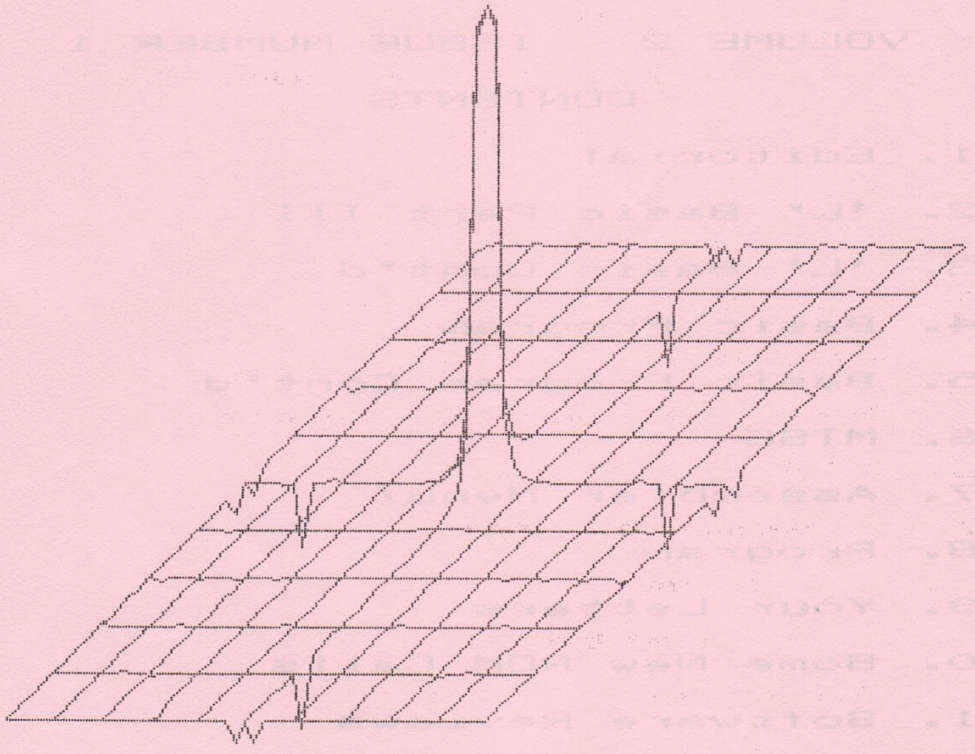
VOL. 2 ISSUE 1 SEPTEMBER 1985

# THE MEMOTECH OWNERS CLUB MAGAZINE



FEATURES:-

DBASE PARTIII

NEW ROM CALLS!!

BASIC PROG

PUBLISHED BY MEMOTECH OWNERS CLUB
23 DENMEAD ROAD
HAREFIELD SOUTHAMPTON

CIRCA ...200 ish

M.O.C.

VOLUME 2 ISSUE NUMBER 1

CONTENTS



o o o O O O o o o

# E D I T O R I A L

Phil Eyres
23 Denmead Road
Harefield
Southampton
SO2 5GS

Richard Adams
18 Nightingale Rd
Pilands Estate
Bursledon
Southampton

Well, as Summer draws to a close?, there's an Autumn 'nip' in the evening air?, the nights are drawing in and yet another computing season starts. Tee..Hee!! ..I'm only 'gee'ing you up about our fantastic summer, I can do this as I'm off to Tenerife for two weeks from the 8th. Oh! because of this there will be a slight delay in answering some of the mail.

On a 'not so good' note, our membership renewals have been a bit disappointing, out of 130 renewals we've had bearly 60 so far, we decided to send out notes to those who have not renewed to prompt them into action, also a new drive will be launched to 'recruit' new members, although from experience this has proved to be a long process.

By the time you read this, we should have been on a visit to Memotech, as they have invited us to one of their 'dealer days'. We are hoping that this will be fruitful in that it will provide us with information as to their position in the British market. (We heard recently that they were trying hard to get a major chain store to accept the Memotech in time for Christmas).

This is a small note to everyone that has bought a Speech Synthesiser kit from us, one member has experienced a problem in that all he could get out of his SP was a continuous 'buzz' regardless of volume control, after investigation it was found that the kit required a voltage de-coupling capacitor extra to the components supplied, should anyone else having one of the early kits have this problem please get in contact with us and we will send on the necessary component. N.B. It's only a little electrolytic capacitor worth about 9p.

Thanks to everyone that has used our Hotline on Monday evenings between 6 & 7pm, remember we always look forward to hearing from you, the number to phone is Bursledon (042121) 5489. Ask for Rich!

If anyone would like back issues they are available for all past magazines for the small remittance of 80p.



Our first year is now complete and we think that it has been a success, although the club still owes £857 we are pleased that we have been able to make a significant repayment on the loan. We have been able to keep the cost of the magazine to the member the same as last year, this was mainly due to the change of print format, but we will have to keep the situation under review because of pending increases in postage and paper.

M.O.C. Accounts Ending 31-8-85

| | | |
|---|---|---|
| Total Money from Membership etc | | 2363.34 |
| Breakdown Of Income | | |
| Magazine Production Costs | | 1071.64 |
| Holding Account | | 876.00 |
| Current Account Money | | 415.70 |
| | Total | 2363.34 |
| Money Borrowed for Equipement | | 1733.00 |
| Money Paid Off Interest Free Loan | | 876.00 |
| Outstanding Loan Balance | | 857.00 |
| Money Retained For Running Cost | | 415.70 |
| Interest From Holding Account | | 5.68 |
| | Total Retained | 421.38 |

******

For Sale

Dave West has the following equipement for sale :-

MTX 500
(All original packing, including Manual,Leads,Tapes etc)
Printer Cable, Obloids, Mission Alphatron, MTX Data File, Brunword, 4 Tapes Home Recorded with various routines.
2 Books
Lots Of Printed Info From Memotech
Back Issues Of MOC (The BRILLIANT MOC Mag!!!)

Asking Price Only £130.00 o.n.o

Please contact Dave West on 0296 33020

******

# L BASIC PART III

This month I have tackled the problem of Saving data to tape. Originally I thought that this was going to be a simple task, it has however turned out to be quite a challenge. I have made two routines to cover this, the first method is simple and barely covers the basics of data saving, using the only command available to the standard MTX. The second method is much more 'inline' with data manipulation techniques using cassette based systems, it does however require the use of data save and load utility on the MTX utilities tape.

The first version, version 1.0 will have to suffice for those who do not have the MTX utility, it works but has a definite limitation in that the program cannot be modified without the loss of the data in FILE$. So until the other routines have been written the program cannot be used. Basically the routine is dependant on only one line, line 2110, this will save, not only the data in the arrays DIM'ed at the beginning of the program, but the whole program. The progam is infact saved whilst it is running, so when it is reloaded the program will continue at line 2120, with all the data intact!. This is ok, but it does take a while to save, especially for those with MTX512's who have made the FILE$ array larger to fill up their memory. Because the program 'auto-runs' on reloading from cassette it is not necessary to have a data load routine for this method.

