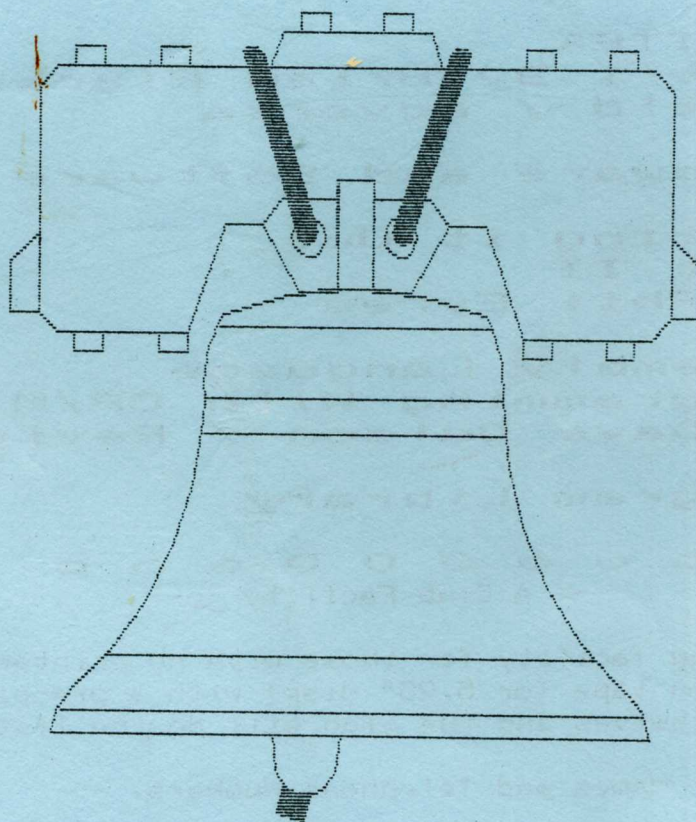


VOL. 3 ISSUE 9

AUGUST 1987

THE
MEMOTECH OWNERS CLUB
MAGAZINE
MEMOTECHNIQUES



PUBLISHED BY: MEMOTECH OWNERS CLUB
13 CORSE ROAD
TOWNHILL PARK
SOUTHAMPTON

CIRCA ...279

M.O.C.

VOLUME 3 ISSUE NUMBER 9

CONTENTS

1. Editorial
2. Spaceship Part II
3. FLITTER
4. Part I By Brian Clarke
5. Cont'd / Adverts
6. Hardware and Software
7. Sorting it Out
8. Part II
9. By Phil Eyres
10. Assembly Language
11. Programming With CP/M
12. By Dave Dulson / Reviews
13. Program Library

o o o o o o o o o o
--- A Club Facility ---

A program listing facility for those with no printer. Just send in your program on tape (or 5.25" disc) with a pre-paid envelope addressed to yourselves and the club will do the listing for you.

--- Names and Telephone Numbers. ---

i. Paul Wood for 3.5" disc copying, general info and Comms specific info.

Tel 0905 24260

ii. Alan Dobson for help with the following adventures:

Alice, The ZOO and Man From Granny

Tel 061-980-6288

If anyone has any good graphics designs for a front cover then we would love to see them!!!

Phil Eyres
13 Copse Road
Townhill Park
Southampton

Firstly, news about the club!!

Having now settled in to my new home, I have had the time to impliment a few ideas that I have had for a while:-

1. I have successfully set up office in the spare room, and for once, can actually find most things when I need them.

2. The burden of photocopying club literature has become ever growing over the past months, up until now a nameless photocopier has helped out very well. But the other day a friend offered me a proper business copier at a good price, this machine now takes pride of place in my new 'office'. I hope that this will aid mail through-put in the coming months.

3. In order for the club to continue forward and not just 'plod-on', it needs new ideas and enthusiasm, so as a few members have offered their services recently, I formulated a plan of attack.

i. Alan Hamilton has kindly agreed to take over the Program Library (This will happen in a month or so, as soon as he has had a chance to play with it a bit). I hope this will provide a better library with new ideas. It will also free me to spend more time on producing the magazine and hopefully think up some interfacing projects.

ii. Alan is also at this time trying to contact National magazines in order to get more club awareness (Another area which has been neglected!).

iii. Reviews of software will be more frequent in the magazine as this has been asked for by many, and we have a couple of new reviewers to help with this. (Please keep pestering me for more tapes to review!!).

iv. Magazine articles; As you can see this month, we have articles the like of which have been all to sparce in the past.

v. I'll leave this one open this month. It is for all members to make suggestions!! I'm open to think about anything.

Now onto SYNTAX. SYNTAX is dead!; Long live ORION software. This company appears to be run by Mrs Hook, and I must say that so far, software deliveries have been very good. (I'll try and find room for a full list of software next month).

Now for some MEGA news, we can supply to YOU, John Grayson's ASSEMBLY LANGUAGE COURSE for only £7.00. So if anyone has been thinking of learning about assembler, well now you have no excuse. (I know, you'd like a review; hopefully next month!).

Nearly out of breath (and out of space) we come to the answers to the competition.

The correct answer was:-

A=3, B=6, C=2, D=7, E=8, F=1, G=5, H=4.

The winner, drawn randomly from a hat containing the correct answers was Neil Gooding, who wins a copy of Highway Encounter.

This month I have another competition. The winner will be the person sending in the best program running a DECIMAL CLOCK!. You can use your imagination as much as you like so long as a day runs from 'moon to moon'.

I can read/write only disc's in 5.25" format and up to 500K, if anyone with 3.5" systems would like something from the club or has something to offer on 3.5" format please send to Paul Wood, his address is listed opposite.

As I am more often than not out playing squash on Mondays between 6 and 7 pm, I feel it would be best to move the Hotline to after 7.40pm. I hope this is ok for everyone. The number to phone now is (0703) 585106, ask for Phil. However, feel free to phone any evening after 6pm.

If anyone would like back issues they are available for the small remittance of 80p each. At present there are 28 back issues, 10 for volume 1, 10 for volume 2 and 8 for volume 3.

It should be noted that all articles are the copyright of the sender and M.O.C., anyone wishing to have articles published elsewhere should inform us first.

'SPACESHIP' by John Grayson, GRAYsoft.

---000---

Last month I gave an explanation and listing to produce a spaceship to fire, move and react as if it were in space. As promised, this article demonstrates how to create a game by incorporating asterioids and a base.

