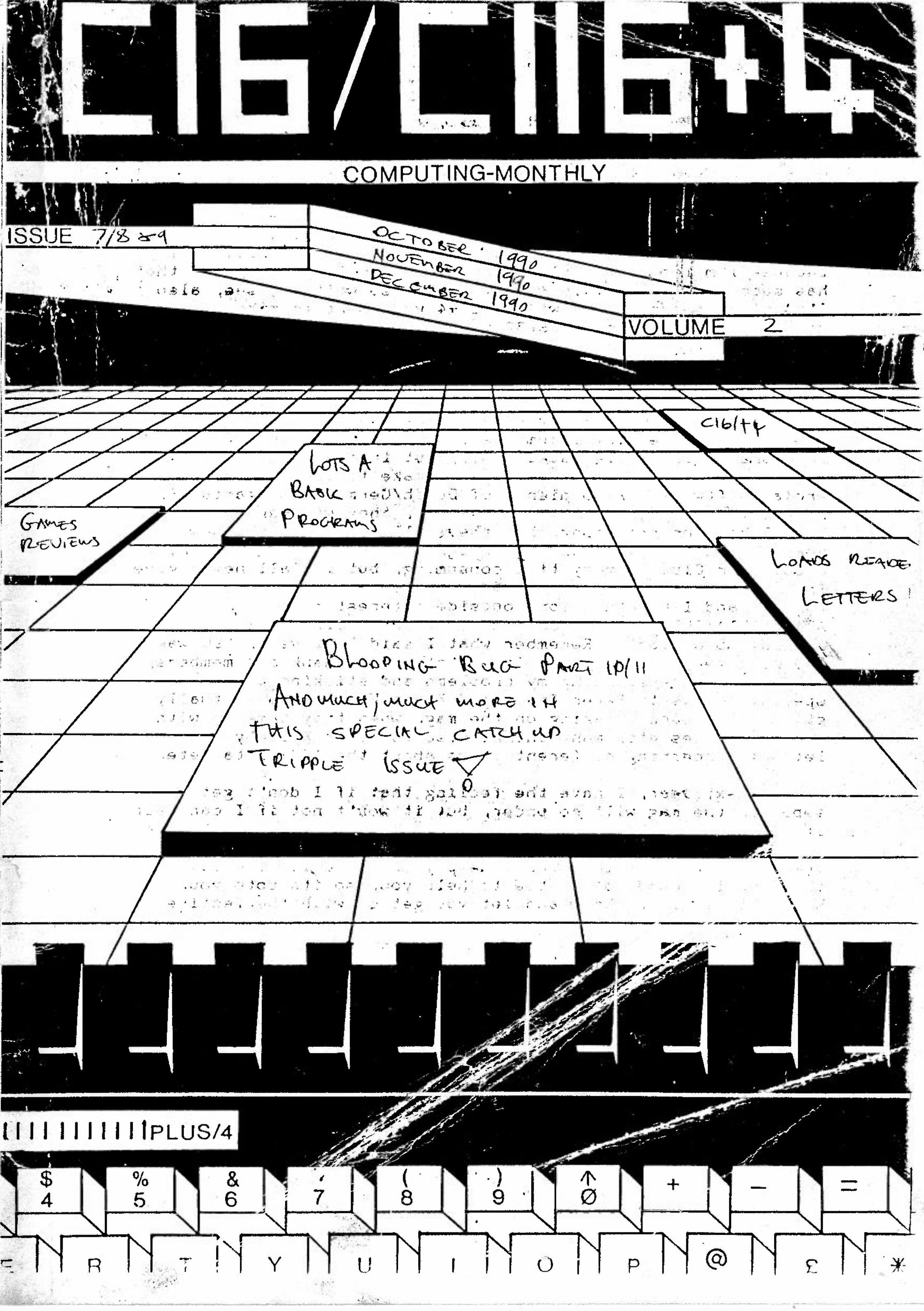
C16/C116+4

COMPUTING-MONTHLY

ISSUE 7/8 & 9

OCTOBER 1990
NOVEMBER 1990
DECEMBER 1990

VOLUME 2



LOTS A BASIC PROGRAMS

GAMES REVIEWS

LOADS READE LETTERS!

BLOOPING BUG PART 10/11

AND MUCH, MUCH MORE IN THIS SPECIAL CATCH UP TRIPPLE ISSUE

Eds Bits

Well whats this then I hear you say, a tripple issue, and maybe it will get to to before Xmas, hopefully, well you should get it in December at least. Well I managed to catch up anyway, thats one big boost for me, and the club, but I promise that 1991 will be a good year, a new era shall we say, because I'm looking at plans from a German Computer Freak that has successfully made is +4 into 256K, so wait and see, also I'm going to experiment with the +4 user port to make a very basic robot arm, also a RS232 port for C16 owners, which is at present be translated, yes it means that C16 owners will also be able to have ago at the expansion projects, also a mouse for the +4, which has been causing trouble, but Eric Jones has given lots of kind help, many thanks Eric for your time, also next issue 2 nice progs by Peter Crack that will be using the user port for reading documents from disk and printing them at the same time, completcated stuff, but interesting, also Plans to build a light pen for members to make to use with the trojan software. Also plenty of Dutch/German & Hungarian PD S/W up for grabs and the book library, they will arrive, I know I keep on promising, but they will appear in the new year. Being a COBOL computer programmer as a job and running a Computer Club is very time consuming, but I shall never give in, because at present I am the only C16/+4 club there is in the UK, and I have had some outside interest from AUSTRAILIA/GERMANY even one letter from RUSSIA, things are looking up at last. Remember what I said last year, '90 was the good year, well I know it was'nt and I thank all members for having understanding my problems and sticking by me, and a special thanks to Peter Crack and Keven Wheals who vertually stopped the back breaking on the mag, when they came up with progs articles etc, many thanks also to Peter Appleby for his letters concerning different items about the 16/+4, ta Pete.

Right, next year, I have the feeling that if I don't get any support, the mag will go under, but it won't not if I can help it. So please all members no matter what you talents etc are send in any articles, phone me up, pick my brains, ask questions, because if I can't help you, there is somebody in the club is bound to be able to help you, so its upto you. Well I'll stop for know and let you get on with the festive season, goodbye!!!!

All the best

Roy Robinson (ED)