The second method, overleaf is much more inline with data save and load methods, although it is still resticted due to the use of a cassette as a magnetic data holding media. The method used for version 2 is infact technically called sequencial filing, that is, all the files are stored one after the other on cassette and are only retrievable in that precise order. So, it is necessary to make sure the order of data saving is well defined, it is also necessary to save other data as well as FILE$ to ensure the correct state of the program is returned to on data load. That is, the number of files stored 'REC(1)' must be saved, also the file FNAME$ and the variable FIELD must be saved. For this to happen correctly, a couple of slight adjustments have to be made to the Open file routine, they are:-

1. All references to REC must be changed to REC(1).
2. All references to FIELD must be changed to FIELD(1).
   !!! This is not necessary for version 1 !!!

Also, whilst on the point of modifying the Open file routine, two other little ideas for improvement have occurred to me, they are:-

1. A check should be made to ensure a maximum of 100 FILE's can be entered. i.e.

```
3325 IF REC(1)=101 THEN CLS: CSR 5,3: PRINT "FILE NOW
     FULL: GOTO 3410
```

2. A couple of lines should be added at the beginning of the routine to clear the arrays FILE$ and FNAME$. I'll leave this to you as it is not too difficult!!

If anyone would like a listing of the program so far, please send a SAE stating which version you would like. Note, for version 2 you will require the MTX utility tape.

The state of the DIM'ed variables and arrays has been printed at the beginning of each listing for clarity.