The object of the game is to land the spaceship onto the base. This can only be performed by moving the ship in exactly the same direction as the base, and only just faster than the base. The asterioids can make this task difficult because they home onto the spaceship. If an asterioid gets destroyed by the spaceship, then another one takes its place, but if either an asterioid hits the ship, or if the ship is moving too fast or not in the same direction as the base; then the game ends.

There is one level of play, but three screen types to combat. The first is just a case of landing onto the base, the second has one asterioid as well as the base. The third screen and beyond has two asterioids, so is harder. Any additional screens have two asterioids.

---000---

```

0 REM          <SPACESHIP>
1 REM          <PROGRAMMED BY JOHN GRAYSON,GRAYSOFT>
2 REM          <MODIFICATIONS TO LAST MONTH'S PROGRAM>
3 GOTO 20
5 IF INCREMENT=1 THEN GOTO 8 ELSE LET LOOP=0
7 LET LOOP=LOOP+1: IF LOOP=INCREMENT THEN GOTO 19 ELSE
IF ABS(HOR-AH(LOOP))<5 AND ABS(VERT-AV(LOOP)
8 IF ABS(DOCKH-H)<.16 AND ABS(DOCKV-V)<.16 AND ABS(HOR-
DH)<5 AND ABS(VERT-DV)<5 THEN LET DOCK=0: GOTO 9 ELSE IF
INCREMENT=1 THEN RETURN ELSE GOTO 10
9 LET INCREMENT=INCREMENT+1: IF INCREMENT=4 THEN LET
INCREMENT=3: GOTO 30000 ELSE GOTO 30000
10 IF ABS(FIREH-AH(LOOP))<4 AND ABS(FIREV-AV(LOOP))<4 THEN
GOTO 11 ELSE GOTO 7
11 SPRITE 6+LOOP-1,20,AH(LOOP),AV(LOOP),0,0,13: PAUSE 500:
SPRITE 4,16,888,888,0,0,1: SPRITE 5,17,888,888,0,0,1
12 LET TIMER=0: LET FIRE=99
13 LET AH(LOOP)=INT(RND*256): LET AV(LOOP)=INT(RND*192)
15 IF ABS(AH(LOOP)-HOR)<15 THEN GOTO 13 ELSE IF
ABS(AV(LOOP)-VERT)<15 THEN GOTO 13
19 RETURN
31 LET AH(1)=INT(RND*256): LET AV(1)=INT(RND*192)
32 IF ABS(AH(1)-HOR)<15 THEN GOTO 31 ELSE IF ABS(AV(1)-
VERT)<15 THEN GOTO 31
33 LET AH(2)=INT(RND*256): LET AV(2)=INT(RND*192)
34 IF ABS(AH(2)-HOR)<15 THEN GOTO 33 ELSE IF ABS(AV(2)-
VERT)<15 THEN GOTO 33
311 GOSUB 5: GOSUB 20000
330 SPRITE 1,T,HOR,VERT,0,0,9: GOSUB 30000
    
```

BREAKDOWN OF PROGRAM, AND HOW IT ALL WORKS:
LINES COMMENTS

7-19 Subroutine to detect if two sprites collide.
31-34 Random positions of asterioids on screen.
20000-20100 Subroutine to move asterioids.
30000-30090 Subroutine to move base - displayed as a
coloured square.

VARIABLES USED:

INCREMENT - Which level we are on. If INCREMENT=1
then no asterioids are on the screen.
AV(n) & AH(n)- Horizontal and vertical coordinates of
asterioids.
DOCK - If zero, base landed, recalculate
velocity and coordinates for next base.
If DOCK=1 then move base.
DOCKH & DOCKV- Horizontal and vertical positions of base.

Below are the amendments needed for the program
printed last month, as well as the rest of the listing.

If there are any queries, post to: John Grayson,
GRAYsoft, Cambalt,Potters Heron Lane, Ampfield, Romsey,
Hampshire.S051 9BW.

```

350 IF INCREMENT<>1 THEN GOSUB 10000
360 GOSUB 30000
365 GOSUB 5
10512 GOSUB 5
20000 REM <MOVE ROCKS>
20001 IF INCREMENT=1 THEN GOSUB 5: RETURN ELSE LET
N=N+1: IF N=INCREMENT THEN LET N=0: GOTO 20100
20002 GOSUB 10000
20005 IF AH(N)>HOR THEN LET AH(N)=AH(N)-2 ELSE LET
AH(N)=AH(N)+2
20010 IF AH(N)>252 THEN LET AH(N)=3 ELSE IF AH(N)<3 THEN
LET AH(N)=252
20020 IF AV(N)>VERT THEN LET AV(N)=AV(N)-2 ELSE LET
AV(N)=AV(N)+2
20030 IF AV(N)>188 THEN LET AV(N)=3 ELSE IF AV(N)<3 THEN
LET AV(N)=188
20040 SPRITE 6+N-1,21,AH(N),AV(N),0,0,3: GOTO 20001
20050 GOSUB 5
20100 RETURN
30000 REM <DOCKING BAY>
30020 IF DOCK=0 THEN LET DOCK=1: LET DOCKH=(RND*3)-1.5:
LET DOCKV=(RND*3)-1.5: LET DH=INT(RND*256): LET
DV=INT(RND*192)
30040 LET DH=DH+DOCKH: LET DV=DV+DOCKV: SPRITE
10,22,DH,DV,0,0,(RND*13)+2
30070 IF DV<0 THEN LET DV=191 ELSE IF DV>191 THEN LET
DV=0
30080 IF DH<0 THEN LET DH=255 ELSE IF DH>255 THEN LET
DH=0
29962 RETURN
    
```

From Brian Clarke; 5, Quickswood, Luton.
Phone (0582)-576680 (Evenings only, after 7pm)

Introducing the game of FLITTER

The basis of the game is that two players each use a joystick to control their respective "CONSTANT RATE ATTACK PERSONNEL SHIP" [CRAPS]. There are 3 screens, the central (VS6) shows a 3D (ish) forward view of both craft, each side screen (VS4 Left & VS5 Right) a crude plan of the two craft, with the respective players craft centre; also show power (vertical bar) and shots available (shown as one * per shot).

On the central screen there are 5 zones, this is where the "COntstant RATE Limiter" (CORAL) can be temporarily overridden - the higher up you are, the thinner the atmosphere, the faster you go. Similarly when you approach the ground, in the thicker atmosphere, the CRAPS slow down. (!!!!!!!!!!!).

The object is to shoot down your opponent. Your guns are both forward and rear firing, and the range is shown on your side screens.

As time goes (slowly) on, the CRAPS convert solar energy into shield energy, which is leeches into your gun power banks.

