restore their original values for further instructions.

Other instructions concerned with the stack includes **EX (SP),HL** which loads the HL registers with the value at the top of the stack and places the data in the HL pair onto the top of the stack.

```
LD BC,#1234
PUSH BC
LD HL,#FA7A
EX (SP),HL
RET
```

After the PUSH BC the top of the stack would contain #1234, and after EX (SP),HL the top of the stack would contain #FA7A, and HL would contain #1234. The program would **return to location #FA7A.**

A **CALL** instruction can be directly compared to Basic's **GOSUB** statement, and **RET** equivalent to Basic's **RETURN** command. In machine code you can also use the instructions within conditional statements:

```
10 IF A > 7  THEN GOSUB 100

        LD A,C              ;Get value in C into A register
        CP 8                ;CP 8 ..testing to see if > 7
        CALL NC, L100       ;If value = or > 8 then Gosub L100
```

All of the conditional statements can be used: NC,C,P,PO,PE. Assembly language subroutines are far more flexible than Basic's GOSUB. However

the same rule applies: **for every CALL there must be an equivalent RET statement,** and you must be sure that the return address has not been corrupted by misuse of the stack.

LD is one mnemonic concerned with transferring data to and from memory, or between registers. The first register in the instruction is **always** the **receiving** register. **LD A,C** means load the A register with the value contained in the C register.

The synatx rules for using the LD instruction are easy to memorise.

LD A,(BUFFER) .....LD A FROM MEMORY LOCATION LABLED BUFFER.
LD A,(HL) .....LD ,FROM MEMORY LOCATION POINTED TO BY HL
e.g .....if HL = #FFD3 and memory location #FFD3 contained #0D then after the above instruction A would contain #0D.

```
20 LET HL = 15360
30 LET A = PEEK(HL)
```

LD (HL),A ..... Load the memory location pointed to by HL with the value held in the A register.

```
10 LET A = 34  LET HL = 15360
20 POKE HL,A
```

LD (BUFFER),A ....Load the memory location labled BUFFER with value in A register.

() Brackets always mean from where, or the contents of.

If you wish to load any of the working registers from a single memory location, the HL pair must be used to transfer the data by loading them with the memory address. This does not, however, apply to the A register.

LD A,(BUFFER) .....O.K
LD C,(BUFFER) ..... WRONG !

```
LD HL,BUFFER        ;LD HL with memory address labled BUFFER.
LD C,(HL)           ;C now hold value from BUFFER.
```

Registers are loaded with **8 bit data** and register pairs can be loaded with **16 bit data** (immediate addressing).

```
LD C,#FF
LD BC,#FFFF
```

Always remember that 16 bit data is stored LSB\MSB. If BUFFER is at address #47A0 then:

```
LD HL,#3AF2
LD (BUFFER),HL
```

After the above code address #FA70 = #F2 and address #47A1 = #3A. The same rule applies when loading from memory:

```
LD HL,(BUFFER)  .........the HL registers will be loaded in reverse order
L = (#F2) ......Address #47A0
H = (#3A) ......Address #47A1
```

PROGRAM LISTING

```
1 PRINT "****************************************"
2 PRINT "************** B A C H *****************"
3 PRINT "****************************************"
4 PRINT "****************************************"
5 PRINT "******  PRELUDE IN C KLEINE TERTS *****"
6 PRINT "****************************************"
7 PRINT "****************************************"
10 LET RR=1: LET X=0
20 READ C1,C3,P
25 LET X=X+1
30 SBUF 5
40 SOUND 1,C1*4,15*64,RR,1,P,1
50 SOUND 0,(C1)*8,15*64,RR,-1,P,1
55 SOUND 2,C3*4,6000,RR,8,P*2,1
60 PAUSE P*40
61 LET CHAN=1
65 IF X=122 THEN  PAUSE 300: GOTO 80
66 IF X=243 THEN  PAUSE 300: GOTO 90
67 IF X=368 THEN  PAUSE 300: GOTO 90
68 IF X=493 THEN  PAUSE 300: GOTO 99
70 GOTO 20
80 GOSUB 95: PAUSE 1000: RESTORE 100: GOTO 20
90 GOSUB 95: PAUSE 1000: RESTORE 210: GOTO 20
94 REM****** uitschakelen geluid ******
95 IF PEEK(CHAN*10+64082)<>PEEK(CHAN*10+64082+4) THEN  GOTO 95
96 SOUND 1,0,0: SOUND 0,0,0: SOUND 2,0,0
97 RETURN
99 GOSUB 95: STOP
100 DATA   238,955,5,212,955,5,200,477,5,318,477,5,212,637,5,357,637,5,357,357,.25,401,601,4.75,425,601,5,477,1204,5,401,1204,5,318,318,5,238,238,5
110 DATA  357,601,5,200,601,5,212,715,5,178,715,5,150,851,5,238,851,5,238,238,.25,253,637,4.75,284,637,5,318,1275,5,253,1275,5,212,212,5,178,178,5
120 DATA  200,955,5,238,955,5,318,477,5,238,477,5,189,536,5,238,536,5,178,601,5,238,601,5,159,637,5,238,637,5,150,715,5,238,715,5
130 DATA  212,1072,5,268,1072,5,357,536,5,268,536,5,212,601,5,268,601,5,200,637,5,268,637,5,178,715,5,268,715,5,159,803,5,268,803,5
140 DATA  238,601,5,300,601,5,318,477,5,300,477,5,238,715,5,200,715,5,268,637,5,318,637,5,357,536,5,318,536,5,268,803,5,200,803,5
150 DATA  300,715,5,318,715,5,357,601,5,318,601,5,300,715,5,401,715,5,425,536,5,401,536,5,357,1072,5,425,1072,5,536,851,5,601,851,5
160 DATA  637,803,5,536,803,5,401,1606,5,536,1606,5,637,901,5,536,901,5,601,955,5,401,955,5,536,1072,5,401,1072,5,477,1204,5,401,1204,5
170 DATA  536,1275,5,401,1275,5,477,1431,5,401,1431,5,450,1606,5,401,1606,5,477,1204,5,477,803,5,401,601,5,401,637,5,300,715,5,300,803,5
180 DATA  300,851,5,300,715,5,357,536,5,357,601,5,212,637,5,212,715,5,212,637,5,212,536,5,268,401,5,268,450,5,159,477,5,159,536,5
190 DATA   159,601,5,178,601,5,200,536,5,212,536,5,238,1072,5,212,1072,5,200,803,5,268,803,5,318,803,5,268,536,5,803,401,5,803,401,5
200 REM *****Herhaling*****
210 DATA  159,803,5,150,803,5,134,401,5,159,401,5,134,450,5,178,450,5,178,178,.5,189,477,4.5,212,477,5,238,758,5,189,758,5,159,955,5,134,955,5
220 DATA  150,715,5,178,715,5,225,1072,5,536,1072,5,189,955,5,238,955,5,268,1431,5,318,1431,5,300,715,5,238,715,5,178,803,5,284,803,5
230 DATA  268,851,5,212,851,5,178,1072,5,150,1072,5,159,851,5,178,851,5,159,803,5,200,803,5,212,401,5,238,401,5,253,425,5,178,425,5
240 DATA  200,477,5,212,477,5,238,568,5,200,568,5,284,477,5,318,477,5,337,425,5,318,425,5,284,851,5,337,851,5,425,955,5,284,955,5
250 DATA  212,1072,5,318,1072,5,337,1136,5,238,1136,5,268,1275,5,318,1275,5,477,803,5,337,803,5,318,851,5,284,851,5,401,955,5,477,955,5
260 DATA   536,851,5,425,851,5,337,955,5,318,955,5,568,851,5,337,851,5,318,1275,5,318,851,5,268,637,1.5,253,637,1.5,268,536,1.5,253,536,1.5,268,450,1.5
270 DATA   253,450,1.5,268,715,1.5,253,715,1.5,268,758,5,225,758,5,189,955,5,238,955,5,159,758,5,268,758,5,300,715,5,268,637,5,238,601,5,300,601,5
280 DATA  357,357,5,379,379,5,357,1012,5,300,1012,5,253,1275,5,318,1275,5,212,1012,5,357,1012,5,401,955,5,357,851,5,318,803,5,401,803,5,477,477,5,506,506,5
290 DATA  477,1351,5,401,1351,5,337,1702,5,425,1702,5,284,1351,5,477,1351,5,506,1275,5,425,1275,5,357,851,5,300,851,5,318,803,5,253,803,5
300 DATA  212,1204,5,178,1204,5,200,1431,5,238,1431,5,318,1275,5,253,1275,5,238,955,5,318,955,5,401,955,5,318,637,5,477,955,5,477,955,5
```

This is definitely worth the effort of typing in all those data statements. The program was sent in by my old friend Nick Passmore from Holland. His mate Rene converted the music and they played it at the Dutch EXPO. Nick says,"Turn out the lights,light a candle and pour yourself a glass of your favourite plonk.

Experiment with differnt sounds by playing around with the parameters in lines 40,50 & 55. Don't get drunk!