You should now have a DBASE program with which you can input and store data. Next month I'll try and shine some light on displaying the data on your T.V./Monitor and printer.

```
5 REM-----------------------------------------
10 REM DBASE By Phil Eyres V1.0
20 REM-----------------------------------------
30 DIM FILE$(100,6,30)
40 DIM FNAME$(6,20),REC(1)
50 LET REC(1)=0
1997 REM-----------------------------------------
1998 REM SAVE DATA TO TAPE
1999 REM-----------------------------------------
2000 CLS
2010 CSR 7,3: PRINT "!!!DATA SAVE ROUTINE!!!"
2020 CSR 7,4: PRINT "   ^^^^^^^^^^^^^^^^^^^^"
2025 IF REC(1)<2 THEN CSR 13,7: PRINT "FILE EMPTY": PAUSE 3000: GOTO 2120
2027 CSR 7,15: PRINT "NAME OF FILE TO BE SAVED": CSR 12,17: INPUT FILNAME$
2030 CSR 4,10: PRINT "START CASSETTE AND PRESS RETURN!";
2040 INPUT W$
2050 IF W$="" THEN GOTO 2100
2060 CSR 9,7: PRINT "!!! SAVE ABORTED !!!"
2070 PAUSE 3000
2080 RETURN
2100 CSR 12,7: PRINT "!!! SAVING !!!"
2110 SAVE FILNAME$
2120 RETURN
```

```
5 REM----------------------------------------
10 REM DBASE BY PHIL EYRES V2.0
20 REM----------------------------------------
30 DIM FILE$(100,6,30)
40 DIM FNAME$(6,20),REC(1)
50 LET REC(1)=0
60 DIM FIELD(1),DUMMY$(1,30)
70 DIM Z$(30)
997 REM----------------------------------------
998 REM INPUT DATA FROM CASSETTE
999 REM----------------------------------------
1000 CLS
1010 CSR 7,3: PRINT "!!!DATA LOAD ROUTINE!!!"
1020 CSR 7,4: PRINT "~~~~~~~~~~~~~~~~~~~~~~~~~~"
1030 CSR 4,10: PRINT "START CASSETTE AND PRESS RETURN!";
1040 INPUT W$
1050 IF W$="" THEN GOTO 1100
1060 CSR 9,7: PRINT "!!! LOAD ABORTED !!!"
1070 PAUSE 3000
1080 RETURN
1100 DISC LOAD REC(1)
1110 DISC LOAD FIELD(1)
1120 FOR I=1 TO FIELD(1)
1130 DISC LOAD FNAME$(I)
1140 NEXT
1150 FOR I=1 TO REC(1)-1
1160 FOR P=1 TO FIELD(1)
1170 DISC LOAD Z$
1173 LET FILE$(I,P)=Z$
1180 NEXT P: NEXT I
1190 RETURN
1997 REM----------------------------------------
1998 REM SAVE DATA TO TAPE
1999 REM----------------------------------------
2000 CLS
2010 CSR 7,3: PRINT "!!!DATA SAVE ROUTINE!!!"
2020 CSR 7,4: PRINT "~~~~~~~~~~~~~~~~~~~~~~~~~~"
2030 IF REC(1)<2 THEN CSR 13,7: PRINT "FILE EMPTY": PAUSE 3000: RETURN
2040 CSR 7,15: PRINT REC(1)-1;" FILES BEING SAVED"
2050 CSR 4,10: PRINT "START CASSETTE AND PRESS RETURN!";
2060 INPUT W$
2070 IF W$="" THEN GOTO 2200
2080 CSR 9,7: PRINT "!!! SAVE ABORTED!!!"
2090 PAUSE 3000
2100 RETURN
2110 CSR 12,7: PRINT "!!! SAVING !!!"
2200 DISC SAVE REC(1)
2210 DISC SAVE FIELD(1)
2220 FOR I=1 TO FIELD(1)
2230 LET Z$=FNAME$(I): DISC SAVE FNAME$(I)
2240 NEXT
2250 FOR I=1 TO REC(1)-1
2260 FOR P=1 TO FIELD(1)
2265 PRINT FILE$(I,P)
2270 LET Z$=FILE$(I,P): DISC SAVE Z$
2280 NEXT P: NEXT I
2300 RETURN
```

```
1 REM ------------------------------------
2 REM Ball Fall By Micheal Green. 15.8.85
3 REM ------------------------------------
10 GOTO 2000
50 FOR F=1 TO 4: MVSPR 1,1,DI: NEXT : RETURN
100 LET B$=INKEY$: IF B$="" THEN GOTO 160
110 IF B$=CHR110 IF B$=CHR$(26) THEN GOTO 500
120 IF B$=CHR$(8) AND X>WIDTH1 THEN LET X=X-1: LET DI=4: GOSUB 50
130 IF B$=CHR$(25) AND X<(WIDTH2+WIDTH1-1) THEN LET X=X+1: LET DI=0: GOSUB 50
140 IF B$=CHR$(10) AND Y<10 THEN LET Y=Y+1: LET DI=2: GOSUB 50
150 IF B$=CHR$(11) AND Y>1 THEN LET Y=Y-1: LET DI=6: GOSUB 50
160 LET TI=TI-1: CSR 17,0: PRINT TI;" "
170 IF TI=0 THEN GOTO 1000 ELSE GOTO 100
500 IF A(Y,X)=0 THEN LET A(Y,X)=1: LET B=131: LET C=132: LET D=131: LET E=132: GOTO 560
510 IF A(Y,X)=1 THEN LET A(Y,X)=0: LET B=129: LET C=129: LET D=130: LET E=130: GOTO 560
520 IF A(Y,X)=2 THEN LET A(Y,X)=3: LET B=135: LET C=132: LET D=130: LET E=136: GOTO 560
530 IF A(Y,X)=3 THEN LET A(Y,X)=4: LET B=131: LET C=137: LET D=138: LET E=130: GOTO 560
540 IF A(Y,X)=4 THEN LET A(Y,X)=5: LET B=139: LET C=129: LET D=131: LET E=140: GOTO 560
550 IF A(Y,X)=5 THEN LET A(Y,X)=2: LET B=129: LET C=133: LET D=134: LET E=132
555 IF A(Y,X)=6 THEN LET B=32: LET C=32: LET D=32: LET E=32
560 CSR 2*X,2*Y: PRINT CHR$(B);CHR$(C)
570 CSR 2*X,2*Y+1: PRINT CHR$(D);CHR$(E)
580 IF B$<>INKEY$ THEN GOTO 100
590 LET TI=TI-1: CSR 17,0: PRINT TI;" "
600 IF TI=0 THEN GOTO 1000 ELSE GOTO 580
1000 LET X=1: LET Y=BAL: LET DI=2: LET B=2: LET P=2
1010 REM +++ BALL POSITIONS,DIRECTIONS,PATTERN +++
1020 IF Y<WIDTH1 OR Y>WIDTH2+WIDTH1-1 OR X<1 THEN GOTO 1800
1030 IF X>10 AND Y=BUK THEN LET BUK=1: GOTO 4000
1040 IF X>10 AND Y<>BUK THEN GOTO 1800
1045 MVSPR 1,2,DI
1050 LET F=A(X,Y)
1060 IF DI=0 AND F<>0 AND F<>2 AND F<>3 THEN GOTO 1800
1070 IF DI=2 AND F<>1 AND F<>3 AND F<>4 THEN GOTO 1800
1080 IF DI=4 AND F<>0 AND F<>4 AND F<>5 THEN GOTO 1800
1090 IF DI=6 AND F<>1 AND F<>2 AND F<>5 THEN GOTO 1800
1100 IF F=6 THEN GOTO 1800
1110 MVSPR 1,2,B
1120 IF F=2 AND DI=6 THEN LET B=4: LET Y=Y-1
1130 IF F=2 AND DI=0 THEN LET B=2: LET X=X+1
1140 IF F=3 AND DI=0 THEN LET B=6: LET X=X-1
1150 IF F=3 AND DI=2 THEN LET B=4: LET Y=Y-1
1160 IF F=4 AND DI=4 THEN LET B=6: LET X=X-1
1170 IF F=4 AND DI=2 THEN LET B=0: LET Y=Y+1
1180 IF F=5 AND DI=6 THEN LET B=0: LET Y=Y+1
1190 IF F=5 AND DI=4 THEN LET B=2: LET X=X+1
1200 IF F=1 AND DI=6 THEN LET B=6: LET X=X-1
1210 IF F=1 AND DI=2 THEN LET B=2: LET X=X+1
1220 IF F=0 AND DI=4 THEN LET B=4: LET Y=Y-1
1230 IF F=0 AND DI=0 THEN LET B=0: LET Y=Y+1
1235 MVSPR 1,2,B
1240 IF P=2 THEN LET P=3 ELSE LET P=2
1250 MVSPR 2,2,P
1320 LET DI=B
1330 MVSPR 1,2,DI
1340 LET PIP=PIP+1
1350 GOTO 1020
1800 FOR F=(11-X)*16 TO 0 STEP -4
1810 MVSPR 1,2,2
1820 IF P=2 THEN LET P=3 ELSE LET P=2
1830 MVSPR 2,2,P
1835 PAUSE 30
1840 NEXT F
1850 LET BUK=0: REM +++ BUCKET MISSED +++
1860 GOTO 4000
2000 DIM A(10,14): REM +++ PIPES ARRAY +++
2020 VS 4: CLS : COLOUR 0,11: COLOUR 4,2: INK 1
2030 LET HISC=0: REM +++ HISCORE +++
2040 CTLSPR 2,2: REM +++ NO. OF SPRS. +++
2060 CTLSPR 1,4: REM +++ DISTANCE +++
2065 CTLSPR 6,2: REM +++ MAGNITUDE +++
2070 GENPAT 4,1,0,7,60,38,17,56,36,19
2080 GENPAT 5,1,8,28,32,31,0,0,0,0
2090 GENPAT 6,1,0,12,146,82,52,132,72,7
2100 GENPAT 7,1,131,3,7,143,127,63,31,15
2110 SPRITE 1,1,24,168,0,0,1: REM +++ HAND +++
2120 GENPAT 1,129,0,255,0,255,0,0,0,0
2130 GENPAT 1,130,0,0,0,0,255,0,255,0
```

```
2140 GENPAT 1,131,80,80,80,80,80,80,80,80
2150 GENPAT 1,132,10,10,10,10,10,10,10,10
2160 GENPAT 1,133,0,192,48,136,100,20,18,10
2170 GENPAT 1,134,0,0,0,0,192,32,144,80
2180 GENPAT 1,135,80,144,32,192,0,0,0,0
2190 GENPAT 1,136,10,18,20,100,136,48,192,0
2200 GENPAT 1,137,10,9,4,3,0,0,0,0
2210 GENPAT 1,138,80,72,40,38,17,12,3,0
2220 GENPAT 1,139,0,3,12,17,38,40,72,80
2230 GENPAT 1,140,0,0,0,0,3,4,9,10
2240 GENPAT 4,2,0,0,0,0,3,7,15,15
2250 GENPAT 5,2,15,15,7,3,0,0,0,0
2260 GENPAT 6,2,0,0,0,0,192,224,240,112
2270 GENPAT 7,2,176,208,224,192,0,0,0,0
2280 SPRITE 2,2,24,176,0,0,15: REM BALL
2290 GENPAT 1,141,3,12,48,64,112,76,35,32
2300 GENPAT 1,142,192,48,12,2,14,50,196,68
2310 GENPAT 1,143,32,16,25,22,8,8,12,3
2320 GENPAT 1,144,68,136,8,8,16,16,48,192
2330 GENPAT 4,3,0,0,0,0,3,7,11,13
2340 GENPAT 5,3,14,15,7,3,0,0,0,0
2350 GENPAT 6,3,0,0,0,0,192,224,240,240
2360 GENPAT 7,3,240,240,224,192,0,0,0,0
2600 LET SC=0: LET ROU=0: REM +++ SCORE & ROUND +++
2700 LET ROU=ROU+1
2705 IF ROU<3 THEN LET WIDTH1=4: LET WIDTH2=8 ELSE LET WIDTH1=1: LET WIDTH2=14
2710 LET BAL=INT(RND*WIDTH2)+WIDTH1: REM BALL POSITION
2720 ADJSPR 2,2,16*BAL+8: ADJSPR 3,2,176
2725 REM +++ ADJUST TI TO SUIT SKILL +++
2730 IF ROU=2 OR ROU>4 THEN LET TI=900 ELSE LET TI=1000
2740 LET PIP=0: REM +++ PIPES USED +++
2750 CLS : CSR 11,0: PRINT "TIME =";TI
2760 ADJSPR 2,1,120: ADJSPR 3,1,168
2770 IF ROU>3 THEN LET SPACE=1 ELSE LET SPACE=0
2780 LET BUK=INT(RND*WIDTH2)+WIDTH1: REM +++ BUCKET POSITION +++
2800 GOSUB 3000
2810 LET X=7: LET Y=1: REM +++ HAND COORDINATES +++
2820 GOTO 100
3000 INK 4
3010 FOR X=1 TO 10
3020 FOR Y=WIDTH1 TO WIDTH2+WIDTH1-1
3030 IF Y=BUK OR Y=BAL THEN LET F=INT(RND*6) ELSE LET F=INT(RND*(6+SPACE))
3040 IF F=0 THEN LET B=129: LET C=129: LET D=130: LET E=130
3050 IF F=1 THEN LET B=131: LET C=132: LET D=131: LET E=132
3060 IF F=2 THEN LET B=129: LET C=133: LET D=134: LET E=132
3070 IF F=3 THEN LET B=135: LET C=132: LET D=130: LET E=136
3080 IF F=4 THEN LET B=131: LET C=137: LET D=138: LET E=130
3090 IF F=5 THEN LET B=139: LET C=129: LET D=131: LET E=140
3100 IF F=6 THEN LET B=32: LET C=32: LET D=32: LET E=32
3200 CSR 2*Y,2*X: PRINT CHR$(B);CHR$(C)
3210 CSR 2*Y,2*X+1: PRINT CHR$(D);CHR$(E)
3230 LET A(X,Y)=F
3240 NEXT Y: NEXT X
3260 INK 1
3270 CSR 2*BUK,22: PRINT CHR$(141);CHR$(142)
3280 CSR 2*BUK,23: PRINT CHR$(143);CHR$(144);
3290 INK 4
3900 RETURN
4000 INK 1: ADJSPR 2,1,0: ADJSPR 2,2,0
4010 CLS : CSR 11,1: PRINT "BALL FALL                                    ----------"
4020 CSR 11,4: PRINT "Scorecard                                  ----------"
4030 CSR 3,7: IF BUK=1 THEN PRINT "Ball in bucket = 3000 pts." ELSE PRINT "Ball missed buck
"
4040 CSR 2,9: PRINT PIP;" pipes used  =";PIP*100;" pts."
4045 LET F=(BUK*3000)+(PIP*100): REM ROUND SCORE
4050 CSR 3,11: PRINT "Round";ROU;" score  =";F;" pts."
4060 CSR 3,13: PRINT "Previous rounds=";SC;" pts."
4070 LET SC=SC+F
4080 CSR 3,15: PRINT "New Score      =";SC;" pts."
4090 CSR 3,19: PRINT "Today's Hiscore=";HISC;" pts."
4100 IF BUK=0 THEN GOTO 4200
4110 CSR 3,22: PRINT "Press any key for next round"
4120 IF INKEY$="" THEN GOTO 4120 ELSE GOTO 2700
4200 CSR 11,17: PRINT "GAME OVER"
4210 IF SC>HISC THEN LET HISC=SC
4220 CSR 3,22: PRINT "Press any key for a new game"
4230 IF INKEY$="" THEN GOTO 4230 ELSE GOTO 2600
9999 SAVE "Ball Fall": RUN
```