P.S Have a happy and save Xmas & New Year, make '91 a year for changes.

```
1 REM *********************
2 REM * ANIMATED SKELETON *
3 REM * HI-RES PICTURE    *
4 REM *********************
5 REM * BY KEVIN WHEALS   *
6 REM *                   *
7 REM *    (C) 1990       *
8 REM *********************
10 PRINT"(CLR)":TT$="PLEASE WAIT"
20 LN=LEN(TT$):TB=20-LN/2:PRINTTAB(TB)TT$
30 POKE55,0:POKE56,60:CLR:POKE1177,62
40 FORI=0TO1023:POKE15360+I,PEEK(53248+I):NEXTI
50 POKE1177,63:POKE65299,60:POKE65298,192:SCNCLR
60 FORL=0TO56:CX=0:FORD=0TO7:READCH:CX=CX+CH:POKE15640+L*8+D,CH:NEXTD
70 READCH:IFCH<>CXTHENPOKE65299,208:POKE65298,196:PRINT"ERROR IN LINE";1000+(L*1
0):STOP
80 NEXTL
90 COLOR0,1:COLOR4,1
100 A=16
110 COLOR1,A
120 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
130 PRINTTAB(14);:PRINT"#$%&'("
140 PRINTTAB(13);:PRINT")*+,[SPACE]-."
150 PRINTTAB(13);:PRINT"/0[3 SPACES]12345"
160 PRINTTAB(14);:PRINT"6789:;<[SPACE]=>"
170 PRINTTAB(15);:PRINT"?[SFT *],[SPACE][SFT A][SFT B][SFT C][SPACE][SFT D][SFT
E]"
180 PRINTTAB(16);:PRINT"[SFT F][SFT G][SFT H][SFT I][SFT J][SFT K][SFT L]"
190 PRINTTAB(16);:PRINT"[SFT M][SFT N][SFT O][3 SPACES][SFT P][SFT Q][SFT R][SFT
 S]"
200 PRINTTAB(15);:PRINT"[SFT T][SFT U][SPACE][SFT V][6 SPACES][SFT W]"
210 A=A-1:IFA=1THENGOTO220:ELSEGOTO110
220 A=A+1:COLOR1,A:PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
230 PRINTTAB(14);:PRINT"#$%&'("
240 PRINTTAB(13);:PRINT")*+,[SPACE][SFT X][SFT Y]"
250 PRINTTAB(13);:PRINT"/0[3 SPACES][SFT Z][SFT +]345"
260 PRINTTAB(14);:PRINT"6789:;<[SPACE]=>"
270 PRINTTAB(15);:PRINT"?[SFT *],[SPACE][SFT A][SFT B][SFT C][SPACE][SFT D][SFT
E]"
280 PRINTTAB(16);:PRINT"[SFT F][SFT G][SFT H][SFT I][SFTJ][SFT K][SFT L]"
290 PRINTTAB(16);:PRINT"[SFT M][SFT N][SFT O][3 SPACES][SFT P][SFT Q][SFT R][SFT
 S]"
300 PRINTTAB(15);:PRINT"[SFT T][SFT U][SPACE][SFT V][6 SPACES][SFT W]"
310 IFA=16THENGOTO110:ELSEGOTO220
999 REM *** UDG DATA FOLLOWS ***
1000 DATA 000,000,000,000,000,001,003,015, 0019
1010 DATA 000,000,000,031,127,255,255,255, 0923
1020 DATA 000,000,127,255,255,255,255,252, 1399
1030 DATA 000,000,224,255,255,255,255,000, 1244
1040 DATA 000,000,000,000,248,254,255,015, 0772
1050 DATA 000,000,000,000,000,000,128,224, 0352
1060 DATA 000,000,000,000,001,003,007,015, 0026
1070 DATA 031,063,127,255,255,255,252,248, 1486
1080 DATA 255,252,240,192,128,000,000,000, 1067
1090 DATA 128,000,000,000,000,000,000,000, 0128
1100 DATA 000,000,003,007,015,031,007,007, 0070
1110 DATA 112,008,192,224,240,248,248,248, 1520
```

```
1120 DATA 007,003,001,000,000,000,000,000, 0011
1130 DATA 208,192,192,128,032,048,024,009, 0833
1140 DATA 015,025,002,006,012,000,000,000, 0060
1150 DATA 240,224,000,128,067,027,057,100, 0843
1160 DATA 000,000,000,000,000,255,128,000, 0383
1170 DATA 000,000,000,000,007,255,006,000, 0268
1180 DATA 000,000,000,000,000,096,224,112, 0432
1190 DATA 001,012,001,000,000,000,000,000, 0014
1200 DATA 000,096,120,014,065,048,024,012, 0379
1210 DATA 000,000,000,000,128,124,013,005, 0270
1220 DATA 000,000,000,000,003,222,240,192, 0657
1230 DATA 006,007,031,248,192,000,000,000, 0484
1240 DATA 222,063,079,079,073,073,073,041, 0703
1250 DATA 000,000,192,192,224,112,120,124, 0964
1260 DATA 024,012,003,001,000,000,000,000, 0040
1270 DATA 000,000,000,128,192,096,048,014, 0478
1280 DATA 006,003,003,001,000,000,000,000, 0013
1290 DATA 001,000,000,128,192,096,048,024, 0489
1300 DATA 001,000,000,000,000,000,000,000, 0001
1310 DATA 092,070,016,003,003,001,001,001, 0187
1320 DATA 000,000,192,192,224,176,248,240, 1272
1330 DATA 014,006,000,000,000,000,000,000, 0020
1340 DATA 000,248,192,000,000,000,000,000, 0440
1350 DATA 012,006,003,001,000,000,000,000, 0022
1360 DATA 000,000,000,128,192,096,048,049, 0513
1370 DATA 000,000,011,003,027,056,096,128, 0321
1380 DATA 000,000,255,192,000,000,000,000, 0447
1390 DATA 013,061,236,000,000,000,000,000, 0310
1400 DATA 238,143,007,001,000,000,000,000, 0389
1410 DATA 000,000,128,192,240,056,014,007, 0637
1420 DATA 000,000,000,000,000,003,001,006, 0010
1430 DATA 027,012,006,059,225,192,128,000, 0649
1440 DATA 000,000,000,000,128,192,096,048, 0464
1450 DATA 003,003,000,000,000,000,000,000, 0006
1460 DATA 128,056,063,129,000,000,000,000, 0376
1470 DATA 000,000,192,248,031,001,000,000, 0472
1480 DATA 000,000,000,000,000,248,056,026, 0330
1490 DATA 000,001,003,000,000,000,000,000, 0004
1500 DATA 030,220,000,000,000,000,000,000, 0250
1510 DATA 024,012,004,000,000,000,000,000, 0040
1520 DATA 006,004,012,008,112,000,000,000, 0142
1530 DATA 000,000,003,007,015,025,025,025, 0100
1540 DATA 112,008,192,224,240,152,152,152, 1232
1550 DATA 014,007,005,004,006,002,001,000, 0039
1560 DATA 112,224,160,032,099,091,185,100, 1003
1570 REM *** END OF DATA ***
```

```
1 REM *******************
2 REM * BOEING 737      *
3 REM * HI-RES PICTURE  *
4 REM *******************
5 REM * BY KEVIN WHEALS *
6 REM *                 *
7 REM *   (C) 1990      *
8 REM *******************
10 PRINT"(CLR)":TT$="NOW ARRIVING AT TERMINAL 4"
20 LN=LEN(TT$):TB=20-LN/2:PRINTTAB(TB)TT$
30 POKE55,0:POKE56,60:CLR:POKE1177,62
40 FORI=0TO1023:POKE15360+I,PEEK(53248+I):NEXTI
50 POKE1177,63:POKE65299,60:POKE65298,192:SCNCLR
60 FORL=0TO126:CX=0:FORD=0TO7:READCH:CX=CX+CH:POKE15360+L*8+D,CH:NEXTD
70 READCH:IFCH<>CXTHENPOKE65299,208:POKE65298,196:PRINT"ERROR IN LINE";1000+(L*1
0):STOP
80 NEXTL
90 COLOR0,2:COLOR4,2:COLOR1,1
100 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
110 PRINTTAB(36);:PRINT"@A"
120 PRINTTAB(35);:PRINT"BCD"
130 PRINTTAB(34);:PRINT"BEFG"



140 PRINTTAB(33);:PRINT"B[SPACE]HIJ"
150 PRINTTAB(32);:PRINT"B[2 SPACES]KLM"
160 PRINTTAB(31);:PRINT"B[2 SPACES]NOP"
170 PRINTTAB(30);:PRINT"B[3 SPACES]QRS"
180 PRINTTAB(14);:PRINT"T[SPACE]U[11 SPACES]HB[4 SPACES]V[SPACE]WXYZ"
190 PRINTTAB(3);:PRINT"[£]↑←←←←←←←←←←←←←←←←←←←←←←!↑←←←[CBM V]#$[SPACE]%M"
200 PRINTTAB(1);:PRINT"&'()S*[SPACE]+,-./+,-01-./+,-./+,-S*[3 SPACES]23456"
210 PRINT"789:[SPACE]SL[7 SPACES]XX;<XX=>?[SFT *][SFT A][SFT B][SFT C][SFT D][SF
T E]S[SFT F][2 SPACES][SFT G][SFT H][SFT I][SFT J]"
220 PRINT"[SFT K][SFT L]YYY[SFT M][SFT N]YYYYY[SFT O][SFT P][SFT Q][SFT R][SFT S
][SFT T][SFT U][SFT V][SFTW][SFT X][SFT Y][SFT Z]XXXXX[SFT +][CBM -][SFT -][CBM
=][CBM *]M"
230 PRINTTAB(1);:PRINT"[SFT SPACE][CBM K][CBM I]YXXXXXXX[CBM T][CBM @][CBM G][CB
M +][CBM M][CBM £][SFT £][CBM N][CBM Q][CBM D][CBM Z][CBM S]XXX[CBM P][CBM A][CB