The game is written completely in basic, and is rather slow and boring. However, over the next few months, I will convert the game, virtually line by line, into machine code, giving where possible, (very brief) explanations and comments. This will then make the game fast and boring! The hope is that this will encourage more of you to take up simple machine code programming, and make your millions. (Don't forget me in your will.)

Finally for this month (I heard that 'thank God' from the back) the programme is sectioned by REM's - most of the sections are self-explanatory, I will go into detail as I convert them. For now I give you a list of Variables used (SEE REM VARS) :-

- AA KEYBOARD INPUT (See REM KEYS - the way, in BASIC that you can access multi key inputs).
- SPLN SPRITE LEFT NUMBER, which with
- SPSN SPRITE SPARE NUMBER, defines which sprite is in front of the other.
- LF & RF FIRE TOGGLE (1=fired, 0=not fired).
- LX & RX X-POSITION OF SPRITES (CRAPS)
- LY & RY Y-POSITION OF SPRITES.
- LZ & RZ Z-POSITION OF SPRITE (Distance from players).
- LG & RG GUN POWER (max 10 - incremented by 1 each cycle).
- LP & RP SHIP POWER (SHIELDS) (max 150 - incremented by 5).
- LS & RS SHIP HIT TOGGLE (FAN HIT THE CRAPS ?) value of 40, deducted from power if ship hit.

That's all for now - have fun typing - even more fun understanding the programme - and wait a few weeks for the next CRAPS installment.

The Listing

```

10 GOTO 900
80 REM FDRAW
90 LINE 29,40,29,157: LINE 32,40,32,157: RETURN
100 REM KEYS
110 OUT (5),223: LET AA=INP(5): LET RF=-(AA=127)
120 OUT (5),247: LET AA=INP(5): LET RX=RX+(AA=127)
130 OUT (5),239: LET AA=INP(5): LET RX=RX-(AA=127)
140 OUT (5),251: LET AA=INP(5): LET RY=RY+(AA=127)
150 OUT (5),191: LET AA=INP(5): LET RY=RY-(AA=127)
160 OUT (5),127: LET AA=INP(6): LET LF=1-(MOD(INT(AA),2))
170 OUT (5),127: LET AA=INP(5): LET LX=LX-1+MOD(INT(AA),2)
180 LET LX=LX+1-MOD(INT(AA/2),2)
190 LET LY=LY+1-MOD(INT(AA/4),2)
200 LET LY=LY-1+MOD(INT(AA/8),2)
210 REM LIMITS
220 IF LY<56 THEN LET LZ=LZ-2: IF LY<16 THEN LET LZ=LZ-
2: IF LY<7 THEN LET LY=7
230 IF RY<56 THEN LET RZ=RZ-2: IF RY<16 THEN LET RZ=RZ-
2: IF RY<7 THEN LET RY=7
240 IF LY>135 THEN LET LZ=LZ+2: IF LY>175 THEN LET
LZ=LZ+2: IF LY>183 THEN LET LY=183
250 IF RY>135 THEN LET RZ=RZ+2: IF RY>175 THEN LET
RZ=RZ+2: IF RY>183 THEN LET RY=183
260 IF LX<83 THEN LET LX=83 ELSE IF LX>173 THEN LET
LX=173
270 IF RX<83 THEN LET RX=83 ELSE IF RX>173 THEN LET
RX=173
280 IF LZ<32 THEN LET LZ=32 ELSE IF LZ>250 THEN LET
LZ=250
290 IF RZ<32 THEN LET RZ=32 ELSE IF RZ>250 THEN LET
RZ=250
300 IF LZ>RZ THEN LET SPLN=3: LET SPSN=1 ELSE LET
SPLN=1: LET SPSN=3
310 REM FIRE
320 IF LG<4 THEN GOTO 330 ELSE IF LF=1 THEN GOSUB 600
330 IF RG<4 THEN GOTO 340 ELSE IF RF=1 THEN GOSUB 650
340 REM POWER
350 LET LP=LP-LG+5: LET LS=0: IF LP>150 THEN LET LP=150
ELSE IF ((LP<41) AND (LP+LG>40)) THEN LET LP=LP+LG: LET
LG=0
360 LET RP=RP-RG+5: LET RS=0: IF RP>150 THEN LET RP=150
ELSE IF ((RP<41) AND (RP+RG>40)) THEN LET RP=RP+RG: LET
RG=0
370 IF LG<10 THEN LET LG=LG+1: LET LP=LP-1
380 IF RG<10 THEN LET RG=RG+1: LET RP=RP-1
410 REM SCRNUF - SCREEN UPDATE
420 SPRITE 5,10,32,(RZ+RZ-LZ-LZ+95),0,0,6: SPRITE
4,10,224,(LZ+LZ-RZ-RZ+95),0,0,12
430 SPRITE 2,(RZ/32),RX,RY,0,0,6: SPRITE
SPLN,(LZ/32),LX,LY,0,0,12: SPRITE SPSN,10,120,1,0,0,0
440 VS 4: LINE 60,40,60,LP+40: LINE 61,40,61,LP+40: ATTR
2,1: LINE 60,LP+41,60,191: LINE 61,LP+41,61,191: ATTR 2,0
450 CSR 5,22: IF LG<4 THEN PRINT " "; ELSE IF LG=4 AND
LG<8 THEN PRINT " *"; ELSE IF LG=8 THEN PRINT "**";
460 VS 5: LINE 4,40,4,RP+40: LINE 3,40,3,RP+40: ATTR 2,1:
LINE 4,RP+41,4,191: LINE 3,RP+41,3,191: ATTR 2,0
470 CSR 2,22: IF RG<4 THEN PRINT " "; ELSE IF RG=4 AND
RG<8 THEN PRINT " *"; ELSE IF RG=8 THEN PRINT "**";
500 REM CRASH - COLLISION - END LOOP
510 IF (ABS(RZ-LZ)<2 AND ABS(RX-LX)<4 AND ABS(RY-LY)<2)
THEN GOTO 850
520 IF LY>55 AND LY<136 AND RY>55 AND RY<136 THEN LET
RZ=RZ+SGN(288-RZ-LZ): LET LZ=LZ+SGN(288-RZ-LZ)
530 IF LP<1 THEN GOTO 700 ELSE IF RP<1 THEN GOTO 750
540 GOTO 100
600 REM LFIRE
610 VS 4: ATTR 3,1: GOSUB 90
620 LET LG=LG-4: IF ((ABS(LZ-RZ)<32) AND (ABS(LX-RX)<10)
AND (ABS(LY-RY)<8)) THEN LET RS=40
630 GOSUB 90: ATTR 3,0
640 RETURN
650 REM RFIRE
660 VS 5: ATTR 3,1: GOSUB 90
670 LET RG=RG-4: IF ((ABS(LZ-RZ)<32) AND (ABS(LX-RX)<10)
AND (ABS(LY-RY)<8)) THEN LET LS=40
680 GOSUB 90: ATTR 3,0
690 RETURN
700 REM RWINS
710 IF RP<1 THEN GOTO 800
720 VS 6: PRINT "RIGHT SHIP WINS"
730 IF INKEY*(">") THEN GOTO 730 ELSE PRINT "PRESS ANY
KEY TO PLAY AGAIN"
735 IF INKEY*("<") THEN GOTO 735
740 IF INKEY*("=") THEN GOTO 740 ELSE GOTO 1000
750 REM LWINS
760 VS 6: PRINT "LEFT SHIP WINS": GOTO 730
800 REM NOWIN
810 VS 6: PRINT " WHOOPS": PRINT "BOTH SHIPS SHOT"
820 GOTO 730
850 REM BANG - COLLISION
860 VS 6: PRINT " YOU TWO FLIERS HAD BETTER TAKE UP TABLE
TENNIS YOU CRASHED INTO EACH OTHER": GOTO 730
870 CLEAR : USER SAVE "PROG1.BAS"
880 REM SPRCTRL
890 CTLSPR 2,5: CTLSPR 6,3
900 REM PATTERNS
905 CTLSPR 2,5: CTLSPR 6,3
910 GENPAT 4,7,0,0,0,0,0,0,0,0
920 GENPAT 5,7,3,0,0,0,0,0,0,0
930 GENPAT 6,7,0,0,0,0,0,0,0,0
940 GENPAT 7,7,128,0,0,0,0,0,0,0
950 GENPAT 4,6,0,0,0,0,0,0,0,1
960 GENPAT 5,6,7,0,0,0,0,0,0,0
970 GENPAT 6,6,0,0,0,0,0,0,0,0
980 GENPAT 7,6,192,0,0,0,0,0,0,0
990 GENPAT 4,5,0,0,0,0,0,0,0,9