## 'THE LATE LATE PAGE?'

This page was set aside for an advert from the Micro Technology Support Centre, when in the final (Ed-> Late Late!) stages of printing they decided to withdraw their advert, when prompted for a reason, they said that they were no longer going to support the Memotech range because of the poor sales. This, all and all, has left us somewhat in the 'mire', we have no software advert and no time to get the latest titles and prices to print our own. So for this month all we can do is offer what we have in stock, next month we will try and make amends by finding some new titles.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

*  S O F T W A R E  *
S U M M E R
*  C L E A R  O U T

*  THESE PRICES ARE FOR ONE MONTH ONLY AND APPLY ONLY TO EXISTING STOCKS.  *

| STOCK QTY | ITEM | MOC PRICE |
|---|---|---|
| 3 | POT HOLE PETE | £5.50 |
| 2 | FELIX FACTORY | £5.50 |
| 2 | OBLOIDS | £5.50 |
| 1 | MUSIC PAD | £5.00 |
| 3 | GOLDMINE | £5.50 |
| 3 | MAXIMA ONLY | £4.50 |
| 2 | DUST COVERS | £3.50 |
| 3 | DUNGEON ADVENT | £7.50 |

*  ALL MEGASTAR GAMES £6.00  *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

## SPEECH SYNTHESISER

Over the past month I have built and tested my Speech Synthesiser, I must say it is quite imppressive, the speech does actually sound like English!, I have utilised the internal port connector used for the LED kit so anyone with the connecting lead will only have to build and plug in. The connector is still available from us and as far as I know we are the only people supplying the lead ready built.

Why buy the speech synthesiser as a kit?, well apart from the saving in cost, building electronics projects is really a great hobby, this kit is fairly easy to build ( a bit more of a challenge than the previous kit), and it will no doubt teach many something about electronic principles. It really is better value for money to actually build it yourself, all you will need is a soldering iron, smaller pair of wire cutters and some spare time.

I have a simple program that uses DATA strings to hold the data for the words and I am currently working on a better program that would allow DSI (Direct Screen Input) to make and modify data. In the future I intend to build a small board that will allow data ROM's to be connected to the Synthesiser to save on time needed to build up speech data.(All I have to find is a supplier!!)

Anyway, I still need about a week to finalise the sheets needed for the project and to get hold of the necessary components, so if you would like a kit then send me your order at the beginning of July.

Interface price list

A full set of components and instructions for the LED kit --)£6.95

A full set of components and instructions for the Speech Synthasiser kit --)£18.00

Connecting cable for the internal port (needed for both projects) --)£4.50

All prices are fully inclusive. Please allow 14 days for delivery and make checks payable to MOC.