M E]←"
240 PRINTTAB(8);:PRINT"[CBM R][3 SPACES][CBM W][SPACE][CBM H][CBM J][CBM L][CBM
Y][CBM U][CBM O][SFT @][CBM F][CBM C][CBM X]"



300 GOTO300
999 REM *** UDG DATA FOLLOWS ***
1000 DATA 000,000,007,008,016,039,072,136, 0278
1010 DATA 000,000,252,002,002,254,004,004, 0518
1020 DATA 001,002,004,008,016,032,064,128, 0255
1030 DATA 008,006,001,002,002,002,004,004, 0029
1040 DATA 004,004,008,008,008,008,008,008, 0056
1050 DATA 000,000,000,001,001,001,000,000, 0003
1060 DATA 004,008,008,240,000,240,016,016, 0532
1070 DATA 016,016,016,016,032,032,032,032, 0192
1080 DATA 000,000,000,000,000,000,000,001, 0001
1090 DATA 032,032,032,064,064,128,128,000, 0480
```

```
1100 DATA 032,032,064,064,064,064,128,128, 0576
1110 DATA 127,064,126,002,002,004,004,004, 0333
1120 DATA 001,001,001,001,001,001,001,001, 0008
1130 DATA 128,000,000,000,000,000,000,000, 0128
1140 DATA 000,000,000,000,000,000,031,016, 0047
1150 DATA 004,008,008,008,016,016,224,000, 0284
1160 DATA 002,002,002,002,002,004,004,004, 0022
1170 DATA 031,000,000,000,000,000,001,001, 0033
1180 DATA 224,032,064,064,128,128,000,000, 0640
1190 DATA 004,004,004,004,004,004,004,004, 0032
1200 DATA 000,000,000,000,000,000,000,129, 0129
1210 DATA 000,000,000,004,012,028,028,060, 0132
1220 DATA 001,001,002,002,002,004,004,004, 0020
1230 DATA 008,008,016,016,032,032,032,064, 0208
1240 DATA 000,000,000,000,000,000,000,255, 0255
1250 DATA 000,000,000,000,000,000,255,000, 0255
1260 DATA 000,000,000,000,000,000,224,064, 0288
1270 DATA 000,000,000,000,000,001,014,112, 0127
1280 DATA 000,000,000,007,056,192,000,000, 0255
1290 DATA 000,007,056,192,000,000,003,004, 0262
1300 DATA 255,000,000,000,000,000,254,001, 0510
1310 DATA 255,000,000,000,000,000,000,000, 0255
1320 DATA 000,000,000,000,000,000,000,000, 0000
1330 DATA 255,000,000,000,000,000,003,004, 0262
1340 DATA 000,000,000,000,000,000,000,000, 0000
1350 DATA 255,000,003,124,128,000,000,000, 0510
1360 DATA 255,028,224,000,000,000,000,000, 0507
1370 DATA 000,001,002,004,248,016,016,032, 0319
1380 DATA 000,000,000,000,000,000,001,003, 0004
1390 DATA 001,002,014,048,127,204,152,016, 0564
1400 DATA 128,000,224,000,255,032,096,065, 0800
1410 DATA 000,000,000,000,192,064,064,192, 0512
1420 DATA 001,001,001,049,049,001,001,001, 0104
1430 DATA 000,000,000,000,115,115,115,115, 0460
1440 DATA 000,000,000,000,156,156,156,156, 0624
1450 DATA 000,000,000,000,231,231,231,231, 0924
1460 DATA 000,000,000,000,057,057,057,057, 0228
1470 DATA 000,000,000,000,206,206,206,206, 0824
1480 DATA 000,001,002,002,058,058,058,058, 0237
1490 DATA 000,252,002,002,114,114,114,114, 0712
1500 DATA 001,000,000,000,000,000,000,000, 0001
1510 DATA 128,064,063,016,016,008,008,004, 0307
1520 DATA 000,255,000,000,000,000,000,000, 0255
1530 DATA 000,255,000,000,001,006,024,224, 0510
1540 DATA 032,192,064,064,128,000,000,000, 0480
1550 DATA 000,000,000,001,006,024,032,096, 0159
1560 DATA 007,024,104,136,008,008,008,008, 0303
1570 DATA 248,007,000,000,000,000,000,000, 0255
1580 DATA 206,240,000,000,000,000,000,000, 0446
1590 DATA 002,002,002,001,000,000,000,255, 0262
1600 DATA 002,002,002,252,000,000,000,255, 0513
1610 DATA 000,000,000,000,000,007,248,000, 0255
1620 DATA 000,000,001,014,241,000,000,000, 0256
1630 DATA 000,000,255,034,255,000,000,003, 0547
1640 DATA 000,000,224,030,232,048,192,000, 0726
1650 DATA 000,000,000,007,005,006,005,007, 0030
1660 DATA 000,000,000,119,084,086,084,119, 0492
1670 DATA 000,000,000,090,090,086,086,082, 0434
1680 DATA 000,000,000,231,161,129,178,226, 0925
```

```
1690 DATA 000,000,000,119,017,049,018,114, 0317
1700 DATA 001,001,013,001,001,001,001,001, 0020
1710 DATA 000,000,000,000,000,000,001,030, 0031
1720 DATA 000,000,000,000,000,127,131,124, 0382
1730 DATA 012,020,036,069,158,096,128,000, 0519
1740 DATA 003,012,112,128,000,000,000,000, 0255
1750 DATA 224,224,224,096,032,024,006,001, 0831
1760 DATA 008,008,008,008,008,008,015,136, 0199
1770 DATA 004,003,000,003,002,002,254,003, 0271
1780 DATA 001,254,000,254,002,002,003,254, 0770
1790 DATA 000,000,000,000,001,002,252,004, 0259
1800 DATA 071,024,033,194,004,005,004,002, 0337
1810 DATA 000,063,192,000,192,064,192,000, 0703
1820 DATA 000,255,000,000,000,003,012,240, 0510
1830 DATA 000,255,000,001,030,228,002,001, 0517
1840 DATA 008,255,003,252,001,014,016,016, 0565
1850 DATA 000,255,224,000,255,000,000,000, 0734
1860 DATA 112,128,000,000,255,014,001,002, 0512
1870 DATA 000,255,062,001,254,004,255,000, 0831
1880 DATA 000,255,032,255,001,000,255,000, 0798
1890 DATA 004,248,007,255,129,064,248,128, 1083
1900 DATA 000,000,128,224,152,100,024,007, 0635
1910 DATA 004,003,000,000,001,062,192,000, 0262
1920 DATA 001,254,000,015,240,000,000,000, 0510
1930 DATA 000,000,031,224,000,001,030,224, 0510
1940 DATA 001,126,128,007,056,192,000,000, 0510
1950 DATA 225,014,112,128,000,000,000,000, 0479
1960 DATA 104,028,003,000,000,000,000,000, 0135
1970 DATA 000,000,128,112,015,000,000,000, 0255
1980 DATA 000,000,000,000,000,248,007,000, 0255
1990 DATA 004,004,004,004,002,001,000,255, 0274
2000 DATA 001,000,000,014,126,014,128,255, 0538
2010 DATA 131,125,005,005,005,005,005,253, 0534
2020 DATA 015,254,035,254,000,254,034,254, 1100
2030 DATA 255,000,255,000,000,000,014,010, 0534
2040 DATA 255,112,159,012,003,000,000,028, 0569
2050 DATA 255,000,255,000,000,192,032,024, 0758
2060 DATA 255,002,255,002,002,002,002,002, 0522
2070 DATA 255,000,255,000,000,064,000,000, 0574
2080 DATA 240,012,251,010,010,011,008,008, 0550
2090 DATA 127,032,240,159,146,146,017,017, 0884
2100 DATA 252,003,001,254,034,044,048,063, 0699
2110 DATA 000,000,000,000,001,030,224,000, 0255
2120 DATA 000,000,000,015,240,000,000,000, 0255
2130 DATA 000,007,056,192,000,000,000,000, 0255
2140 DATA 129,000,000,000,000,000,000,000, 0129
2150 DATA 142,007,003,000,000,000,000,000, 0152
2160 DATA 005,003,001,000,000,000,000,000, 0009
2170 DATA 000,062,194,060,003,000,000,000, 0319
2180 DATA 010,014,000,026,255,000,000,000, 0305
2190 DATA 020,020,028,128,255,000,000,000, 0451
2200 DATA 232,232,228,004,255,000,000,000, 0951
2210 DATA 002,002,002,031,224,000,000,000, 0261
2220 DATA 000,000,000,255,000,000,000,000, 0255
2230 DATA 008,008,008,255,000,000,000,000, 0279
2240 DATA 018,018,031,224,000,000,000,000, 0291
2250 DATA 036,060,224,000,000,000,000,000, 0320
2260 DATA 255,000,000,000,003,030,004,002, 0294
2270 REM *** END OF DATA ***
```