```

Continued Overleaf

1000 GENPAT 5,5,7,2,0,0,0,0,0
 1010 GENPAT 6,5,0,0,0,0,0,0,32
 1020 GENPAT 7,5,192,128,0,0,0,0,0
 1030 GENPAT 4,4,0,0,0,0,0,1,19
 1040 GENPAT 5,4,15,4,0,0,0,0,0
 1050 GENPAT 6,4,0,0,0,0,0,0,144
 1060 GENPAT 7,4,224,64,0,0,0,0,0
 1070 GENPAT 4,3,0,0,0,0,0,1,35,19
 1080 GENPAT 5,3,15,4,0,0,0,0,0
 1090 GENPAT 6,3,0,0,0,0,0,0,136,144
 1100 GENPAT 7,3,224,64,0,0,0,0,0
 1110 GENPAT 4,2,0,0,0,0,1,1,67,35
 1120 GENPAT 5,2,31,12,0,0,0,0,0
 1130 GENPAT 6,2,0,0,0,0,0,0,132,136
 1140 GENPAT 7,2,240,96,0,0,0,0,0
 1150 GENPAT 4,1,0,0,0,0,3,1,195,35
 1160 GENPAT 5,1,31,12,12,0,0,0,0
 1170 GENPAT 6,1,0,0,0,0,128,0,134,136
 1180 GENPAT 7,1,240,96,96,0,0,0,0
 1190 GENPAT 4,10,0,0,0,0,1,3,3,7
 1200 GENPAT 5,10,6,14,12,4,0,0,0
 1210 GENPAT 6,10,0,0,0,0,0,128,128,192
 1220 GENPAT 7,10,192,224,96,64,0,0,0
 1230 GENPAT 0,119,3,3,15,15,15,63,63

1240 GENPAT 0,120,0,0,192,192,192,240,240
 1250 GENPAT 0,121,60,60,252,252,240,240,48,48
 1260 GENPAT 0,122,240,240,252,252,60,60,48,48
 1400 REM SCREENS
 1410 CRVS 4,1,0,0,8,24,32
 1420 CRVS 5,1,24,0,8,24,32
 1430 CRVS 6,1,8,0,16,24,32
 1440 VS 4: COLOUR 0,10: COLOUR 1,12: COLOUR 2,10: COLOUR
 3,12: COLOUR 4,10
 1450 VS 5: COLOUR 0,10: COLOUR 1,6: COLOUR 2,10: COLOUR
 3,6: COLOUR 4,10
 1460 VS 6: COLOUR 0,1: COLOUR 1,15: COLOUR 2,1: COLOUR
 3,4: COLOUR 4,10
 1470 VS 4: CLS : CSR 3,11: PRINT "wx": CSR 3,12: PRINT
 "yz"
 1480 VS 5: CLS : CSR 3,11: PRINT "wx": CSR 3,12: PRINT
 "yz"
 1490 VS 6: CLS : LINE 0,175,127,175: LINE 0,135,127,135:
 LINE 0,56,127,56: LINE 0,16,127,16
 1500 REM VARS
 1510 LET AA=0: LET LF=0: LET RF=0: LET LG=0: LET RG=0: LET
 RP=0: LET LP=0: LET LS=0: LET RS=0
 1520 LET RX=148: LET LX=108: LET RY=95: LET LY=95: LET
 LZ=128: LET RZ=128
 1530 LET SPLN=3: LET SPSN=1
 1600 GOTO 100

INTERFACING PROJECTS

Why not make a break which will lead you into the exciting world of micro electronics. Infact what better way to start than with an MOC D.I.Y. kit. Everything you need is supplied, except a soldering iron, wire cutters and of course a few hours of your time!! So why not order now.

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 Synthesiser kit-->18.00
 Connecting cable for the internal port (needed for projects) -->£4.50
 All prices are fully inclusive. Please allow 14 days for delivery and make cheques payable to MOC.