# WHAT'S ON THE MENU

By
C.J.Staniuszko

As a member of MOC, I feel that it's about time I stopped being just a receiver of information and started to contribute towards the contents of the magazine. Attempting to learn assembler, I decided to write a routine which would enable me to practice some techniques and become more familiar with the instructions. One of my BASIC programs requires a number of menus, so this seemed to be a reasonable place to begin, after all, throwing information onto the screen is fairly easy once you learn a couple of rules about the RST 10 instruction. After a period of typing lots of 'CP' instructions ;- one for every option; it struck me that this was an incredibly inefficient way of testing for results.

After a cup or two of coffee, it struck me that it should be possible to produce a 'look up table' which I could then use to redirect the PC to different parts of a routine upon the results of a keypress. What didn't occur to me at the time of writing the routine was that it could also be used to simulate a sort of assembler 'ON GOTO' instruction, some minor alterations would no doubt produce an 'ON GOSUB' instruction.

```
; MENU/ON-GOTO ROUTINE By C.J.STANIUSZKO  7.7.85
; 'MENU' HOLDS THE CURRENT MENU ie.MAIN=0 MAINT=1
; 'TABLE1'  HOLDS  THE ASCII OF KEYS WHICH  MAY  BE
  PRESSED WHILE AT A MENU.
; 'TABLE2' HOLDS THE LOCATIONS TO BE JUMPED TO UPON
  THE TEST AGAINST 'TABLE1'.
; TABLES  1&2  MAY  LATER BE ALTERED  TO  HOLD  THE
  RESULTS  OF  CALCULATIONS AS LONG AS  THESE  HAVE
  THEIR OWN GROUP 'MENU' No.
; 'MOFFST'  HOLDS THE ACTUAL LOCATION OF THE  FIRST
  KEY TEST BYTE IN 'TABLE1' RE. THE CURRENT MENU.
; 'GOFFST' AS 'MOFFST' BUT FOR 'TABLE2'.

MT1:    NOP          * * * * * * * * * * * * * * * *
MT2:    NOP          THIS  PROGRAM SHOULD BE ENTERED
MT3:    NOP          USING   YOUR  ASSEMBLER.   (SEE
MT4:    NOP          MANUAL    FOR    INSTRUCTIONS)
MT5:    JP INKEY     MEMORY   LOCATIONS  HAVE   BEEN
MT6:    JP MAIN      OMITTED  FROM  THE LEFT  COLUMN
M1:     NOP          AS  THESE DIFFER BETWEEN MTX500
M2:     NOP          AND  MTX512/RS128.
M3:     JP INKEY     * * * * * * * * * * * * * * * *
M4:     JP MAINT
QUIT:   JP END
MENU:   DB 1
MOFFST: DW 1      ;HOLDS ACTUAL POS. OF FIRST APPL. KEY
GOFFST: DW 1      ;  "     "     "    " TABLE 2
TABLE1: DB £80,£81,£82,£83,£8F,0,£80,£81,£82,£83,£84
        DB £8F,0
TABLE2: DW M1,M2,M3,M4,QUIT,0,MT1,MT2,MT3,MT4,MT5,MT6
END:    NOP       ;USE TO CLEAN UP
        POP IY
        POP IX
        POP HL
        POP DE
        POP BC
        POP AF
        EI
        RET
START:  NOP
        PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL
        PUSH IX
        PUSH IY
        JP MAIN
MAINT:  NOP
        RST 10
        DB £6D,£B0,"MAINTENANCE MENU"
        DB £A3,£0A,£0A,£0D,£AD,"F1 START FILE"
        DB £A2,£A0,£0D,£AE,"F2 ADD TO FILE"
        DB £A2,£0A,£0D,£AE,"F3 DELETE ITEM"
        DB £A2,£0A,£0D,£AE,"F4 INSERT ITEM"
        DB £A2,£0A,£0D,£A7,"F5 SORT"
        DB £A2,£0A,£0D,£97,"SF8 RETURN TO MAIN MENU"
        LD HL,MENU
        LD (HL),1      ;TELL THE PROGRAM WHICH MENU
        JP OFFST       ;SORT OUT DISPLACEMENTS
MAIN:   RST 10
        DB £6D,£A9,"MAIN MENU"
        DB £A3,£0D,£0A,£0A,£B2,"F1 EXTENDED REPORT"
        DB £A2,£0D,£0A,£B4,"F2 COMPRESSED REPORT"
        DB £A2,£0D,£0A,£B3,"F3 GRAPHICAL REPORT"
        DB £A3,£0D,£0A,£0A,£B3,"F4 FILE MAINTENANCE"
        DB £A4,£0D,£0A,£0A,£0A,£88,"SF8 QUIT"
        LD HL,MENU     ;TELL THE PROGRAM WHICH MENU
        LD (HL),0      ;IS IN USE MAIN MENU=0
OFFST:  LD HL,MENU     ;OFFST - INKEY FINDS THE FIRST
        LD B,(HL)      ;APPL. DATA FOR EACH MENU
        LD HL,TABLE1
        LD A,B
```

```
        OR A            ;IS IT =0?
        JP Z,SAVE       ;IF MAIN MENU IS BEING USED
LOOP1:  LD A,(HL)       ;READ NEXT ITEM OF DATA
        OR A            ;IS IT A 0?
        INC HL
        JR NZ,LOOP1
        DEC B           ;DEC THE COUNT FROM MENU
        LD A,B          ;IS B=0?
        OR A            ;IF YES THEN THIS IS CORRECT DATA
        JR NZ,LOOP1
SAVE:   EX DE,HL
        LD HL,MOFFST    ;PUT THE ACTUAL POSITION OF
        LD (HL),E       ;FIRST ITEM IN MOFFST
        INC HL
        LD (HL),D
        EX DE,HL        ;PUT ACTUAL POSITION IN HL
        LD DE,TABLE1
        SBC HL,DE       ;FIND OUT HOW FAR ALONG TABLE1
        EX DE,HL        ;IS THE FIRST ITEM. ANS. IN DE
        LD HL,TABLE2
        ADC HL,DE       ;ADD TWICE TO BEGINNING OF
        ADC HL,DE       ;TABLE2 (16 BIT)
        EX DE,HL        ;DE HOLDS FIRST OFFSET VALUE
        LD HL,GOFFST
        LD (HL),E
        INC HL          ;GOFFST HOLDS ACTUAL ADDRESS OF
        LD (HL),D       ;FIRST 'JUMP TO' VALUE
INKEY:  CALL &79        ;CALL KEYBOARD SCAN
        JR Z,INKEY      ;KEY PRESSED?
        CALL COMP
        JR C,INKEY      ;CATCH THE INCORRECT KEYPRESS
        JP GO           ;REDIRECT THE PC
COMP:   LD HL,MOFFST    ;PLACE THE POS. OF THE FIRST
        LD E,(HL)       ;DATA ITEM IN HL, USING DE
        INC HL
        LD D,(HL)
        EX DE,HL
        LD BC,0         ;INTIALISE COUNT
LOOP2:  LD E,A          ;PRESERVE A
        LD A,(HL)       ;GET BYTE FROM TABLE1
        OR A            ;IF A=0 THEN SET ZERO FLAG
        JR NZ,LOOP3     ;A<>0 THEN GET NEXT BYTE
        SCF             ;SET CARRY FLAG
        RET
LOOP3:  LD A,E          ;RESTORE KEY VALUE TO A
        CP (HL)         ;TABLE VAL=KEY PRESSED?
        RET Z           ;RETURN IF SO
        INC BC          ;INCREMENT COUNT
        INC HL          ;MOVE COUNTER ON
        JR LOOP2        ;LOOK AT NEXT VALUE
GO:     NOP             ;REDIRECT PC
        LD HL,(GOFFST)  ;POINT HL AT FIRST REDIRECTION
        RLC C           ;DOUBLE INDEX
        RLC B
        ADD HL,BC       ;OFFSET INTO TABLE
        LD B,(HL)       ;GET ADDRESS LSB ON B
        INC HL
        LD H,(HL)       ;GET ADDRESS MSB IN H
        LD L,B          ;PUT LSB IN L
        JP (HL)         ;DO ON-GOTO ADDRESS
        RET
```