Quick Reviews of some Crap games

World Cup by ARTIC:

A two player football game. Pitch is a screen wide by a screen and a half long. Graphic bug when the pitch scrolls. Easy and thourghly uncontrollable.

Overall Rating: 64%

Diagon by BUG-BYTE:

A noisey space shoot-em-up with confusing graphics. Sprites too small and they blend in with the background. Shoot the hyper galactic aliens to win freedom. Zzzzzzz.

Overall Rating: 67%

Street Olympics by MASTERTRONIC:

Take part in several slow and joystick cripling events. If you spot the jerkily scrolling line you might just beat the record (highly unlikely!). After the joystick has broken in two or the 'Z' and 'X' key are just grains of plastic you'll find out no records have been broken.

Overall Rating: 56%

Gun Law by MASTERTRONIC:

Go about shootin the hell out of planet inhabitating aliens (who look suspiciously like you do) and free your planet once and for all. There are a few graphic bugs and some lousey sound effects. It could have been quite good but in the end it all looks at bit tacky.

Overall Rating: 71% (I'm feeling genourous today!)

Frensis by MASTERTRONIC (they're getting a bad time today)

I don't quite understand this. You have two lines and you have to try and kill various sprites before they get past a certain point. A bit loud and freeky sound effects but not too bad. But at least you get responces from this game.