 " Software available only from GRAYsoft "
 "-----
 " ADVANCED GRAPHIC DESIGNER. Create graphic screens & scale drawings "
 " :Cassette £13.99 SDX 5.25" disk £15.85 "
 " MTX FRUIT MACHINE :Cassette £4.30 SDX 5.25" disk £5.30 "
 " 3D SPACE LINES :Cassette £4.30 SDX 5.25" disk £5.30 "
 " All three programs :Cassette £18.99 SDX 5.25" disk £22.99 "
 " AD.GRAPH.DES + one game :Cassette £16.50 SDX 5.25" disk £19.30 "
 " FRUIT M/C + 3D SP.LINES :Cassette £6.99 SDX 5.25" disk £8.99 "
 "-----
 " Cheques payable ONLY to JOHN GRAYSON please. "
 " Cambalt, Potters Heron Lane, Ampfield, Romsey, Hampshire. SO51 9BW "
 "-----

HARDWARE AND SOFTWARE PRICE LIST

Basic Computer
256K Computer + Tape operating
System £99.95

System One
1 Mbyte 3 1/2" Drive + I/F £166.00

System Two
1 Mbyte 3 1/2" Drive + I/F
512K Silicon Disc, 80 Col.
+ CP/M + Neword £264.00

HX12 Green Screen Monitor £95.00

Twin RS232 Serial Interface £29.95

We can offer DMX 80 printer ribbons for only £7.00 each, so why not order one today and be prepared for the day your ribbon finally 'bites the dust'!!!

The MTX FIG-FORTH requires an MTX512 or expanded 500, the dictionary associated with Forth is held as part of the Ram-Disc which can be saved separately, fairly quickly. The Ram-Disc allows for 24 'edit' screens to be created and in memory simultaneously. A tutorial will be necessary for the beginner, for this the club has obtained a quantity of the publication Fundamental Forth (This may vary according to availability).

Fig-Forth Program £6.00
Tech Data Sheets £2.00
Tutorial Book £7.50 (240 pages)

Cheques payable to MOC please, orders from stock normally despatched by return, else, please allow 2 working weeks.

Ron Gladwin of UK Home Computers, (Tel 0793 695034) has on offer a Spectrum Loader that will convert your 512 into a 40K Spectrum, this will allow you to type in Spectrum programs, it is also said to load some Spectrum software. At £2.95 you may find it worth a try.

Ron also has some FDX Silicon disc's on offer, he has both 256K and 1Meg boards. The 1 Meg boards are only £100 and are available from us. Just think, you could be running your Supercalc or Neword at 3 to 4 times the speed you are used to.

Two other goodies on offer from UK Home Computers are MTX 512's for £43 and RS128's for £68. Please contact Ron directly.

All 'SUPER CHEAPIES' will be despatched by return of post.

!!! SUPER CHEAPIES !!!
(ONLY FROM STOCK)

DESC	QTY	PRICE (Each)	DESC	QTY	PRICE (Each)
RETURN TO EDEN	1	£7.00	THE ZOO	3	£4.50
EMERALD ISLE	1	£7.00	COBRA	1	£4.50
BLOBBO	5	£4.50	MINER DICK	1	£4.50
KILOPEDE	3	£4.50	HELI-MATHS	2	£4.00
REVERSI	1	£4.50			
MINEFIELD	3	£4.50			
FIRE HOUSE FREDD	2	£4.00			
TOADO	4	£3.50			
NEMO	1	£4.50	PONTOON & B' JACK	3	£4.50
SNAPPO	3	£4.50			
PAYROLL	1	£10.00			
PURCHASE LEDGER	1	£7.00			
PHYSICS 1	3	£5.50			
MATHS 1	1	£5.50	TAPEWORM	1	£4.50

Software prices for the best and most popular software:-

Zarkos	£7.00	Chamberoids	£7.00
Qogo2	£7.00	26*26 SpreadSht	£8.50
Karate King	£7.00	Son Of Pete	£7.00
S.M.G	£7.00	T.Snooker	£8.00 512 Only
Doodlebugs	£6.00	Super Bike	£6.00
J.J.Flash	£6.00	Ed/Asm	£8.50
Cee-5	£7.00	MTX Asm Lang Cse	£7.00
Highway Encounter	£8.50		

Order a copy today!!!!!!

Remember PRINTER RIBBONS are only £7.00
REVEAL is only £6.00
SMG II is only £6.00
PUC-MAN is only £5.00

The club has a special purchase offer, on The MTX ASSEMBLER LANGUAGE COURSE by John Grayson. We can supply this program at the NEW incredibly low price of Only £7.00.

Don't forget to order your copy of Fruit Machine from Graysoft today!!!!

EQM0032
SORTING IT OUT
PART II - BY PHIL EYRES

Last month I wrote an article which demonstrated the use of a very simple sort; the Bubble sort. I am sure many people have tried this routine before and found that it is fun to use, but in real life it is not very functional, because of the huge time overhead incurred when sorting more than just a few numbers. This month I have a new routine for you to try out, it works on a completely different principle, but is none the less understandable in its logic, if you have the time to fathom it all out. The routine is interesting because it is very fast at sorting large amounts of data.

```
10 LET Z=0
20 DIM A(100),L(50),H(50),IH(50)
30 FOR N=10 TO 100 STEP 10
40 FOR I=1 TO N
50 LET A(I)=INT(RND*500)
55 PRINT A(I);
60 NEXT I
65 PRINT
70 IL=1: IH(1)=N
80 LET Z=Z+1
85 PRINT "...SORTING";N;" NUMBERS"
90 CLOCK "000000":GOSUB 1000
100 PRINT : PRINT "SORTED ... ";:FOR I=1 TO N: PRINT A(I);:NEXT
I:PRINT
110 PRINT: PRINT TIME$;" - TIME FOR ";N;"NUMBERS":PRINT
150 NEXT N
500 STOP
1000 IF IH(Z)<=IL THEN LET Z=Z-1: GOTO 1080
1010 IF IH(Z)>(IL+1) THEN LET X=INT(RND*(IH(Z)-IL)+IL):LET
T=A(X):LET A(X)=A(IL): LET A(IL)=T
1020 LET K=A(IL):LET L(Z)=IL:LET H(Z)=IH(Z)
1030 IF L(Z)=H(Z) THEN GOTO 1070
1040 LET Q=A(H(Z)):IF K<=Q THEN GOTO 2000 ELSE LET
A(L(Z))=A(H(Z)): LET A(H(Z))=K: LET L(Z)=L(Z)+1
1050 IF L(Z)=H(Z) THEN GOTO 1070
1060 IF K>=A(L(Z)) THEN LET L(Z)=L(Z)+1: GOTO 1050 ELSE LET
A(H(Z))=A(L(Z)): LET A(L(Z))=K: LET H(Z)=H(Z)-1: GOTO 1030
1070 LET Z=Z+1:LET IH(Z)=L(Z-1)-1:GOTO 1000
1080 IF Z=0 THEN GOTO 1090
1082 LET IL=H(Z)+1:GOTO 1000
1090 RETURN
2000 LET H(Z)=H(Z)-1: GOTO 1030
```