ISSUE NUMBER 5

MEMOPAD the official magazine of GENPAT - Memotech Users Group

---

THE TAPE FROM THE MAGAZINE IS NOW AVAILABLE AT 3.50p

CONTENTS: WILLIE WORM : 3D ROTATOR : SEB BACH MUSIC BREAKOUT : PIXEL SCREEN SCROLL (FROM ISSUE 2) : ROCKET LANDER (FROM ISSUE 3)

MEMOPAD HELP LINE FACT SHEETS

RST 10 CALLS ..... HOW THE TO USE THESE SYSTEM ROUTINES. 50p

SYSTEM VARIABLES ... COMPLETELY UPDATED VERSION FROM OCTOBER 50p

INTERRUPTS .....EXPLAINED 60p

Mr G.P Cooke would like to contact any other member who owns a MTX computer and lives in the Essex area. His address is:-

301,Chigwell Road,
Woodfield Green,
Essex. IG8 8PL
Telephone; 01-505-0992

WOULD ANY MEMBERS WHO KNOWS HOW TO PROGRAM IN MICROSOFT OR TANDY BASIC AND WOULD LIKE TO WORK ON PROGRAM CONVERSIONS PLEASE SEND IN THEIR NAMES AND ADDRESSES IN ORDER THAT WE MAY CALL ON THEM TO HELP US.

THE CLUB IS LOOKING FOR SOMEONE WHO IS FAMILIAR WITH FORTH AND WOULD BE WILLING TO WRITE A SERIES OF ARTICLES ON THE SUBJECT IN PREPARATION FOR THE LAUNCH OF THIS LANGUAGE ON THE MTX.

SOFTWARE AUTHORS 

WE PAY A MINIMUM 1.00 ROYALTIES PER SALE FOR ANY PROGRAM THAT IS TAKEN UP BY SYNTAX SOFTWARE,SEND IN YOUR PROGRAM FOR EVALUATION. WE ARE LOOKING FOR THAT GAME OR UTILITY THAT HAS THAT SOMETHING EXTRA ..... WE PUBLISH I EVERY COUNTRY THAT SELLS MEMOTECH COMPUTERS. DON'T DELAY - SEND IT TODAY.

PROGRAM LISTING

ALPHASORT by I. NICHOLS

Alphasort(T) sorts text into a list of words in alphabetical order. Upon RUNing a flashing cursor will appear at the top left of the screen.Text may be typed in, but make sure to leave a space between words and after any punctuation marks. The screen reading routine will scan the entire screen or until it finds a "!". You will then be asked if you wish to enter another page (respond Y or N). The text will then be sorted and displayed as a list, along with the time taken for sorting. If the list is too long for a single screen, pressing any key will cause the list to scroll up. Pressing any key once the whole list has been displayed will reveal a menu with the following options: **a) display list again, b) add to the present list, c) enter a new set of pages for sorting.**

Unfortunately, as strings are compared using ASCII codes, upper case **"Z"** has a higher priority than lower case **"a"** and so words beginning with or printed entirely in block capitals will appear before words in lower case in the sorted list (ie. **ZERO** will appear before **Zarquon** and both will appear before **alpha**). Alas there is no simple way that I know of getting around this.

Alphasort(L) is a more conventional alphabetical sorter, in that each word is entered separately. A facility for correcting wrong entries is available by entering **'L.' (note the full stop)** to change the last word entered. Entering **'L.'** again will change the word before that and so on. As before **'!'** signifies the end of the list, and sorting will commence immediately after entering this character.

ALPHA SORT. T

```
10 VS 5: CLS
20 DIM LST$(400,64),T(34): LET W=0
30 PRINT CHR$(27);"X]";CHR$(27);"X^": CLS : CSR 1,0: DSI
40 GOSUB 1500
50 CLS : PRINT : PRINT "No of words so far:";W: PRINT : INPUT "Another page? ";ANS$
60 IF ANS$(1)="Y" OR ANS$(1)="y" THEN  GOTO 30
70 IF ANS$(1)<>"N" AND ANS$(1)<>"n" THEN  GOTO 50
80 CLOCK "000000": LET L=1: LET R=W
90 CLS : CSR 5,12: PRINT "YOUR TEXT IS NOW BEING SORTED"
100 CSR 10,17: PRINT "please be patient"
110 LET C=0: GOSUB 500: CLS : LET T$=TIME$
120 GOSUB 1000: PRINT
130 CSR 4,23: PRINT "end of list: it took ";T$(1,2);" ";T$(3,2);" ";T$(5,2);CHR$(11): PAUSE 1000
140 IF INKEY$="" THEN  GOTO 140
150 CLS : CSR 5,3: PRINT "Do you wish to"
160 CSR 5,7: PRINT "a) display list again?"
170 CSR 5,9: PRINT "b) Add to current list"
180 CSR 5,11: PRINT "c) enter more pages                    for sorting?"
190 IF INKEY$="A" OR INKEY$="a" THEN  CLS : GOTO 120
200 IF INKEY$="B" OR INKEY$="b" THEN  CLS : GOTO 30
210 IF INKEY$="C" OR INKEY$="c" THEN  CLS : CLEAR : GOTO 10
220 GOTO 190
300 CLS : CSR 5,5: PRINT "Sorry mate, no more room.": PAUSE 2000: GOTO 80
340 RETURN
490 REM ********************************      ***** sorting subroutine *****      ********************************
500 LET LT=L: LET RT=R: LET C$=LST$((LT+RT)/2)
510 IF LST$(LT)<C$ THEN  GOSUB 580
520 IF LST$(RT)>C$ THEN  GOSUB 590
530 IF LT<=RT THEN  GOSUB 600
540 IF LT<=RT THEN  GOTO 510
550 IF L<RT THEN  LET C=C+1: LET T(C)=R: LET R=RT: GOSUB 500: LET R=T(C): LET C=C-1
560 IF LT<R THEN  LET C=C+1: LET T(C)=L: LET L=LT: GOSUB 500: LET L=T(C): LET C=C-1
```

```
570 RETURN
580 LET LT=LT+1: IF LST$(LT)<C$ THEN  GOTO 580 ELSE  RETURN
590 LET RT=RT-1: IF LST$(RT)>C$ THEN  GOTO 590 ELSE  RETURN
600 LET T$=LST$(LT): LET LST$(LT)=LST$(RT): LET LST$(RT)=T$: LET LT=LT+1: LET RT=RT-1: RETURN
990 REM ############################        #### printing subroutine ####        ############################
1000 PRINT CHR$(27);"X\"
1010 FOR ITEM=1 TO W
1030 PRINT ITEM;")",LST$(ITEM)
1040 NEXT
1050 RETURN
1490 REM ############################        #### page reading routine ####        ############################
1500 LET N=W: CSR 0,0: LET CL=0: LET RL=0
1510 FOR R=0 TO 23
1520 FOR C=0 TO 38
1530 CSR C,R: IF SPK$=" " THEN  GOTO 1560
1540 CSR CL,RL: IF SPK$=" " THEN  LET N=N+1: LET N2=0: IF N>400 THEN  LET N=400: LET W=N: GOTO 300
1550 LET N2=N2+1: LET LST$(N,N2)=SPK$: IF LST$(N,N2)=";" THEN  LET N=N-1: GOTO 1590
1560 LET CL=C: LET RL=R
1570 NEXT
1580 NEXT
1590 LET W=N
1600 RETURN
```

ALPHA SORT L

```
10 VS 5: CLS
20 DIM LST$(400,64),T(32): LET N=0: LET C=0
30 LET N=N+1
35 CSR 5,10: PRINT "Item number";N;" please": CSR 7,14: INPUT ITM$: CLS
40 IF ITM$=";" THEN  GOTO 80
50 IF ITM$="L." OR ITM$="l." THEN  GOSUB 300: GOTO 40 ELSE  LET LST$(N)=ITM$: CSR 10,5: PRINT "Item";N;" is ";LST$(N)
60 IF N=400 THEN  CLS : LET N=401: PRINT "Sorry sunshine: no more data allowed": GOTO 80
70 GOTO 30
80 CLOCK "000000": LET LLEN=N-1
90 CSR 5,12: PRINT "YOUR LIST IS NOW BEING SORTED"
100 CSR 10,17: PRINT "please be patient"
110 LET L=1: LET R=LLEN: GOSUB 500: CLS : LET T$=TIME$
120 GOSUB 1000: PRINT
130 CSR 4,23: PRINT "end of list: it took ";T$(3,2);" ";T$(5,2);CHR$(11): PAUSE 1000
140 IF INKEY$="" THEN  GOTO 140
150 CLS : CSR 5,3: PRINT "Do you wish to"
160 CSR 5,7: PRINT "a) display list again?"
170 CSR 5,9: PRINT "b) Add to current list"
180 CSR 5,11: PRINT "c) enter another list                    for sorting?"
190 IF INKEY$="A" OR INKEY$="a" THEN  CLS : GOTO 120
200 IF INKEY$="B" OR INKEY$="b" THEN  CLS : GOTO 35
210 IF INKEY$="C" OR INKEY$="c" THEN  CLS : CLEAR : GOTO 10
220 GOTO 190
300 LET N=N-1
310 CLS : CSR 10,5: PRINT "Item";N;" is ";LST$(N)
320 CSR 5,10: PRINT "Change this to": CSR 5,13: INPUT ITM$: CLS
330 RETURN
490 REM ############################        ##### sorting subroutine #####        ############################
500 LET LT=L: LET RT=R: LET C$=LST$((LT+RT)/2)
510 IF LST$(LT)<C$ THEN  GOSUB 580
520 IF LST$(RT)>C$ THEN  GOSUB 590
530 IF LT<=RT THEN  GOSUB 600
540 IF LT<=RT THEN  GOTO 510
550 IF L<RT THEN  LET C=C+1: LET T(C)=R: LET R=RT: GOSUB 500: LET R=T(C): LET C=C-1
560 IF LT<R THEN  LET C=C+1: LET T(C)=L: LET L=LT: GOSUB 500: LET L=T(C): LET C=C-1
570 RETURN
580 LET LT=LT+1: IF LST$(LT)<C$ THEN  GOTO 580 ELSE  RETURN
```

```
1000 FOR ITEM=1 TO LLEN
1010 IF MOD(ITEM,22)=0 AND INKEY$="" THEN  GOTO 1010
1020 PRINT ITEM;")",LST$(ITEM)
1030 NEXT
1040 RETURN
```

```
590 LET RT=RT-1: IF LST$(RT)>C$ THEN  GOTO 590 ELSE  RETURN
600 LET T$=LST$(LT): LET LST$(LT)=LST$(RT): LET LST$(RT)=T$: LET LT=LT+1: LET RT=RT-1: RETURN
990 REM ############################        #### printing subroutine ####        ##########
```