## DESCRIPTION OF THE ROUTINE

It seemed pertinent to save all of the registers before entering the routine proper, and restore them upon leaving. (The EI is there just in case I had a brainstorm and stuck a DI somewhere-really infuriating). The first thing the routine proper does is throw the main menu onto the screen. It then places the value 0 in the variable 'MENU'. This acts as a counter for the next section which calculates the starting positions of each set of possible results for comparison (within TABLE1) and the equivelent locations to which jumps are required. After this, the routine goes to 'INKEY' and waits for a key to be pressed.

Once a key has been pressed, the routine calls 'COMP' which takes the ASCII value of the key and compares it against the table of possible results. If the key has not been allowed for, the carry flag is set, and the routine returns to the keyboard scan. If the key is one of those allowed for, the routine jumps to 'GO', which reads the equivelent value in 'TABLE2' to the keypress and redirects the program counter.

I have chosen to send the routine to an area of the program (MT1 etc.) from where I redirect it again using JP instructions-purely for my own benefit while coding. At the moment all that the routine does is jump between two menus, but this serves to illustrate the principle.

If the routine was required only as an 'ON-GOTO' routine, the screen parts could, of course be ommitted, but each group of results would require it's own value for 'MENU'. Hence the routine, if taken to the extreme could cope with 255 groups of results, each group holding 255 different results.

Looking at the diagram, assuming that 'MENU' holds the value 1, the routine will take the value of any key pressed and test it gainst the first value after the first 0 in 'TABLE1' if it does not match, it is checked against the next value until it does match and the program then jumps to the location pointed at by the equivelent place in 'TABLE'. If no possible results match the value of the key, the 0 will be reached and the routine goes back to the 'INKEY' section.

# YOUR LETTERS

*** Games High Score Table ***

| | | | | | |
|---|---|---|---|---|---|
| TOADO | 107549 | N.GOODING | BLOBBO | 71233 | T.PICKSTONE |
| NEMO | 11080 | P.CRIGHTON | OBLOIDS | 60040 | M.GELDER |
| P.PETE | 39630 | A.DOBSON | MISS.ALPH | 43840 | T.PICKSTONE |
| KILOPEDE | 33440 | P.CRIGHTON | GOLDMINE | 6025 | P.CRIGHTON |
| CONT RAID | 10810 | M.GILL | STAR COMM | 90410 | P.CRIGHTON |
| MAXIMA | 252830 | M.GILL | TURBO | 23030 | M.GELDER |
| QOGO 2 | 138960* | A.DOBSON | ASTRO PAC | 69390 | A.DOBSON |
| COBRA | 5634* | A.DOBSON | SNAPPO | 67100* | A.DOBSON |
| T FIGHTER | 2350 | N.CRIGHTON | S M/FIELD | 829 | P.CRIGHTON |
| ASTROMIL. | 3070 | A.DOBSON | PHAID | 1965 | A.DOBSON |
| SON OF PETE | 880 | A.DOBSON | F.DEEP | 1290 | A.DOBSON |
| S.SCANNER | 7340* | A.DOBSON | ICEBERG | 17431 | A.DOBSON |
| KNUCKLES | 488650 | P.CRIGHTON | | | |
| FELIX | 18450 | T.PICKSTONE | | | |
| TAPEWORM | 168515 | A.DOBSON AT LEVEL 1 | | | |
| | 150500 | A.DOBSON AT LEVEL 9 | | | |
| BOUNCING BILL | 219,610* | A.DOBSON LEVEL 1 | | | |
| BOUNCING BILL | 158,334* | A.DOBSON LEVEL 5 | | | |
| SNOWBALL | 450 | P.CRIGHTON | | | |
| AGROVATOR | 61828* | A.DOBSON | | | |

* Denotes new high score.

******

Hints & Tips

1. Ed-> Some time ago Tony Street of Hewelsfield, bought a Pothole Pete from the club, to his dismay (and ours) it seemed not to work, well it did, but really slowly. We promptly sent another copy, this time checking it before sending. (We sat back in confidence, ..a problem well solved?) It was funny though, the program loaded on our machhine!.
A week later, sure as 'eggs is eggs' we heard that it would not load. We sent a Son Of Pete, ..it still didn't work!.
Well, a Hardware problem was expected and a Hardware problem it turned out to be. The whole fiasco was caused by ...having a Pascal ROM fitted. So the word of warning is, if you have a ROM fitted, expect trouble with some of your software.

Questions And Answers

Ed-> I am always asking you to make suggestions for improving the magazine and club, so far I have had two sugestions, they are :-

Why do we not have a regular page of software for sale!, well, initially when Memotech first gave us a dealership, we included a list of software then available. We sold some software but we found that we ended up with amounts of stock that we could not get rid of. We therefore decided to take an advert from MTSC, who's prices are probably the lowest around and they seem to stock just about everything!

We will however make an attempt in the future to find the 'best' software and sell that through the club.

The other suggestion read like this :-
"You asked for suggestions! Well here's one, keep going as you are and you won't go far wrong."
Ed-> Many thanks to Alan Dobson for that inspiration!!

ROM Calls By Richard Dennis

After many hours spread over many months of wading through endless areas of ROM, I have found several useful ROM calls that may be of interest.

Using undocumented ROM calls (and most are undocumented!!) can cause problems due to the manufacturers changing the location of the ROM routines. However, in the absense of any direction from Memotech (will they remedy this in the future?) we are left to our own devices.