Over the page is a graph(Figure 2) which shows the time/number relationship for the two types of sort. It is also interesting to note that the Bubble sort time/number relationship conforms to the N^2 algorithm (where N is the number of numbers to be sorted), where as the fast sort conforms to $N \log(N)$. To demonstrate this fact Figure 3 over the page is a graph showing the curves created by these algorithms. It is obvious that these

Continued On Page 9

COMPARISON OF SORT LAWS

Figure 3

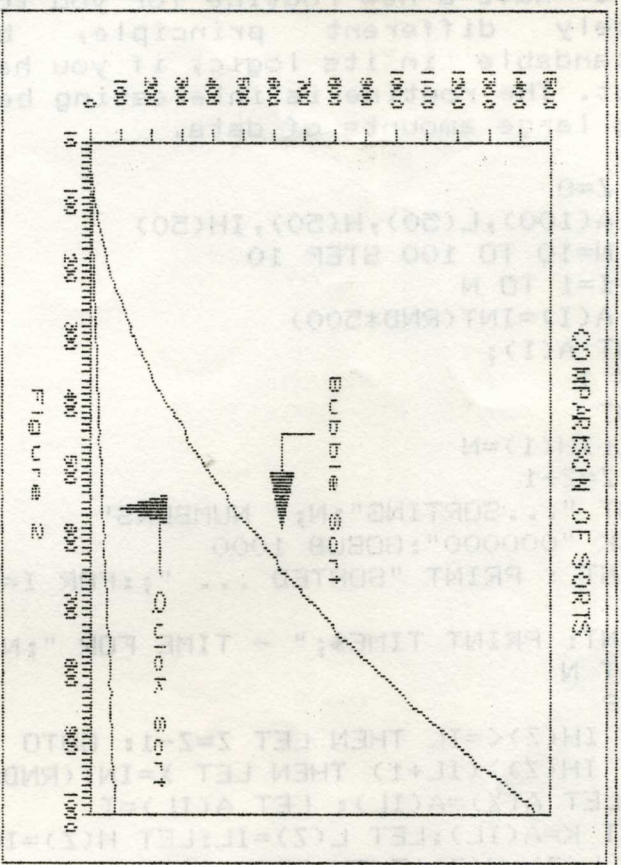
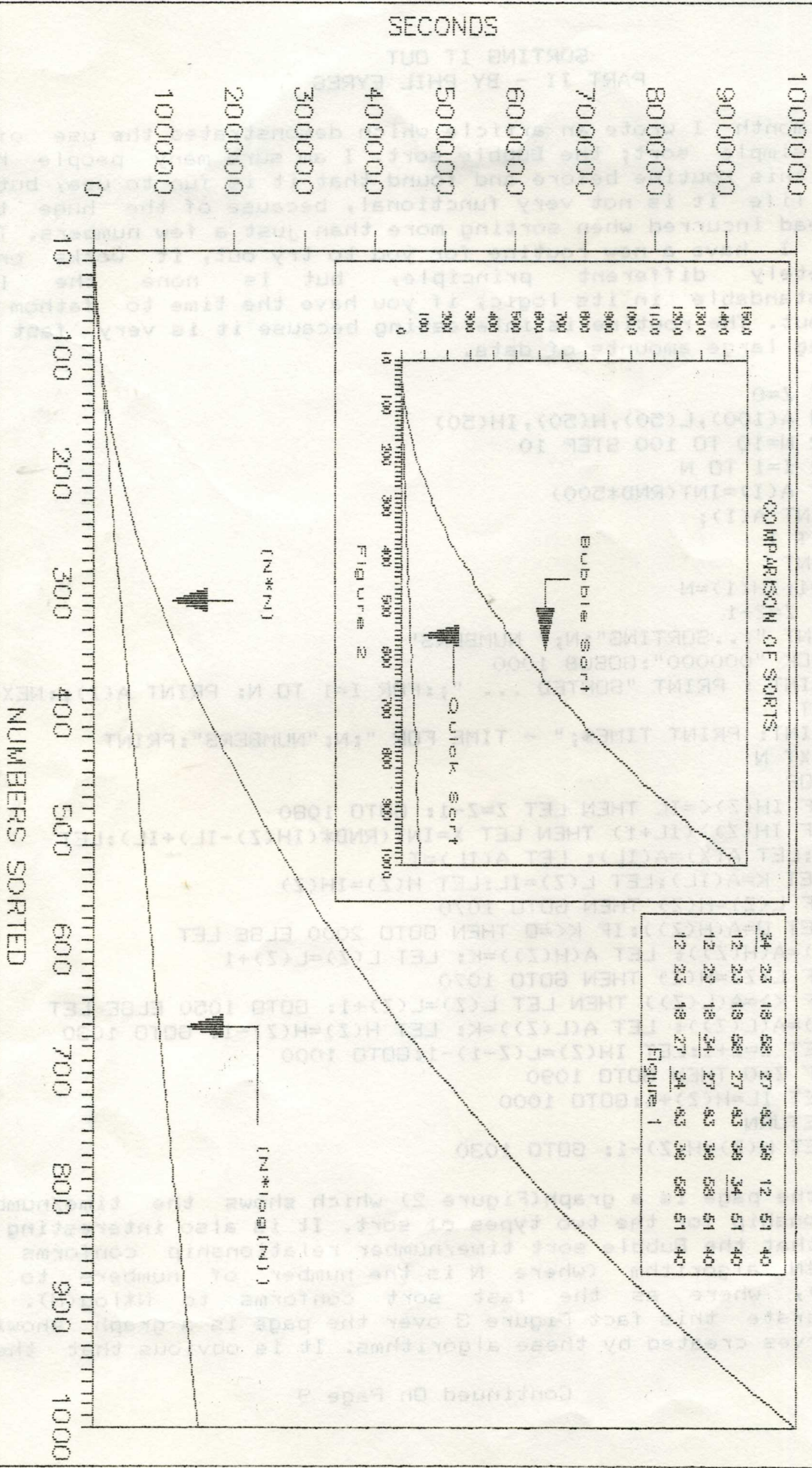


Figure 2

Figure 1

34	23	18	58	27	43	26	12	51	40
12	23	18	58	27	43	26	34	51	40
12	23	18	24	27	43	26	58	51	40
12	23	18	27	34	43	26	58	51	40

laws apply as the graph types are so similar. It should be noted that although the $N \log(N)$ looks straight, it is not, it in fact curves very slightly.

How does the Quick sort work?. Consider the figures on the first line of figure 1.