REVIEW

SPELLI-COPTER & HELI-MATHS by ALAN STURGESS

The task in both programs is somewhat similar: you have to 'fly' a helicopter and collect the answer to a question. In **Spelli-Copter** a word is 'bombed' by a red helicopter. This knocks out one letter. You then have to collect this letter from an alphabet line at the foot of the screen. Once collected, you sit and watch as it is flown up to the damaged word. A short burble follows success whereas failure results in your helicopter assuming the flying characteristics of a brick. In **Heli-Maths** you are shown a sum and a line of landing pads. Each pad contains one of several possible answers. Your task is to land on the correct pad. Thereafter you become a passive observer (as in **Spelli-Copter**).

Unfortunately, I can see a lot of weaknesses in these programs. Of the two, **Spelli-Copter** is the weakest:

1) 116 'word sets' are available, but no clue is given as to what constitutes each set. (There is a crafty way to find out - try holding down **ESCAPE** at the point where you are asked to select a word set. When the screen begins to flicker, **RETURN** and **LIST**).

As examples, sets 1-5 contain the 5 vowels; sets 11-15 cover sh,ch,th,wh,ph; set 25 contains 'qu' words.....

2) The program never collects more than one letter. Digraphs and dipthongs are totally ignored - even though they form the basis of many sets. (A landing pad format could have been employed at later levels of play).

3) The treatment of errors is appalling and relies entirely upon that old terror - the closed loop. Talk about traditional skills and rote learning!!.

4) The action is very slow - a 26 second-cycle when collecting the letter **'a'**. The user spends a long time passively watching and waiting.

In my opinion, **Spelli-Copter** is of use and interest to younger children, but will be of limited use with older or more able children.

Heli-Maths contains several of the same weaknesses as its stablemate - but it is generally the more interesting program. It could easily be used in conjunction with mental work, pencil and paper or calculators. Its biggest (and most serious) weaknesses is the lack of levels of difficulty. There is no way to specify, say, addition bands up to 10, 20 or 50 - or times tables only up to 4, 5 etc. Had this been allowed, I would have felt able to give this program an enthusiastic thumbs-up. As it stands, I rate it somewhere between **fair** and **quite good** if used by older or more able children. It is of little use with young or less able children - but it **is** worth a look.

HIGH SCORES : HIGH SCORES...Can you do better ??

GOLDMINE	4,543	Richard Nash
ASTRO-PAC	69,390	Alan Dobson
BOUNCING BILL	14,184	Alan Dobson
SNAPPO	107,430	Richard Franks
KNUCKLES	360,200	Sean Haverty
NEMO	17,610	Richard Nash
COBRA	8,924	Richard Nash
MISSION ALPHATRON	49,500	Christian Burnel
TAPEWORM	86,070	Alan Dobson
TOADO	58,000+	James & Vanessa (twins) Gover
POT HOLE PETE	39,630	Alan Dobson
MAXIMA	159,000	Daljinder Singh
STAR COMMAND	70,400	Sean Haverty
PHAID	23,470	Eric Peters
OBLOIDS	46,850	Sean Haverty
KILOPEDE	58,569	Richard Nash
3D TACHYON FIGHTER	6,130	Sean Haverty
CONTINENTAL RAIDERS	106,240	Sean Haverty
BLOBBO	72,514	Richard Nash

Can you beat these high scores ? Do you have a high score for a game not mentioned above ?

ISSUE NUMBER 5

MENOPAD the official magazine of GENPAT - Memotech Users Group

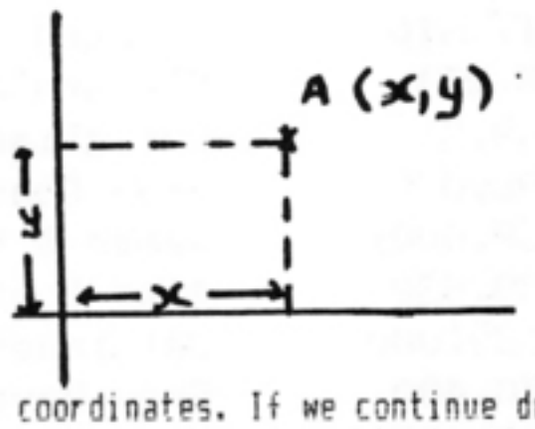
JR MAJOR

This is the first of a series of articles which will lead to routines handling 3-D graphics. The first article deals with how a 2-D image is defined, how you make that image shrink, expand, twist, and moving the image from one part of the screen to the other. Another technique introduced is *clipping* - how we remove those lines or parts of lines which fall outside our screen and which would make our program crash if we tried to plot them.

A few definition of terms. SCALING is the term for making the image larger or smaller. TRANSLATION is the term usually used for moving the object from one part of the screen to another and ROTATION is ....well, rotation. All these are implemented in our routines and instead of giving you the routines like stone tablets handed down the mountainside, we will build them up so you can see how they work and so be able to adapt them your needs. The routines are all in basic so they will reach the widest audience, but it will become clear that in order to manouevre a complex object in three dimensions on the screen, while Basic will handle it, machine code routines are necessary if you want to achieve acceptable speed. The computer generated images you see in adverts and in films of an exploded parts view of a car rotating in three dimensions and reassembling itself is more likely to be carried out on a Cray supercomputer than on an MTX. And not in interpreted basic. Still, you can achieve worthwhile aims on our system and in Basic.

To handle the 3-D images we will be using matrix maths and again this will be introduced step by step. If you follow these articles you will be equipped with techniques to enable you to go further, not just a routine to produce a pretty cardiod or rotate a wire cube in 3-D. The clipping introduced here is a preliminary to methods for removal of hidden lines and surfaces.

It is assumed everyone understands the idea of coordinates. Point A on the diagram below is defined by the coordinate pair x,y. If we want to draw to or from this point, or plot it, that's how we refer to it.



A line will obviously need two pairs of coordinates. If we continue drawing a shape by storing the coordinates of the vertices then we can store these points in two arrays x and y and thus be able to handle the plotting within loops using the loop index to locate ourselves in the array -

```
10 dim x(400):dim y(400)
```

```
20  data  4,20,20,100,20,180,20,20,20
30  read  numpts
40  for  n=  1  to  numpts: read x(n),y(n):next n
50  crvs 2,1,0,0,32,24,32:vs 2: cls
60  for  n=1 to numpts : line x(n), y(n),x(n+1), y(n+1): next n
70 if inkey$() "R" and inkey$()"r" then goto 70
```

Now you change the data statement and define your own pictures in lines - notice that the first item in the data statement tells the program how many points there are so that this can be used as a loop index whenever we are handling the data. The idea is that you adapt this program - store a library of images so you can call them to the corner of the screen, move them to the right position on the screen, stretch, twist shrink or change the shape till it is right and so compose a final picture from related elements - which could be electronic components or motifs from Turkish knitting patterns. Notice we try to do only plotting in the plot loop as if we do all calculations outside this loop where possible we will speed up our plotting.

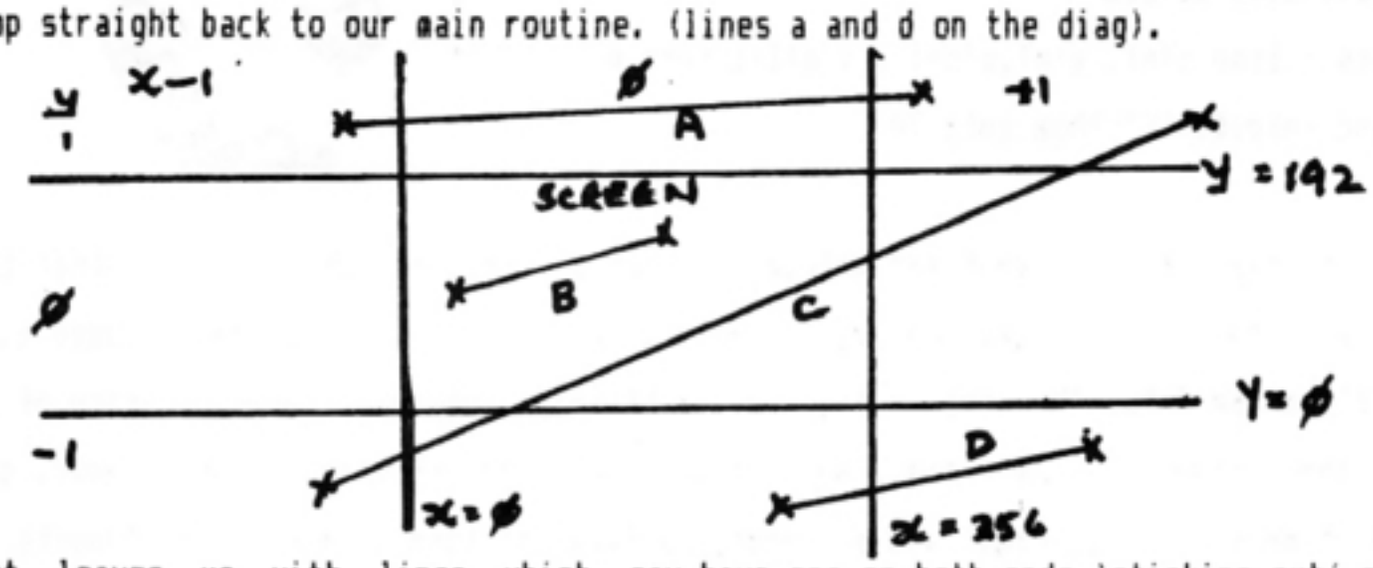
Suppose now we want to shrink or expand our image. If we multiply x and y by 0.1 we will get an image 10% of the original size. We set up variables XSCALER and YSCALER. Insert instructions to input these variables and a loop to multiply each x and y by the appropriate figure before plotting. You will find this done in the listing. Obviously, if we put a small x scaler and a large y our image will stretch up the screen. Play around a bit - you will soon find the reason for the clipping routine as your loop produces points to be plotted outside the screen and comes up with the typically useful MTX error message SE.B illegal escape sequence. Someone in Memotech has a sense of humour. Just before we add a clipping routine to deal with this problem, we add a way of moving our image around on the screen. Instead of multiplying, if we add an offset to each point we can move the whole image around on the screen by doing no more than changing the offset. Your loop setting this up should now be -

```
for  n  =  1  to  numpts
let  xplot(n)=x(n)*xscaler  +  xoffset
let  yplot(n)=y(n)*yscaler  + yoffset
next n
```