To illustrate the use of several of these calls I have written a short routine that illustrates what can be done. The actual program is not in itself anything special, but it will serve to give others an insight into what can be done.

```
4 REM ******************************
5 REM * Prog to illustrate several*
6 REM *     useful ROM calls       *
7 REM *                            *
8 REM *    R.Dennis...May 1985     *
9 REM ******************************
10 CODE
          LD DE,MESSAGE
          CALL £28DC      ;Get input from KBD buff
          PUSH HL
          POP DE          ;DE=KBD buff address
          RST 30          ;Convert Decimal to Hex
          LD (ADDRESS),BC;Store Hex Number
          LD A,"£"
          CALL £CAB       ;Send Reg A as ASCII
          CALL £1B50      ;Send BC to scrn as Hex
          RST 10
          DB £19          ;Cursor forward 1 space
          LD HL,(ADDRESS);Addr from I/P
          LD A,(HL)       ;Get byte from memory
          CALL £1B55      ;Send Reg A to Screen
          RET
MESSAGE:  DB £22,"ENTER DECIMAL ADDRESS ",£22,£FF,£FF
ADDRESS:  DS 2
20 REM Note this program does not check for incorrect
entry
```

# ROM CALLS

The two following routines were sent to me from a 'Hacking' member (Ed-> Sorry I've lost your name!), I think you may find them very handy if you're into assembler programming.

```
100 REM ********************************
110 REM * ROM MESSAGE PRINTING  V2     *
120 REM * Reg DE points to start of    *
130 REM * message, then call £02E8     *
140 REM *                              *
150 REM * Last byte of each message    *
160 REM * must be £FF.                 *
170 REM ********************************
200 CODE

812F            LD DE,STR1 ; String No.1
8132            CALL £02E8 ; Print it
8135            LD DE,STR2 ; String No.2
8138            CALL £02E8
813B            RET
813C STR1:      DB "STRING PRINTING, AT ROM ADDRESS £02E8",£FF
8162 STR2:      DB 10,13    ; CRLF
8164            DB "LAST BYTE OF STRING = £FF",£FF
817E            RET

Symbols:
STR1    813C    STR2    8162


100 REM ********************************
110 REM * ROM INPUT ROUTINE  £1B85     *
120 REM * Characters input are stored*
130 REM * in the keyboard buffer.      *
140 REM * System var. at £FA83 points*
150 REM * to start of keyboard buffer*
160 REM * On return from £1B85 Reg C  *
170 REM * = number of characters in    *
180 REM * buffer, last entry = £FF     *
190 REM ********************************
200 CODE

8179            LD DE,MESS1
817C            CALL £02E8
817F            CALL £1B85 ; Get input
8182            LD DE,MESS2
8185            CALL £02E8 ; Print message 2
8188            LD DE,(£FA83)       ; Address of buffer
818C            CALL £02E8 ; Print input name.
818F            RET
8190 MESS1:     DB "ENTER YOUR NAME ",£FF
81A1 MESS2:     DB 10,10,13
81A4            DB "YOUR NAME IS....",£FF
81B5            RET

Symbols:
MESS1   8190    MESS2   81A1
```

# SOFTWARE REVIEWS

ICEBURG.
Publisher SyntaxSoft
Price (About) £4.95

Float around the sleepy lagoon merrily destroying iceburgs with this 'wet' version of the old classic 'Asteroids'.

This is a nice simple game for all ages, so don't expect too much, nothing shoots at you but you can be eliminated by touching an iceburg. (Be warned most of the iceburg is below the surface.)

The graphics are uncomplicated, only a two colour display. In the introduction you are warned to hurry before the blizzard arrives, but when it does nothing happens !!

A great feature of this game is the ability to switch off the sound, I wish more games featured this option. The sound, by the way, consists of bleeps, warbles and buzzes - you control the volume.

You also can select a slow game, but I think most players would not need this option. Both myself and my 4 year old daughter found it easier to play using joysticks.

This is not a game you would rush home each night to play but nevertheless nice to play now and again.

Playability =4
Lasting Interest =3
Graphics =2
Value for Money =3

Reviewed by Tony Street (& Co.!)

Bouncing Bill by Brian Rogers
This game is infuriatingly simple, but, be warned it can become an obsession. High scores must be possible, somehow I managed 19218 in just 10 minutes of play.

The game consists of 10 horizontal lines scrolling in both directions across the screen. Bill starts at the bottom and using the cursor keys must jump throu' gaps in successive lines to reach the top for a bonus score. Each time Bill manages to climb to the top, increasing numbers of white balls roll down the screen knocking poor Bill flat and causing him to fall throu' the gaps and, if he is really unlucky, right to the bottom to lose a life.

There are 5 levels of difficulty but I didn't find much difference between them. In common with 'Iceburg' the graphics are simple, colours limited and the sound bearable. I can recommend this game especially for those family computer evenings where you can share a laugh or two with others.

Rating
Playability - 4
Graphics - 2
VFM - 3
Lasting Interest - 3

Reviewed by Tony Street & Co.

Miner Dick
Publisher: Xaviersine
Price: £6.95 (I think!)

This review very nearly wasn't, as we had great difficulty in loading the game (we only managed to load it after adjusting the azimuth on the tape head). Once we had overcome this initial difficulty, it was pleasing to see a title picture instead of the usual blank screen on loading (although this does add to the overall loading time).

The game itself bears a "slight" resemblance to a game that was moderately popular on the Spectrum (sorry we had to mention it!) (Ed->Arrhhh!) a while back called "Manic Miner". The plot is identical, progressing through 20 screens by collecting all the keys and avoiding things such as the "sly spiders" then leaving through the flashing exit before the air runs out.

The graphics in this game are well above average, with many varied meanies, the sound is quite good too, with a catchy tune playing continously (although it can be turned off if desired). There are other features, such as a demo mode, 1 or 2 player option (using L and R joysticks) and the ability to start on any of the screens which have been completed.

There are a couple of bugs in this game, one being that his positioning is not quite correct giving him the ability to stand in mid air to the right of a block and fall through the left hand edge, although there are a couple of screens which are impossible without this. The other is that the rub-out on the name entry for the high score table does not work!.

Conclusion
The game is fairly addictive although with only 20 screens the lasting appeal could be limited. The aforementioned bugs are not serious, but can be irritating until you get used to them.

Playability 4
Graphics 4+
VFM ?
L Interest 3+

Reviewed By Ian Heath
& (The Heme! Hackers)
Sean Newman

# SILICON DISC REVIEW

By

Phil Eyres

Last month I took delivery of two 250k (formatted) Silicon discs, and as promised, I have reviewed them.

The Silicon board is a fairly large piece of hardware, measuring about 6" * 8", it is designed to fit into the card cage in the FDX main unit. Fitting is relatively simple, just remove the twelve screws holding the case together, this then reveals the card cage which is accessible by removing another couple of screws. If as in my case the necessary edge connectors are not present, then you will have to buy some and solder them in, this makes the task of fitting much more complicated as the main unit will have to be totally dismantled. This part alone took me 3+ hours and for most of the time my heart must have thought it was running a marathon!.