Overall Rating: 75%

Matt.

```
2 REM ******************
3 REM * STREET ROD      *
4 REM * HI-RES PICTURE *
5 REM ******************
6 REM * BY KEVIN WHEALS*
7 REM *                *
8 REM * (C)1990        *
9 REM ******************
10 COLOR0,2:COLOR4,2:COLOR1,1
20 POKE55,0:POKE56,60:CLR:POKE1177,62
30 FORI=0TO1023:POKE15360+I,PEEK(53248+I):NEXTI
40 POKE1177,63:POKE65299,60:POKE65298,192:SCNCLR
50 FORL=0TO98:CX=0:FORD=0TO7:READCH:CX=CX+CH:POKE15360+L*8+D,CH:NEXTD
60 READCH:IFCH<>CXTHENPOKE65299,208:POKE65298,196:PRINT"ERROR IN LINE";1000+(L*1
0):STOP
70 NEXTL
100 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
110 PRINTTAB(12);:PRINT"@ABCD"
120 PRINTTAB(7);:PRINT"EFGGHIJKLMNOP"
130 PRINTTAB(6);:PRINT"QRSTUVWXYZX[£]↑"
140 PRINTTAB(6);:PRINT"←!#$%&  '(  )*+,"
150 PRINTTAB(6);:PRINT"-./012    ( 345678":REM USE NORMAL SPACES
160 PRINTTAB(6);:PRINT"9:;<=>?──♠─│──‾─│ │":REM USE SHIFT * * A * B C D E F G H
170 PRINTTAB(7);:PRINT"╮ ╰╯L╲╱┬┬┬●─●│ ╱╳"
171 REM USE SHIFT I J K L M N O P P P Q R S T U V
180 PRINTTAB(7);:PRINT"○♣ │♦┼┼▒ ││││π│◥ ▌▄":REM USE SHIFT W X Y Z + +  CBM -
185 REM SHIFT - - - -  CBM = SHIFT -  CBM *  SHIFTED SPACE  CBM K I
300 GOTO300
500 REM *** UDG DATA FOLLOWS ***
1000 DATA 000,000,003,004,009,008,019,039, 0082
1010 DATA 000,000,255,000,255,000,255,248, 1013
1020 DATA 000,000,255,000,240,014,245,010, 0764
1030 DATA 000,000,240,012,003,000,000,128, 0383
1040 DATA 000,000,000,000,000,128,064,032, 0224
1050 DATA 000,000,000,000,000,000,000,127, 0127
1060 DATA 000,000,000,000,000,000,000,245, 0245
1070 DATA 000,000,000,000,000,000,000,085, 0085
1080 DATA 000,000,000,000,001,001,002,252, 0256
1090 DATA 071,076,136,152,090,093,079,160, 0857
1100 DATA 000,000,000,000,000,000,255,000, 0255
1110 DATA 005,005,002,002,002,002,252,001, 0271
1120 DATA 064,064,160,160,160,144,143,192, 1087
1130 DATA 016,008,004,002,002,005,248,000, 0285
1140 DATA 000,000,000,000,000,000,128,127, 0255
1150 DATA 000,000,000,000,000,024,032,255, 0311
1160 DATA 000,000,000,000,000,000,000,192, 0192
1170 DATA 001,002,002,005,007,006,005,006, 0034
1180 DATA 160,064,128,031,224,177,080,176, 1040
1190 DATA 000,000,000,255,000,255,042,085, 0637
1200 DATA 000,000,000,255,000,255,170,085, 0765
1210 DATA 000,000,000,255,000,255,168,080, 0758
1220 DATA 016,016,009,242,018,186,018,018, 0523
1230 DATA 159,128,064,255,000,255,000,000, 0861
1240 DATA 255,000,000,255,000,255,000,000, 0765
1250 DATA 255,000,000,255,000,255,000,003, 0768
1260 DATA 255,000,000,255,000,255,000,000, 0765
1270 DATA 128,000,000,255,000,255,000,000, 0638
```

```
1280 DATA 000,000,000,255,000,255,000,000, 0510
1290 DATA 048,012,003,192,032,192,060,003, 0542
1300 DATA 000,000,000,192,048,008,004,194, 0446
1310 DATA 005,011,012,008,008,024,120,052, 0240
1320 DATA 000,000,000,000,000,000,000,000, 0000
1330 DATA 081,252,126,127,127,127,120,112, 1072
1340 DATA 000,000,000,000,000,000,000,000, 0000
1350 DATA 255,042,085,127,255,000,000,000, 0764
1360 DATA 255,170,085,255,192,056,004,002, 1019
1370 DATA 255,168,080,254,000,000,000,000, 0757
1380 DATA 018,018,018,018,018,018,018,018, 0144
1390 DATA 004,000,000,000,000,000,000,000, 0004
1400 DATA 032,032,032,032,032,032,032,032, 0256
1410 DATA 000,000,001,006,027,044,080,160, 0318
1420 DATA 000,000,255,255,000,000,000,000, 0510
1430 DATA 049,012,194,185,006,001,000,000, 0447
1440 DATA 000,128,112,040,020,140,072,036, 0548
1450 DATA 059,021,027,022,024,016,016,033, 0218
1460 DATA 192,128,000,003,015,031,120,247, 0736
1470 DATA 000,000,000,252,255,255,001,254, 1017
1480 DATA 001,000,000,000,000,192,224,248, 0665
1490 DATA 000,192,032,020,008,004,002,001, 0259
1500 DATA 018,018,018,018,018,018,018,034, 0160
1510 DATA 001,002,005,005,010,010,046,046, 0125
1520 DATA 064,128,006,025,055,111,222,189, 0800
1530 DATA 048,000,126,255,255,255,000,255, 1194
1540 DATA 000,000,000,128,224,240,120,188, 0900
1550 DATA 018,010,005,002,003,001,041,072, 0152
1560 DATA 000,096,160,032,192,064,064,160, 0768
1570 DATA 067,071,063,031,015,007,003,000, 0257
1580 DATA 239,220,185,115,103,111,111,111, 1195
1590 DATA 255,003,249,252,254,255,159,015, 1442
1600 DATA 124,191,223,239,121,121,121,121, 1261
1610 DATA 000,000,224,248,255,255,255,255, 1492
1620 DATA 194,034,017,015,000,255,255,255, 1025
1630 DATA 128,127,000,255,128,255,128,255, 1276
1640 DATA 000,255,000,255,000,255,000,255, 1020
1650 DATA 032,175,064,255,000,255,000,255, 1036
1660 DATA 071,175,015,207,032,255,031,255, 1041
1670 DATA 123,119,121,094,220,221,221,221, 1340
1680 DATA 255,255,000,126,255,255,231,195, 1572
1690 DATA 220,238,247,123,059,187,187,187, 1448
1700 DATA 124,060,044,158,192,255,192,255, 1280
1710 DATA 160,080,080,080,040,241,006,252, 0939
1720 DATA 000,000,000,000,000,224,000,000, 0224
1730 DATA 105,105,104,100,114,057,028,015, 0628
1740 DATA 105,153,241,002,004,249,003,255, 1012
1750 DATA 108,104,107,103,239,223,191,127, 1202
1760 DATA 000,000,252,255,255,255,255,255, 1527
1770 DATA 000,000,000,255,255,255,255,255, 1275
1780 DATA 000,000,000,129,255,255,255,255, 1149
1790 DATA 000,000,003,255,255,255,255,255, 1278
1800 DATA 000,000,255,255,255,255,255,255, 1530
1810 DATA 001,001,192,254,254,255,255,255, 1467
1820 DATA 221,221,221,220,222,121,183,203, 1612
1830 DATA 090,101,060,129,126,000,255,255, 1016
1840 DATA 184,184,184,056,120,243,231,207, 1409
1850 DATA 000,000,000,000,000,248,255,255, 0758
1860 DATA 000,000,000,000,000,000,128,240, 0368
```

```
1870 DATA 007,000,000,000,000,000,000,000, 0007
1880 DATA 254,000,000,000,000,000,000,000, 0254
1890 DATA 255,031,000,000,000,000,000,000, 0286
1900 DATA 255,255,003,000,000,000,000,000, 0513
1910 DATA 255,255,255,000,000,000,000,000, 0765
1920 DATA 255,252,000,000,000,000,000,000, 0507
1930 DATA 255,000,000,000,000,000,000,000, 0255
1940 DATA 225,000,000,000,000,000,000,000, 0225
1950 DATA 159,000,000,000,000,000,000,000, 0159
1960 DATA 255,248,000,000,000,000,000,000, 0503
1970 DATA 252,254,063,000,000,000,000,000, 0569
1980 DATA 000,000,128,000,000,000,000,000, 0128
1990 REM *** END OF DATA ***
2000 REM
2010 REM *****************************
2020 REM * 1936 FORD 3 WINDOW COUPE *
2030 REM *****************************
```

For Sale:

COMMODORE PLUS/4 COMPUTER
COMMODORE GREEN SCREEN
COMMODORE DATASETTE
COMMODORE PLUS/4 PSU
2 x CHEETAH STAR PROBE JOYSTICKS (WITH ADAPTORS FOR BOTH)
80+ TOP SOFTWARE TITLES (ORIGINALS)
LOADS OF PD AND OTHER PROGRAMS

A bargain at only £124.99

Everything is in perfect working order with all leads and manuals necessary. 99% of all games have proper instructions.

If you would like to buy all/anything mentioned above, ask any questions, get a software list, etc. write to:

Matthew Newton-Lewis
24, Jubilee Road,
Chichester,
West Sussex,
PO19 1XJ.

Or ring: Chichester (0243) 782176 any day after 7pm.

Any offers are more than welcome, please ask.

Matt

Curve:

```
10 GRAPHIC 1,1
20 X=10: Y=10
30 IF X>185 THEN GOTO 70
40 DRAW,10,X TO X,185
50 X=X+3
60 GOTO 30
70 FOR L=1 TO 500: NEXT L
80 GOTO 10
```

Curves:

```
10 GRAPHIC 1,1
20 X=10: Y=10
30 IF X>185 THEN GOTO 70
40 DRAW,10,X TO X,185
50 X=X+3
60 GOTO 30
70 X=10: Y=10
80 IF X>185 THEN GOTO 120
90 DRAW,X,10 TO 185,X
100 X=X+3
110 GOTO 80
120 FOR L=1 TO 500: NEXT L
130 GOTO 10
```

Matt

Ovals

```
10 GRAPHIC 1,1
20 FOR L=150 TO 1 STEP -2
30 CIRCLE,160,100,L,50
40 NEXT L
50 FOR L=1 TO 2000: NEXT L
60 GRAPHIC 0
```

NOTE: NEW ADDRESS
Or visit our COMPUTER CAVERN shops
22, MORLEY STREET SWINDON
1, LONDON STREET BASINGSTOKE
21, HARRIS ARCADE READING
1, WOOD STREET WALLINGFORD
ALSO MILTON KEYNES SUNDAY MARKET
AUGUST 1990 EDITION

CAPRI MARKETING LTD.
COMPUTER CAVERN
9, DEAN STREET
MARLOW
BUCKS
SL7 3AA
0628-891101

****************** C16 & PLUS 4 ONLY 99P **********************

LEAPER
AUTOBAHN
DIAMOND MINE
ARRDVAAK
BERKS
MISSION MARS

DROID ONE
DIAMOND MINE II
SKY HAWK
DIAGON
JAILBREAK
ROBIN RESCUE

RIG ATTACK
WATER GRAND PRIX
APOLLO RESCUE
ZIP
SOLO
TWIN KINGDOM VALLEY

****************** C16 & PLUS 4 £1.99 ************************

MASTERTRONIC

GUNLAW
STORM
XADIUM
FORMULAR 1 SIMULATOR
GWYNN
FRENISIS
MASTERCHESS
MEGABOLTS
STARFORCE NOVA
RETURN OF ROCKMAN
STREET OLYMPICS
FINGERS MALONE
OBLIDO
DINGBATS
SPEED KING
ONE MAN & HIS DROID
BATTLE
POWERBALL
MOLECULE MAN
P.O.D.
KANE
PROSPECTOR PETE
SPORE