Notice we preserve our original image from the library of images we build by using another array for the manipulated data.

Now the clipping routine. This is done in the larger of the listings. Look at the diagram below and you will see that if a point falls outside the screen it can be classified into one of eight areas. In the routine which does this 1,-1 or 0 are assigned to tell the program which area of the screen each point falls into. If all the x and y's for both ends of the line fall inside the screen area we plot and jump back.(line b on the diag) If the x for both ends of the line or the y for both ends of the line have 1 or -1 assigned to

them - look at the drawing and you will see that these lines fall completely outside the screen so we can jump straight back to our main routine. (lines a and d on the diag).



So that leaves us with lines which may have one or both ends 'sticking out' either side of the screen. (line c on the diagram). We drop down through the rest of the clipping routine looking to see which end is outside the screen, sliding the point back to the edge of the sscreen. Then we jump back to reclassify the new ends of line to see is they are OK for plotting now. If they are then we plot and return. If not we take another pass through the clipping routine. Obviously you caan make up to four passes before you get out- clipping the ends four times. Look at the diagram. If you find this sequence of program confusing - rewrite it! Don't worry if this or the matrix maths give you a few problems at first - God must like the not-so-smart people bettter - that's why he made more of us. It certainly took me a bit longer to work out than I expected.

So now we have the means to create an image, move it around the screen, shrink or expand it. We also remove lines or parts of lines outside the screen. Suppose however we want to use a screen smaller than the whole screen - we tell our clipping routine and it will handle it and we use our scalers to get the image down to size. So now we have to input screen dimensions as well. Let's set this up so that we can ddo this off a menu and so avoid setting up the same screen again and again. Notice the establishing of the default values - ie the program assumes x and yscalers of 1, a screen size of the full size possible and no x and y offsets unless you specify them from the program. Alter these to what you want.

Now mention was made of the matrix. Some examples of matrices are given below.

A_{11} A_{12} B_{11} B_{12} B_{13}
A_{21} A_{22} B_{21} B_{22} B_{23}

Wheree the first subscript refers to the row and the second subscript refers to the coluan.

We can set up some of the work we have done above in a matrix:

$$(x \quad y) \quad * \quad \begin{pmatrix} a & d \\ c & b \end{pmatrix} = \quad (x' \quad y')$$

Where x' y' are our new coordinates after having scaled. Mathematical operations on matrices are carried out in accordance with certain rules. We want to multiply these matrices so, using the same letters as above x' y'= x*a + y*c, x*d +y*b

The way this works is row member in the first matrix times column member in the second matrix. Try a few more yourself as we need these operations to handle the 3-D graphics maths. Implementing this in our program, we now have xplot(n)=x(n)*a +y(n)*c and yplot(n)=x(n)*d + y(n)*b

The a and b correspond to our xscaler and yscaler, but what about the c and d? Set up a routine to input these - call them xshear and yshear and you will find that they give your image a twist, distorting it round one way or the other.

This distorting can be done methodically and rotate the image. The matrix for this is given below.

(cos theta sin theta)

(-sin theta cos theta)

Slip these values into xscale, yscale ,xshear,yshear and you will rotate your object through the angle theta - the angle is measured in radians - explained on p. 101 of you MTX manual. 2*PI radians is 360 degrees,PI radians is 180, PI/2=90, PI/4=45 etc. You will find this implemented in the "rotate" option on the menu of the program listing and it is worth understanding this to work it in three dimensions and spin your spacecraft or turn your yacht design.

Now at the moment we don't have translation on our matrix. Add another row to the matrix and we get this:

1 0 0

0 1 0

LR UD 1

And LR is left/right (our xoffset) and UD is up/down(our yofffset) So, to keep our matrix operations straight we need to add a third point to our x,y coordinates - a variable we shall keep as one - you define your x, y as 2,4 or using the third number as 2,4,1 or 4,8,2 or 1,2,0.5 - so you can see what we are doing is having three points p,q,r such that x=p/r and y=q/r. This is the rocky road to 3-D graphics.We can keep this third point we are adding as 1 always and think of it as being just necessary for the matrix operation. Just keep playing with the program and trying your own figures in and the light dawns. Our matrix has now become:

(x y 1)* (a^d^g) = xplot,yplot

(b^e^h)

(c^f^i)

Where you can superimpose our old matrices to identify which of these is xscaler, yshear or our offsets.We multiply them out to get: xplot,yplot,1= x*a +b*y +c*1, x*d + y*e +f*1 , g*x +h*y +I*1.

Substituting in our matrix

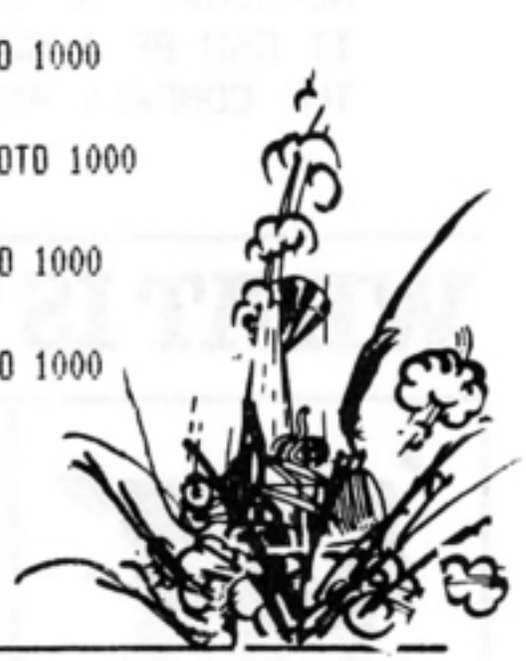
(1 0 0)
(0 1 0)
(lr ud 1)

this gives us back the routines we used - which you must expand to allow for all these variables. In next months article you will see how the handling of these routines in matrices makes it conceptually easier to handle 3-D graphics.