Once they were physically installed inside the machine, it was all 'bolted' back together, ready now for the big switch on ...and my heart was going for it's second marathon! Power up did not infact change the machine in any way, as the system CONFIGuration had not been altered to inform the machine that I had these new Silicons present. This is all quite easy (when you read the manual!) and within half an hour I had the machine recognising two 5 1/4 floppies and a 512K Silicon disc. (Although I bought two 256K formatted discs, they are read as one 512K disc when inside the machine). Before I could use my new drive, designated drive F, I had to format it, so I did just that, I took about a second! ...Um! quite fast (I thought!). I then copied the contents of a Floppy onto it, even more surprised, it's data transfer rate was starting to impress me. I loaded my Pascal onto it and tried compiling a few large source files on it, great, it reduced compilation times to about 5% of what they are to compile to Floppies. Then I got a bit over confident!, I made a new program up and compiled it (a few times, just for the hell of it!), time was getting on ... well into the early hours!, by pure impulse I removed the Floppies and switched off. Arrrhhh!, No!, Dammmit!!!, I forgot to copy the contents of the Silicon disc to a Floppy, thus in one easy movement I'd lost about 4 hours work!.

Anyway, speed was one reason for buying the Silicon discs, the manual described their use as a print spooler, that is, load them up with data to be printed out and the machine will take care of everything whilst I continue working on other documents, or at least that is the theory. To use it as a 'Spooler' you have to again change the System using Config, spooler is then invoked by typing (believe it or not) "SPOOLER". All seemed well, so I loaded a text file into the silicon's memory, it started to printout, great (I thought), then it stopped!, half way through a printout, it just sat there for a few minutes while I thumbed through the manual for the umteenth time, when I pressed a few keys it would rather erratically print out another line of text, Well this is 'garbage' I thought, it would take all day just to print one letter. I loaded Neword in the hope that this would prove to be more fruitful, but it turned out even worse, it would print a couple of lines and then that was your lot, no more print what-so-ever came out. I am still rather disappointed about this side of it as it would have been very handy.

Another 'let-down' is that with the Silicon configuation present on the system tracks I cannot use Basic, the blue screen comes up but it is 'promptless' and stubbornly remains so regardless of how much I curse at it!. This means that I have to keep discs with my old system on them just to use Basic. (All rather Messy).

The manual also says that it is possible to upgrade the memory capacity by fitting 'larger' chips, say 256K chips instead of the 64K chips, I would like to see someone do this as the chips are soldered directly to the PCB and I would say that de-soldering that lot would be impossible!. I don't think that I shall worry about this too much as the 512K that I have is just about perfect for my needs.

The price of the Silicons is infact quite reasonable at £165 for the 256K version and £275 for the 512K version. (Note :- These prices only apply through the club and are for a limited period of 2 months!).

******

Just a little note to fill a spare corner :-
Did you know that if you have one of the new single disc's you loose the use of the Basic command USER. This seems a bit of a shame as quite a few utilities I have use this.

Phil

******

# P R O G R A M  L I B R A R Y

I must say that over the past month I have been totally overwhelmed with the response for software from the library, at a rough count, I reckon I've sent out over 50 tapes. I hope this months contribution of 5 new titles brings the same response. To obtain anything from the library send £1 per cassette, 2 programs per cassette, one on each side. As I will be on holiday for two weeks next month, deliveries will probably take a little longer than usual, but I will do my best to ensure the shortest possible delays.

Phil

1.Basic & Assembler Programs

1.Hex-Dec-Bin Convertions. (Binary Bit In Assembler)
2.CGEN Sprite Generator.
3.3D Drawing Board. Rotate a skeleton of a cup & saucer in 3D.
4.Whist. The Card Game
5.Memory Save. This Utility will Save a block of memory to tape and retrieve it.
6.MTX Drawing Board.
7.LOGO Drawing Board.
8.Simplex Tablaeux. Applications Program
9.Breakeven. Applications Program
10.Statistics Applications Program
11.An Unsolved Problem Applications Program
12.Radio Routines Applications Program
13.Light Cycles. Arcade Game
14.Hex/Dec Dec/Hex conversions using USER commands!
15.Renumber II Renumbers Including GOTO's etc
16.RELOC Relocs Assembler Properly!!
17.Character Editor
18.Quasimodo Excellent Arcade Game
19.Planner **New** YASG (Yet Another Sprite Generator)
20.Hanoi **New** Classic Puzzle (Brilliant simple use of
21.Noble **New** Simple Text Game Graphics)
22.Hi-Lo **New** Just like Bruce's Play Your Cards Right
23.Composer **New** Our First Sound Generator!!

3.Articles From Previous Magazines
1.PANEL2 Utility. As above but updated to include a second feature.
2.Undocumented Neword dot commands.(Vol1 Iss.7)
3.Hisoft Pascal Review (vol1 Iss.8)
4.Neword Rom Review (Vol1 Iss.5)
5.RST10 Codes Explained (Vol1 Iss.3)
6.VDP Explained Using assembler (vol1 Iss4,5,6)
7.System Variables (Not Previously Published!!)

5.Program Reviews

Planner
This is I believe our third Sprite Generator, in it's own way it is the best, although I don't think it will be of much use to anyone without a printer!. Allows for Sprites up to 16*16, generates both sprite and Genpat codes at printer. All usual features.

Hanoi
Well, I thought I had better load this and try it out,...am I glad I did!, it is absolutely brilliant. The screen layout is faultless, I challenge anyone to better it, for that matter the key press layout is so simple a four year old could 'handle it', also for a basic program the key de-bounce is perfect. If your'e looking for a program to impress computer illiterates and have a few hours fun with a couple of friends, then this is the one.Superbly simple graphics that are really smooth. You'll need to see it to believe it.

Noble
This is I suppose, a very simple text game, it is infact much like the Farmer program published by us a long time ago in one of our first mags. You,'A Nobleman' are placed in several situations, your decisions controlling your outcome, ...usually hanging!!!.

HI-LO
This is a very good card game. It works on the principles of Bruce Forsyths T.V. game Play Your Cards Right. The graphics are nice and colourful and the action smooth. This one along with Whist and Hanoi make a good set of colourful graphics based Basic games.

Composer
This is our first Sound Composer program, infact I believe it is our first Sound program of any sort. It comes with Instruction and Explanation pages to help those who do not know the format of sound generation. The actual program allows up to 3 Sound channels to be exploited, all sounds can be tried out at the press of a button, sound is also cut off by the press of button. I found this quite an interesting program as I know only a little about programming Sound. I think that it would be a help to anyone wanting to learn a bit about the MTX sound commands.

## Abridged Terms & Conditions (Downloads)

### Disclaimer

www.primrosebank.net, (***the website***) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

***Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download.*** Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

***Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.***



As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. ***This material is likely still under copyright of the original author and/or publishers.*** The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave Stevenson at the e-mail address above.