The leftmost integer (34) is under-lined and is referred to as the 'key' item. The key is compared with each number in turn working in from the right-hand side until an item is found that is smaller than the key. Thus we pass over 40 and 51 to reach 12, which is smaller than 34.

These items are swapped over, as shown in the second line. The key - still 34 - is compared with each item in return, this time working in from the left hand side, until an item is found that is larger than the key, and these are then swapped. The third line shows the result of swapping 34 and 58.

We continue in this manner alternately looking for items smaller than the key to the right and larger than the key to the left - each time starting from where we left off - until we have compared the key with all the items. The final result is shown in the bottom line.

The important thing to notice about this method is that the key has now divided the list into two sets, all the items to the right of the key are larger than it and all items to the left are smaller than it.

The key is thus in its final sorted position and need be considered no further. Also the two sets of integers either side of the key can now be sorted independently of each other.

This is a very substantial gain, and as the partitioning process is so successful, we can continue using it to sort the remaining numbers until the sorted sequence is achieved.

You may be wondering why there is a random number generated within a program used for a logical sort?.

Well, the idea is that, the quick sort is not very good at sorting numbers that are already in sequence, and so it is best to randomly jumble the sequence a bit to give the program something to do, which is quicker!!. Try generating a set of numbers in sequence and sorting them without the random bit.

I hope that everyone finds this routine useful and will find a place to keep it as I think it is most interesting, especially if you are into something like trying to predict the football pools. (If anyone wins using this? any chance of a 10% commission?)

ASSEMBLY LANGUAGE PROGRAMMING WITH CP/M

by Dave Dulson.

The CP/M operating system was created for use with the Intel family of processors, the 8080 and 8085. Therefore the programs supplied with the CP/M system for assembly programming only work with Intel mnemonic operation codes and not Z80 mnemonic's. But because the Z80 and the 8080 use the same machine codes the CP/M operating system will work with both processors.

Below is a conversion table from 8080 mnemonic's to Z80 mnemonic's this will help people get started in assembly language programming who do not have a CP/M Z80 assembler, however there is a very good Z80 assembler available through the club free of charge, just contact Phil for more details.

8080 mnemonic	object code	Z80 mnemonic		8080 mnemonic	object code	Z80 mnemonic
------------------	----------------	-----------------	--	------------------	----------------	-----------------

DATA TRANSFER GROUP

LDA	3A	LD A,(nn)		LDA B	0A	LD A,(BC)
LDAX D	1A	LD A,(DE)		LHLD	2A	LD HL,(nn)
LXI B	01	LD BC,nn		LXI D	11	LD DE,nn
LXI H	21	LD HL,nn		LXI SP	31	LD SP,nn
MOV M,r	>70	LD (HL),r		MOV r,M	>46	LD r,(HL)
MOV r,r	>40	LD r,r		MVI M	36	LD (HL),n
MVI r	>06	LD r,n		SHLD	22	LD (nn),HL
STA	32	LD (nn),A		STAX B	02	LD (BC),A
STAX D	12	LD (DE),A		XCHG	E8	EX DE,HL

8080 mnemonic	object code	Z80 mnemonic		8080 mnemonic	object code	Z80 mnemonic
------------------	----------------	-----------------	--	------------------	----------------	-----------------

ARITHMETIC GROUP

ACI	CE	ADC A,n		ADC M	8E	ADC A,(HL)
ADC r	>88	ADC A,r		ADD M	86	ADD A,(HL)
ADD r	>80	ADD A,r		ADI	C6	ADD A,n
DAA	27	DAA		DAD B	09	ADD HL,BC
DAD D	19	ADD HL,DE		DAD H	29	ADD HL,HL
DAD SP	39	ADD HL,SP		DCR M	35	DEC (HL)
DCR r	>05	DEC r		DCX B	0B	DEC BC
DCX D	1B	DEC DE		DCX H	2B	DEC HL
DCX SP	3B	DEC SP		INR M	34	INC (HL)
INR r	>04	INC r		INX B	03	INC BC
INX D	13	INC DE		INX H	23	INC HL
INX SP	33	INC SP		SBB M	9E	SBC A,(HL)
SBB r	>98	SBC A,r		SBI	DE	SBC A,n
SUB M	96	SUB (HL)		SUB r	>90	SUB r
SUI	D6	SUB n				

8080 mnemonic	object code	Z80 mnemonic		8080 mnemonic	object code	Z80 mnemonic
------------------	----------------	-----------------	--	------------------	----------------	-----------------

Continued From Page 10

LOGICAL GROUP

ANA M	A6	AND (HL)	ANA r	>A0	AND r
ANI	E6	AND n	CMA	2F	CPL
CMC	3F	CCF	CMP	BE	CP (HL)
CMP	>B8	CP r	CPI	FE	CP n
ORA M	B6	OR (HL)	ORA r	>B0	OR r
ORI	F6	OR n	RAL	17	RLA
RAR	1F	RRA	RLC	07	RLCA
RRC	0F	RRCA	STC	37	SCF
XRA M	AE	XOR (HL)	XRA r	>AB	XOR r
XRI	EE	XOR n			

8080 mnemonic	object code	Z80 mnemonic	:	8080 mnemonic	object code	Z80 mnemonic
------------------	----------------	-----------------	---	------------------	----------------	-----------------

STACK I/O & CONTROL GROUP

DI	F3	DI	EI	FB	EI
HLT	76	HALT	IN	DB	IN A, (n)
NOP	00	NOP	OUT	D3	OUT (n), A
POP B	C1	POP BC	POP D	D1	POP DE
POP H	E1	POP HL	POP PSW	F1	POP AF
PUSH B	C5	PUSH BC	PUSH D	D5	PUSH DE
PUSH H	E5	PUSH HL	PUSH PSW	F5	PUSH AF
SPHL	F9	LD SP, HL	XTHL	E3	EX (SP), HL

8080 mnemonic	object code	Z80 mnemonic	:	8080 mnemonic	object code	Z80 mnemonic
------------------	----------------	-----------------	---	------------------	----------------	-----------------

BRANCH GROUP

CALL	CD	CALL nn	CC	DC	CALL C, nn
CM	FC	CALL M, nn	CNC	D4	CALL NC, nn
CNZ	C4	CALL NZ, nn	CP	F4	CALL P, nn
CPE	EC	CALL PE, nn	CPO	E4	CALL PO, nn
CZ	CC	CALL Z, nn	JC	DA	JP C, nn
JM	FA	JP M, nn	JMP	C3	JP nn
JNC	D2	JP NC, nn	JNZ	C2	JP NZ, nn
JP	F2	JP P, nn	JPE	EA	JP PE, nn
JPO	E2	JP PO, nn	JZ	CA	JP Z, nn
PCHL	E9	JP (HL)	RC	D8	RET C
RET	C9	RET	RM	F8	RET M
RNC	D0	RET NC	RNZ	C0	RET NZ
RP	F0	RET P	RPE	E8	RET PE
RPO	E0	RET PO	RST	>C7	RST n
RZ	C8	RET Z			

The letter 'n' represents a numeric value between 0 & F and the letters 'nn' the numeric value between 0 & FF.