One point remains - different screen sizes - using the crvs command we can make windows with the MTX. This is the reason for the SW and SH variables - screen height and screen width and this will allow us to set up our images for a different size graphics screen. Well, that'll do for this months - next month we enter the third dimension. This article was written on a cassette based wordprocessor written by the author for the MTX512. The wordprocessor is available from Genpat as is a home accounts program. A graphics package with pie charts, bar charts, graphing facilities etc will be out shortly with indexing and so will the first tape of a series for learning Spanish on your MTX. Happy programming!

```
2 REM graphics handling progam
4 REM set up screen
5 CRVS 2,1,0,0,32,24,32: VS 2: CLS
9 REM set up arrays
10 DIM X(400): DIM XPLOT(400): DIM XCLIP(2): DIM SX(2)
11 DIM Y(400): DIM YPLOT(400): DIM YCLIP(2): DIM SY(2)
12 LET XSHEAR=0: LET YSHEAR=0: LET XOFFSET=0: LET YOFFSET=0: LET SW=250: LET SH=190: LET XSCALE=1: LET YSCALE=1
13 CLS : PRINT "2-D graphics menu"
14 PRINT "~~~~~~~~~~~~~~~~~~": PRINT "1. input x scale": PRINT "2. input y  scale": PRINT "3. input x offset"
15 PRINT "4. input y offset": PRINT "5. input x shear": PRINT "6. input y shear": PRINT "7. input screen width":
PRINT "8. input screen height": PRINT "9. plot"
16 PRINT "10. rotate": PRINT : INPUT "input number of reqd. operation";OP
17 IF OP<1 OR OP>10 THEN  GOTO 13
18 ON OP GOTO 14,30,40,50,60,82,84,70,80,90,85
30 CLS : INPUT "INPUT XSCALE";XSCALE: GOTO 13
40 CLS : INPUT "INPUT YSCALE";YSCALE: GOTO 13
50 CLS : INPUT "INPUT X OFFSET";XOFFSET: GOTO 13
60 CLS : INPUT "INPUT Y OFFSET";YOFFSET: GOTO 13
70 CLS : INPUT "INPUT SCREEN WIDTH";SW: GOTO 13
80 CLS : INPUT "INPUT SCREEN HEIGHT";SH: GOTO 13
82 CLS : INPUT "input x shear";XSHEAR: GOTO 13
84 CLS : INPUT "input y shear";YSHEAR: GOTO 13
85 CLS : INPUT "input angle of rotation in PI radians";ROT: LET ANG=PI*ROT: LET XSCALE=COS(ANG): LET YSCALE=COS(ANG)
: LET YSHEAR=-SIN(ANG) : LET XSHEAR=SIN(ANG)
86 GOTO 13
89 REM x,y coordinates of the vertices of object to be plotted
90 DATA  5,20,20,20,100,100,100,100,20,20,20
95 CLS
100 RESTORE 90
110 READ NUMPTS
119 REM read in pts and calculate plot points
120 FOR N=1 TO NUMPTS
130 READ X(N),Y(N)
140 LET XPLOT(N)=INT(X(N)*XSCALE)+INT(Y(N)*XSHEAR)+XOFFSET
150 LET YPLOT(N)=INT(Y(N)*YSCALE)+INT(X(N)*YSHEAR)+YOFFSET
160 NEXT N
```

```
199 REM clip and plot
200 FOR Q=1 TO NUMPTS-1
205 FOR Z=1 TO 2: LET XCLIP(Z)=0: LET YCLIP(Z)=0: NEXT Z
210 FOR N=1 TO 2
219 REM check what needs clipping
220 LET SX(N)=XPLOT((Q-1)+N)
230 LET SY(N)=YPLOT((Q-1)+N)
240 IF SX(N)>SW THEN  LET XCLIP(N)=1
250 IF SY(N)>SH THEN  LET YCLIP(N)=1
260 IF SX(N)<1 THEN  LET XCLIP(N)=-1
270 IF SY(N)<1 THEN  LET YCLIP(N)=-1
280 NEXT N
290 LET P=1: GOSUB 1040
300 NEXT Q
400 IF INKEY$<>"R" AND INKEY$<>"r" THEN  GOTO 400
409 REM jump back to menu when r key pressed
410 GOTO 13
1000 LET XCLIP(P)=0: LET YCLIP(P)=0
1005 IF SX(P)>SW THEN  LET XCLIP(P)=1
1010 IF SY(P)>SH THEN  LET YCLIP(P)=1
1020 IF SX(P)<1 THEN  LET XCLIP(P)=-1
1030 IF SY(P)<1 THEN  LET YCLIP(P)=-1
1039 REM if the line is outside the screen return without plotting
1040 IF XCLIP(1)*XCLIP(2)=1 OR YCLIP(1)*YCLIP(2)=1 THEN  GOTO 300
1059 REM if the line is now within screen then plot and return
1060 IF XCLIP(1)=0 AND YCLIP(1)=0 THEN  LET P=2: IF XCLIP(2)=0 AND YCLIP(2)=0 THEN  LINE SX(1),SY(1),SX(2),SY(2): RETURN
1069 REM if x less than 1 then clip back to 1
1070 IF XCLIP(P)=-1 THEN  LET SY(P)=SY(1)+(SY(2)-SY(1))*(-SX(1))/(SX(2)-SX(1)): LET SX(P)=1: GOTO 1000
1079 REM if x greater than 250 then clip back to 250
1080 IF XCLIP(P)=1 THEN  LET SY(P)=SY(1)+(SY(2)-SY(1))*(SW-SX(1))/(SX(2)-SX(1)): LET SX(P)=SW: GOTO 1000
1089 REM if y greater than 190 then clip to 190
1090 IF YCLIP(P)=1 THEN  LET SX(P)=SX(P)+(SX(2)-SX(1))*(-SY(1))/(SY(2)-SY(1)): LET SY(P)=SH: GOTO 1000
1099 REM if y less than 1 then clip back to 1
1100 IF YCLIP(P)=-1 THEN  LET SX(P)=SX(1)+(SX(2)-SX(1))*(-SY(1))/(SY(2)-SY(1)): LET SY(P)=1: GOTO 1000
1140 LET XPLOT(N)=INT(X(N)*XSCALE)+INT(Y(N)*XSHEAR)+XOFFSET
1150 LET YPLOT(N)=INT(Y(N)*YSCALE)+INT(X(N)*YSHEAR)+YOFFSET
```



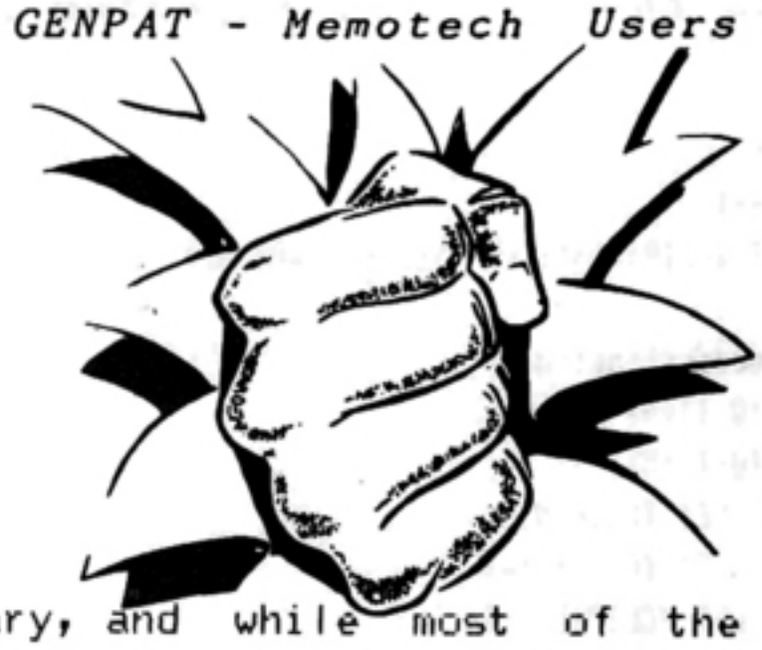
Competition

ONE LINERS

```
10 LET F=1: INPUT "FACTORIAL ? ";N: IF N
>33 OR N<>INT(N) THEN  GOTO 10 ELSE  PRI
NT N;"! =";: FOR X=N TO 1 STEP -1: LET F
=F*X: PRINT X;"*";: NEXT : PRINT " 1=";F
: GOTO 10
20 REM ************************************
30 REM ** 1 LINE FACTORIAL PROGRAM *****
40 REM ** D.J.ELLIOTT   23/12/84   *****
50 REM ** WITH   CHECK FOR NUMBERS *****
60 REM ** TOO GREAT & NON-INTEGER  *****
70 REM **    USE ABBREVIATIONS TO  *****
80 REM **         ENTER CODE       *****
90 REM ************************************
```

ISSUE NUMBER 5

MEMOPAD the official magazine of GENPAT - Memotech Users Group



COME ON MEMOTECH GIVE US A BREAK

It is now almost the end of January, and while most of the other companies are either brawling in clubs, cutting prices, and even advertising on television, Memotech seem to have gone one better and **stopped advertising altogether!**

The machine failed to make an impact on the market during the latter part of the year - not the computer's fault. There was a massive hole in the home computer market during the last four months of the year, and the time was ripe for the engineering excellence of the MTX to step in and capitalise on the fact that most manufacturers had only re-hashed their old machines. However, as most of you know, this didn't happen. Instead, a company with a reputation for shoddy goods, and unreliable equipment utilised one of the best marketing promotions ever seen in this country, with the result that the Amstrad computer became a household name.

It is now time for the marketing people at Memotech to show a little loyalty to the people who have supported the company by purchasing the machine. **SO COME ON MEMOTECH - WE SHOWED FAITH IN YOUR PRODUCT, WE KNOW IT CAN BE A NUMBER ONE SELLER - GIVE US A BREAK AND START SHOUTING ABOUT THE COMPANY AND THE MTX !**

WHAT IS THE MAN SAYING?

HACKERS TIPS

Here's an interesting hack for **Maxima** from Chris Sawyer :-

POKE 35725,6 and POKE 35729,6 (for the MTX500)
POKE 19341,6 and POKE 19345,6 (for the MTX512)

Here are a number of hackers tips from Colin Rees :-

Load **TOADO**, press RESET and type...

MTX500	MTX512
POKE 51731,no. of lives	POKE 35347,no. of lives
POKE 49232,243	POKE 32848,243
LET A=USR(49232)	LET A=USR(32848)

Load **AGROVATOR**, press RESET and type...
POKE 16619,no. of lives
POKE 16640,no. of bullets
LET A=USR(16408)

Load **POT HOLE PETE** and just before the loader program has loaded there will be a high pitched tone, press BRK, stop the tape and type....

MTX500	MTX512
POKE 32804,201	POKE 16420,201
2 STOP	2 STOP

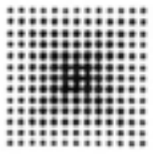
RUN <RET> and press play on tape. After the program has loaded type...

POKE 51979,0	POKE 35595,0
POKE 51980,0	POKE 35596,0
POKE 52103,no. of lives	POKE 35719,no. of lives
LET A=USR(49408)	LET A=USR(33024)

Here is a tip for you Single FDX system owners :-

This simple tip can be used to good effect when loading e.g. NEWWORD
1 PAPER n: USER RUN "SNW.RUN"
By doing this NEWWORD will retain n background colour.