TYNESOFT

BATTLE STAR *NEW*
EUROPEAN GAMES *NEW*
GUNSLINGER
WORLD CUP
POGO PETE
O 'LEVEL PHYSICS
O 'LEVEL CHEMISTRY
O 'LEVEL MATHS

ENCORE

FIREBIRD

VIDEO CLASSICS
ZOLYX
THRUST
INTO THE DEEP
GOLD RUSH
FURY
NET RUN 2000
HARVEY HEADBANGER
SHARK

COMMODORE

MAYHEM
HARBOUR ATTACK

CHANNEL 8

CIRCUS
ESCAPE PULSAR 7

PLAYERS

HOLLYWOOD POKER (+4)
REFLEX
BUBBLE TROUBLE
DIZZY DICE
JOE BLADE II
AUTOZONE
TOMCAT

LLAMASOFT

PSYCHEDELIA C16

CASCADE

ACE C16 ONLY
ACE 2 +4 ONLY

CODEMASTERS

TERRA COGNITA
DANGER ZONE

BLUE RIBAND

S.DAVIS SNOOKER

ALTERNATIVE

INDOOR SOCCER
PHEENIX
SOCCER BOSS
SPACE PREEKS TRIPLE
SABOTEUR
MONKEY MAGIC TRIPLE
INVASION FORCE TRIPLE
ROBIN RESCUE TRIPLE
ARTHUR NOID
LIBERATOR
FIENDS
GALAXIONS TRIPLE
TAZZ TRIPLE
TRAIL BLAZER TRIPLE
WHO DARES WINS II

AUDIOGENIC

CHIP FACTORY
LEAPING LUIE

ATLANTIS

LEAGUE CHALLENGE
COPS & ROBBERS
PANIC
DEATH RACE
SURVIVORS

SCOOBYDOO
BOMBJACK C16 I or II
AIRWOLFE I or II
FRANK BRUNO BOXING C16

LIVEWIRE

GRIDTRAP
SALVAGE (+4 ONLY)

3D GLOOPER
SPACE ESCORT
MYRIAD

****************** C16 & PLUS 4 £2.99 *************************

HIT SQUAD

YIE ARE KUNG FU
GREEN BERET

PLAYERS GOLD

TASKFORCE

ENDURANCE/ADDICTIVE

INTERNATIONAL KARATE
FOOTBALL MANAGER

CRL

ROOM TEN
SUNSTAR (C16)

ENCORE

COMMANDO (C16 ONLY)
GHOST & GOLINS
PAPERBOY

MICRODEAL

ULTIMATE ADVENTURE
WILLIAMSBURG ADVEN(C16)
JERUSALEM ADVENTURE
ARENA 3000

TYNESOFT

INDOOR SPORTS *NEW*
PHANTOM *NEW*

MICRO VALUE

WORLD CUP II *NEW*

AUDIOGENIC

PAINTBOX

IMAGINE

WORLD SERIES BASEBALL

MASTERTRONIC

BANDITS AT ZERO
ON CUE
COMPILATION

GREMLIN

WIMBLEDON

ANIROG

SKRAMBLE
3D TIME TREK

BLUE RIBAND

SYSTEM 8
TURFFORM

ALTERNATIVE

STUNT EXPERTS

****************** C16 & PLUS 4 £3.99 *************************

CRL

DEATH OR GLORY

ELITE

HIT PAK (4 GAMES)

B.CLOUGH FOOTBALL +4

COMMODORE DISC

HULK (PLUS 4 ONLY)

A.D.L.

POWERPACK (4 GAMES)
GRAND MASTER CHESS

COMMODORE CARTRIDGES

STRANGE ODYSSEY
VIDUZZLES

TYNESOFT

MICRO GOLD
4 GREAT GAMES 1,2 & 3

****************** PLUS 4 DISC £4.99 **************************

INFORCOM

CUTHROATS
ZORK I,II,III

COMMODORE

TYPING PROFESSOR
READING PROFESSOR

We also have many single titles in stock at simular prices, if you have a particular title you want, please ask the chances are we have it.

************************** STOP PRESS ******************************

ANCO STRIP POKER II <PLUS 4 ONLY> £ 6.99
COMPANY PACK - BUSINESS ACCOUNTS PACKAGE <PLUS 4 DISK - FEW ONLY> £12.99
COMPUTER HITS 4 (12 GAMES) £ 5.99 ¦COMPUTER HITS 5 (5 GAMES) £ 5.99

PERIPHERIALS

Phono Lead	£ 1.50	Capri Joystick Adaptor	£ 2.95
C16 RAMPACK	£14.95	Konix Speedking Joystick	£ 9.95
QUICKJOY III Joystick	£ 9.95	MEGABLASTER JOYSTICK	£ 4.99
Dust cover (State Model)	£ 3.95	Head Alignment tape + game	£ 7.99

Com. Plus 4 P.S.U's £22.95 ¦ Cbm C16 PSU'S £ 9.95

NOTE: All above joysticks need adaptor.

N.B.As we offer such a large range some titles are occasionally out of stock therefore please state your ALTERNATIVE choice when ordering

NOTE: prices are for CHEQUE (BACKED BY CARD NUMBER FOR IMMEDIATE DELIVERY) WITH ORDER PLUS 50P PER ORDER P&P.£5 PLUS FREE. EUROPE PLEASE ADD £1.00, ELSEWHERE ADD £2.50. FOR POSTAGE & PACKING.

-------------------------------- cut ------------------------------------

If you wish to pay via ACCESS or VISA please complete & return this form.

HOLDERS FULL NAME.............................EXPIRY DATE..........

ADDRESS..

....

SIGNATURE...............................ACCOUNT No.................

We thankyou for your order and look forward to receiving more, please tell your friends as we stock over 3000 titles for ALL home computers & GAMES MACHINES. We will keep you upto date with any new C16/+4 products EVERY TIME YOU ORDER.)

CHRISTENINGS
BIRTHDAYS

C.R.VIDEO

ENGAGEMENTS
WEDDINGS

W.D. BRIGHTON 55a OCCUPATION LANE, SHEFFIELD, S12 4PS TELEPHONE: 641046

VIDEO 4

TITLE MASTER I

Commodore Plus/4 owners!

This easy to use programme. Six lines with thirteen rows of large self centering text. 107 colours possible for border, screen, background and mixed character colours. Plus normal size text on line six for scrolling R-L or usual mode in mixed colours. Full Instructions Included.

cassette ~~ONLY £9.99~~ Inc. p&p

TITLE MASTER II

COMMODORE PLUS/4 & C64/128

A useful alternative to scrolling titles. Menu driven program allows you to design screens using five self centering text sizes (one size per line) Automatic or manual line selection. Change colour of any line or whole screen Easy to use. full instructions included Fastloading cassette ~~£15.00~~ inc. P&P (State micro).