The letter 'r' represents any of the registers A, B, C, D, E, H, L.

Continued from page 11

The character '>' in front of some object codes denotes that these object codes are variable dependent upon the operation to be carried out. The files supplied on the CP/M system disc for assembly language programming are ED.COM, ASM.COM, DDT.COM and LOAD.COM. The file ED.COM short for EDITOR is not required has Newword does the job a lot better. The file ASM.COM is the Intel 8080 assembler this can be replaced with a Z80 assembler such as ZSM.COM or M80.COM if desired. The file DDT.COM short for Dynamic Debugging Tool, which operates on 8080 mnemonic's is a more powerful version of the front panel found on the MTX, the Z80 equivalent of this is ZSID.COM which stands for Z80 symbolic instruction debugger. The operation of both are very similar. The final file LOAD.COM works with both Z80 and 8080 hex files to produce COM files, these are the files you can run as programs under CP/M.

...More Next Month

REVIEWS... REVIEWS... REVIEWS... REVIEWS... REVIEWS...

Title: MURDER AT MANOR AUTHOR:
Machine: MTX
From: SYNTAX
Price: £8.20 Review by Sean Newman

Title: GHOSTLY CASTLE AUTHOR:
Machine: MTX
From: SYNTAX
Price: £3.80 Review by Sean Newman

This is a text adventure, and as far as I can see it will work on any size machine.

This is also a text adventure, and will also work on any size machine. The object of this adventure is to move around the castle grounds and buildings collecting treasure. The only real twist, to this standard adventure is that ghosts and ghouls follow you randomly around the castle. If these nasties catch up with you, you end up dead, and I haven't yet found something that will fend them off, if indeed there is anything!

It is based on that classic board game 'Cluedo'. Most of the adventure is staged in and around a large mansion. The object is to find the person who murdered Count Dresizird, who lays dead in his dining room. You must go around the house and gardens collecting clues, and try and relate where you found the clues to 'who dunnit'. Members of staff appear randomly around the house each time you play, and they tell you short statements which should help you solve the mystery.

All in all a rather disappointing adventure, written exclusively in Basic it is very slow at acting on your commands. It is a bit limited on the vocabulary you can use, and the location descriptions are very short. It is a bit short of things that you can examine, and when you do find something, it doesn't go mad on the length of the description of that either.

The screen layout is quite nice, it is divided up into two halves. The top half of the screen is for the statements like, where are you?, and what can be seen?. The bottom half of the screen is where all of the input is taken. It will accept most words and abbreviations that you would expect of an adventure, and so far I have not had any problems in making it understand what I am trying to say.

It is fairly cheap compared to most adventures, and its simplicity shows this. It is not to bad for the money, but does not have much lasting appeal.

For the size of memory that this adventure fits into, it is very good, It has had me playing it a lot since I got it, and I think that if you like adventures it is well worth a look.

PROGRAM LIBRARY
£1.20 Per Cassette, 2 Programs per Cassette

This month we have two new programs, which I hope will be of use to you. Hopefully from next month Alan Hamilton will be taking over the running of the Program Library. We will publish the address and give more details as we sort them out. In the mean time continue sending everything to Copse Road. We appear to have found the reason for me not being able to read/write 1Meg disc's formatted to 500K, apparently, members with 1Meg machines formatting down to 500K, format the wrong half of the disc, and consequently I am unable to read it. As this is the case, the cheapest solution is for the club to provide the disc. This will add about 80p to the fee for copying.

1. Basic & Assembler Programs

All programs available on cassette, 2 programs per cassette, £1.20 per cassette. Or on disc, £2.50 per disc, please enclose a disc, stating capacity. (Some programs are only available on cassette!!).

---- Diskette Three ----

62.Account	The Third Money Manager
63.Mastermind	Another Good Game
64.Connect4	Two Player Game
65.Journey Into Danger	Adventure Game
66.Connect4 Version 2	As for number 64
67.MTX DRAWv1.B	As for number 6
68.Patience	Card Game
69.Life	Curious Puzzle
70.Enigma	Like Mastermind
71.FKEY	Function Key Definer
72.Skydiver	Graphical Game
73.Digger	Graphical Game
74.MPG	Calculates MPG
75.FIG-FORTH (Cass)	RAM Disc
76.Optics/Colours	Educational Programs
77.Elements	Educational Programs
78.Dbase3 (Cass)	Database

---Diskette Four---

79.MKBOOK (Cass)	School Application
80.Settler	Yankee bet calculator

3. Articles From Previous Magazines

(Available as listings, please provide sufficient postage to cover club costs. TA!)

- 1.PANEL2 Utility. An updated version of PANEL1, which includes 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!!)
- 8.SDX Disc Review.

The item number 7, System Variables, I only have as a paper copy at present, if I had it as a Newword disc copy, I could update it as I receive information. The only trouble is its 10 pages long!. Any offers to type it in??.

5. Reviews

79.MKBOOK - By Allan Ayre

This program may be of any use to some who has to keep records of tests marks for pupils. The program utilises Eric Roy's RamDisc program available from the library.

The aim was to write a program that would store the test marks and the total mark for all the 4th Year pupils at the school taking GCSE Chemistry. The program had to be flexible to cope with such things as changes of pupils between sets, not all the marks being available at one time, pupils leaving, new pupils arriving etc. Also, the information had to be printed out in alphabetical order and by sets in rank order.

After a great deal of time and frustration, the program has reached its present form. Allowing for the limitations of a 40 column display, the program fulfils the aims set and easy to use. Error traps were put in where possible and is very trouble free (famous last words!). It was certainly a good exercise in string handling techniques.

This program has full help screens and makes extensive use of the ram disc program. It may be of use as a basis to any sort of database program. All in, a nice friendly program.

80.Settler - By Bob Bazley

This is an interesting program that will be of use to anyone who has the occasional 'flutter' on the 'gee-gees'.

The program will calculate the returns on your Yankee bets, for from 2 to 5 winners. The program shows returns for Doubles, Trebles, Fourfolds etc.

Unfortunately, it does not pick the winners for you!!