ADVENTURE HELP LINE



Christian Burnel would like some help with **Alice in Wonderland.** He can't take the boat SW at the beginning of the game. Also he is blocked in the Duchesses front garden, in the Queen of Hearts' garden and he can't get passed a gateway with a cheering sound coming from the North (the gateway is near the Mad March and Grimbley Forest).
Tom Duggan cannot get out of the ditch with the sword - when he enters "climb ladder" it responds with "no such command". He can't out of Grimbley Forest after getting the warrant. Do you have to start at the beginning every time after saving the game ?
Please tell us if you can help with these problems. If you are having difficulties with the adventure games available on the Memotech then we will publish any queries so that other members can give you advice.

PROGRAM LISTING

BREAKOUT

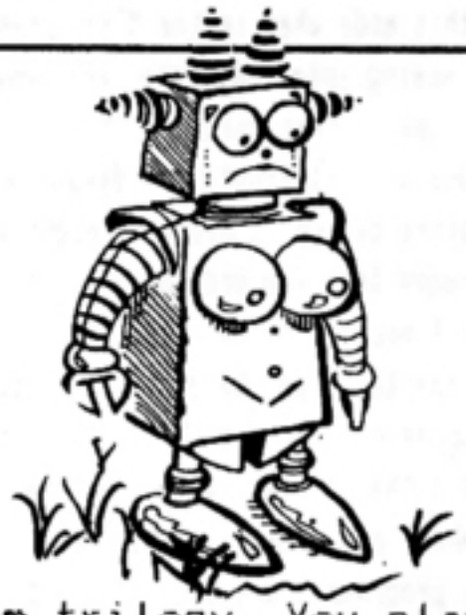
```
10 REM
100 REM
110 REM
120 REM @ John Grayson, 1984
130 REM ‡  AMPFIELD, HANTS  ‡
140 VS 5: PAPER 15: CLS : PAPER 15: INK 7: CSR 15,0: PRINT "BREAKOUT": CSR 15,1: PRINT "**********": PRINT : PRINT "This game is a version of one of the"
150 PRINT "first arcade games.  The object of the game is to demolish the wall block by  block until the wall is destroyed."
160 PRINT : PRINT "This is done by bouncing the 'ball' on the bat on the base of the screen.  If a wall is knocked down, then another isbuilt."
170 PRINT : PRINT "You have only three lives to complete  this task.  On the top of the screen isdisplayed both the score and lives left"
180 CSR 5,22: PRINT "press any key to continue..."
190 IF ASC(INKEY$)<8 THEN  GOTO 190 ELSE  CLS : PRINT "The following keys can be used during the game:": PRINT : PRINT
200 PRINT "P - Pauses the game until P is pressed again.": PRINT : PRINT "The left and right keys to move the bat": CSR 6,22: PRINT "press any key to start.."
210 IF ASC(INKEY$)<8 THEN  GOTO 210 ELSE  CLS
220 LET SCORE=0: LET LIVES=3
230 LET POS=INT(RND‡2): IF POS=0 THEN  LET H=12: LET D=1
240 IF POS=1 THEN  LET H=244: LET D=4
250 LET V=12: LET I=2: LET DIR=0: LET CX=112: LET BLOCK=0: LET PH=4
260 VS 4: PAPER 1: CLS : PAPER 1: CTLSPR 0,1: CTLSPR 2,3: CTLSPR 6,1: GENPAT 3,1,0,0,6,6,0,0,0,0: GENPAT 1,129,255,255,255,255,255,255,255,255
270 GENPAT 3,2,0,0,255,0,0,0,0,0: FOR Y=5 TO 10: LET I=I+2: INK I: FOR X=0 TO 31: CSR X,Y: PRINT CHR$(129);: NEXT : PRINT : NEXT
280 CSR 5,1: PRINT "SCORE:";SCORE: CSR 19,1: PRINT "LIVES:";LIVES
290 SPRITE 1,2,CX,6,1,1,13
300 ON D GOTO 300,310,330,350,370
310 LET V=V+4: LET H=H+PH: SPRITE 3,1,H,V,1,0,9: IF H<256-PH AND V<187 THEN  GOTO 380 ELSE  IF H>255-PH THEN  LET D=4 ELSE  LET D=2
320 GOTO 380
330 LET V=V-4: LET H=H+PH: SPRITE 3,1,H,V,1,0,9: IF H<256-PH AND V>3 THEN  GOTO 380 ELSE  IF H>255-PH THEN  LET D=3 ELSE  GOTO 580
340 GOTO 380
350 LET V=V-4: LET H=H-PH: SPRITE 3,1,H,V,1,0,9: IF H>-4+PH AND V>3 THEN  GOTO 380 ELSE  IF H<-3+PH THEN  LET D=2 ELSE  GOTO 580
360 GOTO 380
370 LET V=V+4: LET H=H-PH: SPRITE 3,1,H,V,1,0,9: IF H>-4+PH AND V<187 THEN  GOTO 380 ELSE  IF H<-3+PH THEN  LET D=1 ELSE  LET D=3
380 IF V<20 OR V>170 THEN  LET DIR=0
390 IF V=8 THEN  GOTO 540 ELSE  IF DIR=1 OR V<103 OR V>142 THEN  GOTO 450: FOR T=102 TO 142 STEP 8: IF T=V THEN  GOTO 400: NEXT : GOTO 450
400 IF V>142 THEN  LET CHAR=25 ELSE  LET CHAR=23
410 CSR INT(H/8),INT(CHAR-(V/8)): IF ASC(SPK$)=32 THEN  GOTO 450 ELSE  CSR INT(H/8),INT(CHAR-(V/8))
420 PRINT " ": FOR T=1 TO 15: SOUND 1,400,15: NEXT : SOUND 1,0,0
430 LET DIR=1: IF D=1 THEN  LET D=2 ELSE  IF D=2 THEN  LET D=1 ELSE  IF D=3 THEN  LET D=4 ELSE  LET D=3
440 LET BLOCK=BLOCK+1: LET SCORE=SCORE+(ABS(INT(CHAR-(V/8))-11))‡10: IF BLOCK=192 THEN  GOTO 230
450 CSR 11,1: PRINT SCORE: LET BAT=ASC(INKEY$): IF BAT=8 OR BAT=25 THEN  GOTO 500
460 IF BAT<>80 THEN  GOTO 480 ELSE  CSR 13,14: PRINT "Pause": PAUSE 2000
470 IF INKEY$<>"P" THEN  GOTO 470 ELSE  CSR 13,14: PRINT "     ": PAUSE 300
480 IF PH=8 THEN  PAUSE 25 ELSE  IF PH=6 THEN  PAUSE 30 ELSE  PAUSE 35
490 GOTO 300
500 IF BAT=25 THEN  GOTO 520 ELSE  LET CX=CX-6: IF CX<6 THEN  LET CX=CX+6
510 GOTO 290
520 LET CX=CX+6: IF CX>251 THEN  LET CX=CX-6
530 GOTO 290
540 IF H>CX-14 AND H<CX+7 THEN  GOTO 550 ELSE  GOTO 300
550 IF H>CX-4 THEN  LET D=1 ELSE  LET D=4
560 IF H<CX-10 OR H>CX+2 THEN  LET PH=8 ELSE  IF H<CX-7 OR H>CX-1 THEN  LET PH=6 ELSE  LET PH=4
570 GOTO 300
580 LET LIVES=LIVES-1: CSR 25,1: PRINT LIVES: IF LIVES>0 THEN  PAUSE 2000: LET CX=120: LET D=1: LET H=12: LET V=12: LET PH=4: GOTO 290
590 INK 7: CSR 9,14: PRINT "‡  The End  ‡": INK 9: CSR 9,16: PRINT "Another game?"
600 IF INKEY$="Y" THEN  GOTO 220 ELSE  IF INKEY$="N" THEN  STOP  ELSE  GOTO 600
```

REFERENCE

STAR COMMAND UPDATE by DAVID GLOVER

In **Star Command** two important keys are not shown on the 3 instruction screens which appear when you start a game. The first is the 'O' key which causes an options screen to be displayed and lists all the available keys, which is very useful. The other key, which is listed on the options screen, is the 'F' key which returns you to a forward view out of the cockpit, after you have been looking at the options screen, long-range scanner or galactic map. Without knowing the existence of the forward-view key, the only way to return to the cockpit view is to move to a new sector. The galactic map shows all presence of enemy ships an friendly star bases - your position is also shown. By using the cursor keys (or joystick) you can position your ship on a new sector: operating the hyperdrive will take you there.

REVIEW



RETURN TO EDEN by NIGEL CHURCH

Return to Eden is the second in **Silicon Dream** trilogy. You play the role of **Kim Kimberley**, who, whilst trying to save **Snowball 9** from destruction, appeared to hurl a bomb into a room. It is for this that you are accused of murder and the crew of **Snowball 9** are 'out to get you' for murder. You manage to escape in a stratoglider life-boat and crash on the planet **Eden.** It is here that **Return to Eden** starts.

When the game starts you have a limited number of moves before the sky-hook descends to inform you that you are guilty of murder and that your sentence of death will be carried out. Then after a few more moves the **Snowball 9** turns its engines on you and you are destroyed with a score of 0 out of 1000. This is very off putting for no matter how many times I played I was killed before I could really get into the game.

Return to Eden is no doubt up to **Level 9's** standards, it's just that I could not get far enough into it to tell. I can not recommend **Return to Eden** to any beginners at adventure games.

ISSUE NUMBER 5

MEMOPAD the official magazine of GENPAT - Memotech Users Group

PROGRAMMING TECHNIQUE

PLEASE INTERRUPT!