£18-00 for both to Clubmembers only!

from: Mr. W. D. Brighton
55a Occupation Lane
SHEFFIELD S12 4PS
0742 641046

VHS Beta VIDEO 2000

```
2 REM ******************
3 REM * EDWARD KA-SPEL *
4 REM * HI-RES PICTURE *
5 REM ******************
6 REM * BY KEVIN WHEALS*
7 REM *                *
8 REM * (C) 1990       *
9 REM ******************
10 COLOR0,2:COLOR4,2
20 POKE55,0:POKE56,60:CLR:POKE1177,62
30 FORI=0TO1023:POKE15360+I,PEEK(53248+I):NEXTI
40 POKE1177,63:POKE65299,60:POKE65298,192:SCNCLR
50 FORL=0TO100:CX=0:FORD=0TO7:READCH:CX=CX+CH:POKE15576+L*8+D,CH:NEXTD
60 READCH:IFCH<>CXTHENPOKE65299,208:POKE65298,196:PRINT"ERROR IN LINE";1000+(L*1
0):STOP
70 NEXTL
100 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
110 PRINTTAB(14);:PRINT"[££]↑←££££!"
120 PRINTTAB(14);:PRINT"#$%&'()*  +":REM USE 2 NORMAL SPACES
130 PRINTTAB(14);:PRINT",-''''''./ +":REM USE NORMAL SPACE
140 PRINTTAB(14);:PRINT"012345''6 +":REM USE NORMAL SPACE
150 PRINTTAB(14);:PRINT"78 9: ;'< +":REM USE NORMAL SPACES
160 PRINTTAB(14);:PRINT"=> ?─♠│‾‾ +":REM USE NORMAL SPACES  SHIFT * A B C D
170 PRINTTAB(14);:PRINT"‾─│ │╮ ᐯL╲ +":REM USE SHIFT E F G H I J K L M  NORMAL SPA
CE
180 PRINTTAB(14);:PRINT"#╱┌┐ ●─♥│  +":REM USE SHIFT N O P Q R S T  NORMAL SPACES
190 PRINTTAB(14);:PRINT"#  ╱╳ ○ ♣│ +":REM USE SHIFT U V W X Y  NORMAL SPACES
200 PRINTTAB(14);:PRINT"#  ♦┼▒ │π◥ +":REM USE NORMAL SPACES  SHIFT Z +  CBM -
205 REM SHIFT -  CBM = *  SHIFTED SPACE
210 PRINTTAB(14);:PRINT"#  ▌▄‾┘ ▒ ▙":REM USE CBM K I T @ G + M £  NORMAL SPACES
220 PRINTTAB(14);:PRINT"#◤ │├   ▪┐''":REM USE SHIFT £  CBM N Q D Z S  NORMAL SPAC
ES
230 PRINTTAB(14);:PRINT"▂'' ┌ ┴┬┤│ ''":REM USE CBM P A E R W H  NORMAL SPACES
240 PRINTTAB(14);:PRINT"''▌  ▛‾▄''":REM USE CBM J L Y U O  NORMAL SPACE
250 PRINTTAB(14);:PRINT"''▟ ▀▘▀▚''":REM USE SHIFT @  CBM F C X V B
300 GOTO300
500 REM *** UDG DATA FOLLOWS ***
1000 DATA 255,128,128,128,128,128,128,128, 1151
1010 DATA 255,000,000,000,000,000,000,000, 0255
1020 DATA 255,000,000,000,007,015,031,063, 0371
1030 DATA 255,000,000,000,255,255,255,255, 1275
1040 DATA 255,000,000,000,128,192,224,240, 1039
1050 DATA 000,000,000,000,000,000,000,000, 0000
1060 DATA 255,001,001,001,001,001,001,001, 0262
1070 DATA 000,000,000,000,000,000,000,000, 0000
1080 DATA 128,128,128,128,128,128,128,128, 1024
1090 DATA 000,000,000,007,015,015,031,030, 0098
1100 DATA 000,000,127,255,255,255,255,255, 1402
1110 DATA 063,063,063,255,255,255,255,255, 1464
1120 DATA 255,255,255,255,255,255,255,255, 2040
1130 DATA 248,254,255,255,255,255,255,255, 2032
1140 DATA 000,000,131,255,255,255,255,255, 1406
1150 DATA 000,000,000,224,240,240,248,248, 1200
```

```
1160 DATA 001,001,001,001,001,001,001,001, 0008
1170 DATA 128,128,128,128,128,131,131,131, 1033
1180 DATA 030,031,063,127,255,255,255,254, 1270
1190 DATA 248,248,252,255,255,255,255,255, 2023
1200 DATA 000,000,000,000,128,192,192,192, 0704
1210 DATA 135,135,135,135,131,130,134,134, 1069
1220 DATA 255,223,191,255,031,007,015,030, 1007
1230 DATA 255,252,255,252,243,128,000,000, 1385
1240 DATA 095,223,157,032,192,064,032,016, 0811
1250 DATA 255,247,251,243,001,001,000,000, 0998
1260 DATA 255,255,255,255,251,255,035,001, 1562
1270 DATA 224,224,224,240,240,248,248,248, 1896
1280 DATA 134,142,142,142,143,143,143,143, 1132
1290 DATA 030,223,254,254,248,240,240,240, 1729
1300 DATA 016,008,004,002,002,001,002,002, 0037
1310 DATA 000,000,000,000,000,000,128,128, 0256
1320 DATA 255,255,255,127,063,031,031,015, 1032
1330 DATA 248,252,252,252,252,252,252,248, 2008
1340 DATA 143,135,135,135,135,135,135,135, 1088
1350 DATA 248,248,252,224,240,112,112,120, 1556
1360 DATA 002,000,008,008,016,000,032,064, 0130
1370 DATA 064,032,016,008,005,002,001,003, 0131
1380 DATA 000,014,048,199,031,255,255,237, 1039
1390 DATA 003,007,098,248,225,226,242,241, 1290
1400 DATA 255,255,127,063,031,015,015,015, 0776
1410 DATA 252,252,252,252,252,252,252,232, 1996
1420 DATA 131,129,129,128,128,128,128,128, 1029
1430 DATA 060,020,030,138,070,046,062,060, 0486
1440 DATA 000,001,007,000,000,001,003,196, 0208
1450 DATA 000,192,000,060,126,254,188,254, 1074
1460 DATA 003,011,003,005,002,002,065,065, 0156
1470 DATA 222,207,195,255,247,241,120,120, 1607
1480 DATA 225,225,225,192,130,196,064,024, 1281
1490 DATA 007,007,007,003,003,001,000,001, 0029
1500 DATA 200,200,028,158,222,254,126,254, 1442
1510 DATA 044,020,010,002,001,000,000,000, 0077
1520 DATA 224,097,032,032,016,128,192,096, 0817
1530 DATA 030,248,024,008,008,000,008,000, 0326
1540 DATA 060,000,000,000,000,000,000,048, 0108
1550 DATA 014,003,001,000,000,000,000,000, 0018
1560 DATA 001,001,129,193,161,145,081,073, 0784
1570 DATA 252,252,248,248,240,224,192,192, 1848
1580 DATA 048,016,008,004,002,002,001,000, 0081
1590 DATA 008,008,008,112,192,008,008,136, 0480
1600 DATA 080,000,056,048,016,000,000,000, 0200
1610 DATA 004,034,032,033,033,034,036,034, 0240
1620 DATA 128,128,128,000,000,000,000,000, 0384
1630 DATA 136,072,040,036,016,008,004,002, 0314
1640 DATA 000,000,001,003,007,014,004,006, 0035
1650 DATA 032,127,060,124,024,001,096,120, 0584
1660 DATA 000,000,000,128,064,032,144,000, 0368
1670 DATA 033,032,032,064,128,000,000,000, 0289
1680 DATA 000,128,064,120,092,062,047,039, 0552
1690 DATA 000,000,000,000,000,000,000,128, 0128
1700 DATA 001,000,003,006,012,012,016,048, 0098
1710 DATA 004,200,000,002,004,000,000,000, 0210
1720 DATA 024,002,001,096,112,048,000,000, 0283
1730 DATA 000,000,000,128,000,000,000,000, 0128
1740 DATA 000,000,128,064,032,016,016,008, 0264
```

```
1750 DATA 055,063,047,047,047,047,111,111, 0528
1760 DATA 128,128,192,255,255,255,255,255, 1723
1770 DATA 001,001,001,001,193,255,255,255, 0962
1780 DATA 000,000,000,000,001,007,031,127, 0166
1790 DATA 000,003,015,063,255,255,255,255, 1101
1800 DATA 224,192,192,192,192,192,224,224, 1632
1810 DATA 000,000,000,000,000,064,064,064, 0192
1820 DATA 008,004,005,001,003,131,131,135, 0418
1830 DATA 255,255,247,247,247,247,215,255, 1968
1840 DATA 131,143,191,255,255,255,255,255, 1740
1850 DATA 224,224,240,240,240,248,248,248, 1912
1860 DATA 000,000,016,024,024,008,000,000, 0072
1870 DATA 032,032,032,000,000,000,000,000, 0096
1880 DATA 135,135,071,065,064,064,064,064, 0662
1890 DATA 255,255,253,221,093,093,093,093, 1356
1900 DATA 248,248,248,248,252,252,252,254, 2002
1910 DATA 004,004,000,002,002,000,001,000, 0013
1920 DATA 000,000,000,008,004,004,000,000, 0016
1930 DATA 064,065,064,032,033,033,033,000, 0324
1940 DATA 222,199,207,237,237,237,237,237, 1813
1950 DATA 254,255,255,255,255,255,255,255, 2039
1960 DATA 000,000,000,000,000,128,128,255, 0511
1970 DATA 000,001,000,000,000,000,000,255, 0256
1980 DATA 000,000,000,000,000,000,000,255, 0255
1990 DATA 160,032,016,048,000,000,000,255, 0511
2000 DATA 237,237,119,127,127,127,127,255, 1356
2005 REM *** END OF DATA ***
```