JOHN MULLINS

Last month we discussed the ways in which the programmer can use interrupts via the MTX operating system, this month we will move onto the more complicated topics of the I vector and the Z80 CTC. However, before we do that let me clear up the gremlin that sneaked into last months article. On page 12 of Memopad you are told that in order to activate the clock you should **POKE 64862,13**, this is wrong and in actual fact should be **POKE 64862,31**. Right let's get on with it.

Unlike most other microprocessors the Z80 has a special register which is totally concerned with interrupts. It is unusual in that it is impossible to put a value directly into this register but has to be done indirectly via the accumulator thus:

```
LD A,VALUE      ;Put value for I into A
LD I,A          ;Now put into I
```

This register is used when the Z80 is operating in interrupt mode II (the Z80 operates in this mode when in the MTX). When the Z80 receives an interrupt, assuming interrupts are enabled, it issues an acknowledgement and the current value of the program counter is saved onto the stack and interrupts are disabled. It is now the job of the interrupting device to place an eight value on the data bus, this then becomes the low order eight bits of a sixteen bit address with the I vector forming the high order eight bits. The CPU then puts the two bytes held at this address and the one above into the program counter (in LSB, MSB format), and then goes off and executes the code at this address. Two important points to remember are that interrupts must specifically re-enabled by the programmer and when the interrupt servicing routine is complete return to the main program must be executed by a **RETI** instruction to indicate to the interrupting device that it has been serviced. This may all sound very complicated, but this is not so and is best explained by way of an example.

Assume the I register has the value #FF, and the CPU receives an interrupt. The instruction currently being executed is completed, the interrupt acknowledged, the current value of the program counter is saved on the stack and interrupts are disabled. The device which caused the interrupt now deposits a value on the data bus (let's assume that #F0 is placed there), this is now appended to the I vector to produce an address (this will be #FFF0). Now the program counter is loaded from this address and off the **Z80** trundles to service the interrupt. This is exactly what happens when the MTX is operating, the address held at #FFF0 is #0780 and examination of the code here reveals it to be the interrupt handler discussed last month.

From the above it can be seen that it is possible to have 128 different devices which can cause an interrupt, and our table of jump addresses can be located anywhere in memory, providing it starts on a 256 byte boundary (commonly called a page boundary). On the MTX this is greatly simplified since we only have four interrupting devices, these are the four channels of the **Z80 CTC**. The **CTC** is a device which does the job of precisely counting or timing certain events (hence it's name, Counter Timer Circuit). The **CTC** has four channels each of which can be programmed to count a specific number of events or allow a specific amount of time to pass, upon completion of the count or time lapse the **CTC** will generate an interrupt. It is also possible to determine what value the **CTC** will deposit on the data bus after the interrupt acknowledge cycle, this is done thus: the vector is written to channel 0 of the **CTC** with bit 0 reset and the value each cahnnel deposits in the data bus is calculated thus:

Value = Vector + 2*channel number

The three low order bits of the interrupt vector are set to zero, so our table of jump addresses must sit on an eight byte boundary. Each channel of the **CTC** has two registers and two counters (all eight bit), these include a channel control register, an eight bit time constant register, a readable down counter, and a prescaler. The control register determines the operation of the channel, whether it is used as a counter or a timer, whether the prescaler is used or not. The time constant holds a value loaded under software control which is used to time or count a specific number of events. The prescaler is used to divide the system clock by 16 or 256.

In order to select an operating mode for a particular channel of the **CTC** we must write to the particular channel a byte with bit 0 set, the remaining seven bits are used to indicate the various modes of operation thus:

Bit 7=0 Channel interrupts are disabled

Bit 7=1 Channel interrupts are enabled to occur every time the down counter reaches zero

Bit 6=0 Timer mode is selected. The down counter is clocked by the prescaler. The period of the counter is:

T*P*TC

Where T=system clock period
P=value of prescaler (either 16 or 256)
TC=value of time constant

Bit 6=1 Counter mode is selected. The counter is clocked by the system clock and the prescaler is not used

Bit 5=0 This indicates prescaler=16

Bit 5=1 This indicates prescaler=256

Bit 4=0 In both timer and counter mode a negative edge trigger (i.e. a switch off) decrements timer or counter.

Bit 4=1 A positive edge trigger decrements counter or timer

Bit 3=0 Used in timer mode only, indicates timing starts immediately after loading of a time constant

Bit 3=1 Used in timer mode only, indicates timing to begin as soon as a trigger is received

Bit 2=0 Indicates no time constant will follow the channel control word, a time constant must be written to the channel to initiate operation.

Bit 2=1 The next word written will be the down counter.

Bit 1=0 Channel continues counting

Bit 1=1 This stops the channel operating. If Bit 2=1 then the channel will resume counting after reloading a time constant else a new control word must be written

Bit 0=1 This must be so, otherwise not a control word

When operating in timer mode the CTC is clocked by the system clock (the 4MHz crystal inside your MTX), when in counter mode the external clock is used. The external clocks for each of the channels is thus:

Channel 0 VDP interrupt line
Channel 1 4 MHz/13
Channel 2 4 MHz/13
Channel 3 Cassette ear socket

Channel 3 is used when a program is loaded from tape and the adventurous amongst you may like to examine the code at #0AAE to see just how this is done. The most interesting channel is channel 0, which uses the VDP interrupt line as it's clock. Assuming the VDP interrupt is set the VDP's interrupt line becomes active after the last line of the active display has been processed by the T.V. (or monitor if you use one). This means that the electron beam of the VDU is now displaying the border of the screen, thus if any printing to the screen or sprite movement is performed during this time there will be absolutely no flicker on the display. Thus by generating interrupts at this time and performing our sprite movement we can achieve some incredibly smooth movement (this has been used to great effect by Andrew Key in 3D Tachyon Fighter). I will now present a short program which shows the code required for this set up with comments as necessary.

Try to undestand how this works, since a good working knowledge of the CTC and interrupts will enable you to achieve some excellent effects in your own programs. For those of you that are baffled by now let me suggest some additional reading material. "Programming the Z80" and "Z80 Applications" both published by Sybex are two very good reference books which are at any Z80 programmers side. ●

SOFTWARE SOFTWARE

Program name.....[Type of program,Issue reviewed,Price,Availability]

Type of program : BS=Business AR=Arcade TW=Tactical wargame UT=Utility
ED=Educational AD=Adventure BG=Board game CG=Card game WD=Withdrawn
Price : a=#4.95 b=#5.95 c=#6.02 d=#6.95 e=#7.95 f=#8.75 g=#12.75
h=#21.25 i=#4.50 j=#16.57 k=#13
Availability : I=In stock E=Expected soon U=Unavailable at present
N.B. Please note that the prices quoted are special club discount prices

PAYROLL.....[BS,//,h,I]
SALES LEDGER.....[BS,//,g,U]
NEMO.....[AR,//,c,I]
SUPER MINEFIELD....[AR,//,c,I]
PHAID.....[AR,//,c,I]
TOADO.....[AR,//,c,I]
TAPEWORM.....[AR,//,c,I]
ASTROMILON.....[AR,//,c,I]
POT HOLE PETE.....[AR,02,c,I]
MUSIC PAD.....[WD,02,c,I]
DENNIS
& THE CHICKEN...[AR,//,c,U]
PONTOON&BLACKJACK..[CG,//,c,I]
MAXIMA.....[AR,01,c,I]
M CODER.....[UT,//,c,U]
JOHNNY REB.....[TW,//,c,I]
THE KEY TO TIME....[AD,//,c,I]
STAR COMMAND.....[AR,01,d,I]
TURBO.....[AR,02,d,I]
KNUCKLES.....[AR,//,e,I]
REVERSI.....[BG,//,e,I]
FIRST LETTERS 1....[ED,03,f,I]
MATHS 1.....[ED,//,f,I]
SNOWBALL.....[AD,04,f,I]
LORDS OF TIME.....[AD,//,f,I]
COLOSSAL ADVENTURE.[AD,02,f,I]
SPELLI-COPTER.....[ED,05,b,I]
UTILITIES 1.....[UT,//,a,I]
COMPOSER.....[UT,//,k,I]
SALTY SAM.....[AR,//,a,I]
MISSION OMEGA.....[AR,//,a,I]
BRUNWORD.....[BS,02,j,I]
GOLDMINE.....[AR,//,c,I]
LITTLE DEVILS.....[AR,04,a,I]
AGROVATOR.*.....[AR,04,b,I]
HUNCHY.....[AR,//,a,I]
MISSILE KOMMAND &
ARCAZIONS.....[AR,//,a,I]

PURCHASE LEDGER.....[BS,//,g,I]
BASIC BUSINESS.....[BS,//,b,I]
KILOPEDE.....[AR,//,c,I]
BLOBBO.....[AR,//,c,I]
MISSION ALPHATRON.....[AR,//,c,I]
OBLOIDS.....[AR,01,c,I]
CONTINENTAL RAIDERS....[AR,//,c,I]
ASTRO PAC.....[AR,//,c,I]
QOGO.....[AR,02,c,I]
SNAPPO.....[AR,//,c,I]
DENNIS
GOES BANANAS.....[AR,//,c,U]
THE ZOO GAME.....[AD,03,c,I]
GAUNTLET.....[AR,//,c,U]
COBRA.....[AR,//,c,I]
MURDER AT THE MANOR....[AD,//,c,I]
FIREHOUSE FREDDIE.....[AR,//,c,I]
DRAUGHTS.....[BG,//,d,I]
3D TACHYON FIGHTER.....[AR,04,d,I]
BACKGAMMON.....[BG,//,e,I]
CHESS.....[BG,//,f,I]
WORD & PICTURE.....[ED,04,f,I]
PHYSICS 1.....[ED,//,f,I]
ADVENTURE QUEST.....[AD,//,f,I]
DUNGEON ADVENTURE.....[AD,01,f,I]
RETURN TO EDEN.....[AD,//,f,I]
HELI-MATHS.....[ED,05,b,I]
TUMBLEDOWN TOWER.....[AD,//,i,I]
EDASM.*.....[UT,//,d,I]
DOODLEBUG DESTROYER....[AR,//,a,I]
GRAPHICS.....[UT,01,b,I]
THE MAN FROM GRANNY.*..[AD,//,a,I]
ALICE IN WONDERLAND....[AD,//,c,I]
HAWKWARS.....[AR,//,a,I]
BOUNCING BILL.....[AR,//,a,I]
BRIDGE.....[CG,//,d,I]

* runs only on the MTX512 and RS128

HARDWARE

Floppy 5 1/4" Discs	-	Verified to 96 TPI (for 10 in plastic library case)....**#17.50**
Microvitec 1431/AP/MS	-	PAL/TTL with audio.....**#225.00**
Microvitec 1431/MS	-	TTL.....**#205.00**
Cosmos 80	-	High density 80 cps matrix printer. Friction & tractor feed. Dot hi-res graphics...**#199.00**
Ensign	-	Near letter quality printer 165 cps. Bi-directional.....**#335.00**
Seikosha GP550A	-	Matrix printer.....**#205.00**
Silver Reed	-	High quality daisy wheel....**#699.00**
Centronics cable	-	For the above.....**#12.95**
Printer ribbons	-	For the DMX80.....**#8.50**
Dust cover	-	Keep your computer clean....**#3.50**

Competition ONE LINER

There was a large response to the "one line" competition in Issue 4. Here are a few examples of the entries received.

A maths tester by Paul Wood (he wrote Biorhythms in Issue 4) :-

```
1  LET A=INT(RND*10): LET B=INT(RND*10): PRINT A;" +";B;"=";:INPUT C:  IF
C<>A+B THEN PRINT "WRONG IT IS";A+B ELSE PRINT "RIGHT": GOTO 1
```

A hex to decimal converter by M Paver :-

```
1  LET  D=0:  INPUT "TYPE HEX";H$: LET L=LEN (H$)-1: FOR I=0  TO  L:  LET
A=ASC(H$(L+1-I)): LET A=A-SGN(A-60)*3.5-51.5: LET D=D+A*16^I: NEXT: PRINT
"DEC ";D: GOTO 1
```

A "shell picture" by T J Seldon :-

```
1 VS 4: CLS: PLOT 127,85: FOR A=0 TO 15: ATTR 3,1: INK A: CLS: ATTR  3,0:
FOR  X=1  TO 8: LET T=X+8*A: DRAW 5: ARC 1.5*T,2*PI:  PHI  2*PI/T:  NEXT:
NEXT: PAUSE 0
```

A reaction testing game by T J Seldon :-
(Press the space bar when the screen turns red. N.B. cheats get zero)

```
1 PAUSE RND*9999: PAPER 6: CLS: FOR X=0 TO 39: CSR X.9: PRINT " >": SOUND
1,X*9,9: IF INKEY$=" " THEN PAPER 4: PRINT "SCORE ";-(39-X)*(X>0): GOTO 1
ELSE NEXT
```

At the time of going to press entries were still being received so the winner will be announced in the next issue.

EDUCATION

*I am receiving very good reports from members who have taken advantage of the COMPUTER TRAINING COLLEGE'S "TEACH YOURSELF MACHINE CODE" Flexi-course book. For the benefit of new members: You can purchase the college's assembly language course at a special price of £7.95p. If you get stuck at any place in the book, the college will try to help you by enlarging on the area that is causing you the difficulty. AN OFFER NOT TO BE MISSED. Send cash with order to: MR R. BROOME F.I.C.O.M.R.A.M .C.A THE COMPUTER TRAINING COLLEGE, Norvic House, 1-7, Hilton Street, Manchester M4 1LP **Quote your Genpat number.*

FOR THE BENEFIT OF NEW MEMBERS

GRAPHICS is a utility package that allows you to design Sprites, User definable, and the Ascii character set. You can then save your new designs and incorporate them within your programs without the need for Genpat statements - you treat your designs as though they were in ROM. It is a really efficient and easy way to design you characters.

EDASM: This is a really first class MACRO-ASSEMBLER. It supports all the Zilog commands including EQU & ORG which means you can load your programs into a specific memory location without the need to write re-location code. The Macro can be changed to write in 6509 6502,6800, or your own language if you wish to do so. ** 512 only

AGROVATOR This is definitely going to be one of the best sellers of 1985. It is a completely orginal maze game which really does become addictive. 30 random mazes an a host of things to collect....you never get bored with this one ! ** 512 only

LITTLE DEVILS: Really good game - lots of fun and fast action. One of the favourites at Genpat.

BOUNCING BILL A very simple game and will have the rest of the family in stitches as they scoff at your downfall. Ideal fun for all the family.

HAWKWARS I had doubts about accepting ths game onto the lable, but the response from purchasers has made me glad I took the chance. Very fast and A b------ to play!

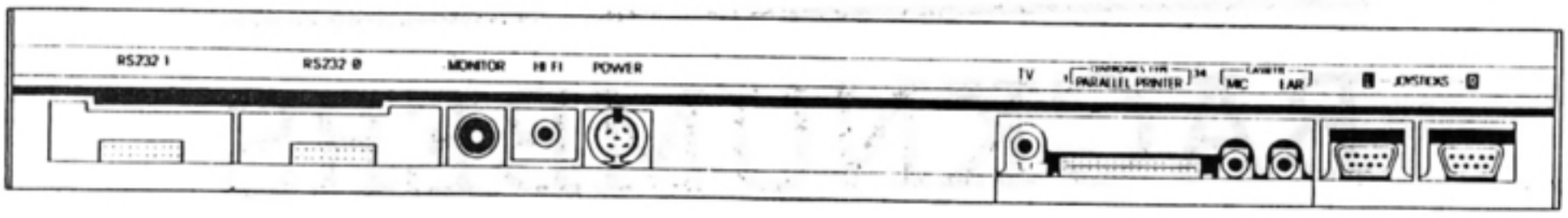
SALTY SAM You wouldn't call this game a masterpiece, but it is a game that the young or old can play. The programmer has spent a lot of time on keeping the game simple ,and its nice to have a game that doesn't rely on you blasting your way through hundreds of aliens. It's a ladders and monsters game .... and a favourite with the younger members of the family.

DOODLEBUG DESTROYER What can I say about this game ? I can't beat it ! It is the first import from another country, and is written by Craig Dunn from Scandavisions. The action is furious, and is one of those games you can't walk away from.

NEW RELEASES

HUNCHY If you don't avoid the arrows you end up in the moat with the swans ! If you manage to ring the bell ..... well that's another story.
Syntax Software #4.95p

WATCH OUT FOR "SON OF PETE".....Comming shortly .... the first release from MEGGA STAR.



END STATEMENT

This has been the month I would most like to forget. I must apologise, again, for being late with this month's magazine, but I think we are now out of the woods, and February's edition should reach you on time.

January has been a good month for membership: we have recruited 50% of our total membership during this month and I am pleased to see that a lot more owners are interested in sharing their knowledge.

This is truly an international club with members in Poland, Belgium, Ireland Holland, Denmark, France, Germany, Sweden, Finland, Australia, and the U.S.A. Incidentally, would any members be interested in having a long week-end in Holland sometime in April ? I am hoping to visit Nick Passmore, and our friends in the Dutch User Club. If enough members are interested I can arrange for a coach to take us there Friday, and bring us back on the Sunday evening. If you would like to join me, drop me a line.

From this week I will not be available on Wednesdays, and will close at 6 - 30pm on Saturdays. The reason for this is simple: I haven't been home at weekends for the past 3 weeks, and my good lady is starting to look at other men ! I would also be most grateful if you could keep your phone calls as short as possible as many people cannot get through. Most of the time this is due to the vast number of calls I receive, for instance, when I returned from the printers this evening the answer-phone had clocked up 237 calls ! However, having said that, I have now finished my book, so please feel free to contact me if you have a problem. But please remember, I sometimes have to leave the premises, and I can't guarantee that I will be here all the time.

A lot of members are puzzled at what I do for a living. Well, my wife has a Take-away, and I work above in the back room. Genpat is not my role in life, it does not pay me a living. I earn my wages by writing book,magazine articles, and contract programming for other companies. I started the User Group after consultation with Memotech, but at that time neither the Company, nor myself realised how fast the club would grow. However, grown it has, and for all it's faults, it is still one of the best clubs in the U.K.

If you would like to see any subject, however trivial you make think it is, covered in the magazine please drop me a line. I would appreciate response from school teachers, a I would like to keep the education slot running, and allow those occupied in education to air their views and publish their programs. Another area which lacks contributions is the hardware section. Surely some of you most have discovered a brilliant way of inerfacing the MTX. What about the ladies ? We have yet to receive a contribution from a lady member.

Before I close I would like to answer a question that has been raised over who has the right to enter competitions etc. It doesn't matter who is the registered member. Computing should be a family affair, and all the family are certainly entitled to join in anything to do with Genpat.... this is what we are about.