THIS MAGAZINE NEEDS YOUR ARTICLES!!

PROGRAMS

REVIEWS

SO SEND THEM TO:—
ROY ROBINSON, 112 CLIFF ROAD, HORNSEA N. HUMBERSIDE, HU18 1JE.
TELEPHONE: 0964-534611 after 6pm but not on Friday Night.

OR ELSE!!!!

IN FACT ANYTHING TO DO WITH THE C16/C116/+4 IS MOST WELCOME..

MAGS FOR SALE

```
                COST     SELL
                ------------
ISSUE 41  - £1.00    0.40   → VIZ ADULT'S ONLY
-----------------------------------------------------------------------------
COMPUTER & VIDEO GAMES
----------------------

NOV    89  - £1.20    0.40
---------------------------------------------------------------------------
COMPUTER SHOPPER
----------------

MARCH   90  - £0.88   0.28
---------------------------------------------------------------------------
ACE
---

DEC     89  - £1.50   0.50
---------------------------------------------------------------------------
AMSTRAD ACTION
--------------

NOV     90  - £1.45   0.45
---------------------------------------------------------------------------
YOUR COMMODORE
--------------

NOV     88  - £1.30   0.30
OCT     89  - £1.30   0.30
NOV     89  - £1.30   0.40
---------------------------------------------------------------------------
COMMODORE COMPUTING INTERNATIONAL
---------------------------------

JULY    89  - £1.20   0.25
AUG     89  - £1.20   0.30
OCT     89  - £1.20   0.30
NOV     89  - £1.20   0.30
DEC     89  - £1.20   0.30
JAN     90  - £1.20   0.40
FEB     90  - £1.30   0.40
MAR     90  - £1.30   0.40
APR     90  - £1.30   0.40
---------------------------------------------------------------------------
POPULAR COMPUTING WEEKLY
------------------------
AUG 24-30  1989
NOV   9-15  1989
DEC   7-13  1989
JAN  11-17  1990
MAR  29-04  1990
AUG   9-15  1990
```

POPULAR COMPUTING WEEKLY (CONT)

AUG 23-29 1990
AUG 30-05 1990
SEP 06-12 1990

ANY ISSUE FOR 15p

NEW COMPUTER EXPRESS

ANY ISSUE FROM NUMBER 17. ANY ISSUE COSTS 20P EACH

IMPORTANT NOTICE

WHEN ORDERING PLEASE ENCLOSE A S.A.E SUFFICENT FOR YOUR ORDER.

C16 POKES
90 OAKBANK
FIELDWAY ESTATE
NEW ADDINGTON
CROYDON
SURREY
CRO-9EF

CHEQUES\POSTAL ORDERS ARE MADE PAYABLE TO MR P.SAUNDERS

SPECIAL OFFERS - ONLY LASTS UNTIL DEC 1ST 1990

BOX OF 10 3" DISKETTES (3.5 S/S)

COST £11.21 (STILL WRAPPED) SELL FOR £2.24.
PLEASE ADD £1.26 FOR POSTAGE & PACKING

MPS 801 PRINTER

WITH PRINT CHIP IV (TO BE CONNECTED) COST £125.
SELL FOR £55. ALL IT NEDDS TO GET IF RUNNING
IS A RIBBON & SOME COMPUTER PAPER.

--CHEQUES\POSTAL ORDERS ARE MADE PAYABLE TO--
-- MR PHLIP SAUNDERS --

Mr Roy Robinson,
112 Cliff Road,
Hornsea,
N. Humberside, HU18 1JE.

89 Old Road,
East Peckham
Tonbridge,
Kent, TN12 5EN.
Tel: 0622 871066

6th December 1990

Dear Roy,

I hope members have realised that I made a mistake in last month's magazine when I suggested that PRINT was used on the Graphic Screen. As you will know CHAR only is available on the Graphic Screen but what I was saying about CHAR and Location 740 was correct. Location 740 tells CHAR where its character set is to be found but only for use on the Graphic Screen. This means that one character set can be in use on the Graphic Screen and another (usually the normal Commodore character set) can be in use on the text screen. This means that if your program can be run on the Graphic Screen there is no need to make the text screen character set into your custom character set which causes unreadable information on the screen when things go wrong. Simply point the CHAR graphic set to the start of the character set you have made.

Does anyone know anything about using a light pen with the Plus/4? I missed out on buying one recently advertized in this magazine and so I have bought a kit from Maplin but I need information on connecting it to the computer and a program or an outline of the concept behind writing a program. Does anyone have a commercial program I could borrow? A light pen or mouse program would be very usefully combined with my 'Graphic Editor' program.

Have you realised that 'Graphic Editor' allows you to plan where CHAR statements will appear since the on-screen data is written as CHAR coordinates and if you would like to analyse a character on the screen, the graphic and data after the coordinates let you see how the character is made up.

Here is a program called 'Multicolor Load' which is to be used before using 'Graphic Editor'. It locks all the different COLORS into memory so avoiding annoying incorrect colours on the screen where the data had been (the wrong colours do not mess up your graphic if you SAVE the graphic to tape or disk which is done from MONITOR with S'GREAT GRAPHIC 1',8,2000,3F40 if disk is being used).

```
10 REM ********************************
20 REM *     MULTICOLOR LOAD          *
30 REM *      BY R MARSHALL           *
40 REM *         30.7.90              *
50 REM ********************************
60 IFPEEK(762)=1THEN170
70 COLOR1,12,6
80 COLOR0,3,5:REM THIS WILL BE COLOR 2
90 COLOR3,3,0
100 COLOR4,3,0
110 POKE758,RCLR(1):POKE759,RLUM(1)
120 POKE760,RCLR(0):POKE761,RLUM(0):REM THIS WILL BE COLOR 2
130 GRAPHIC1,1:GRAPHIC0:POKE762,1
140 KEY1,"GRAPHIC0"+CHR$(13)
150 KEY4,"DLOAD"+CHR$(34)+"GRAPHIC EDITOR"+CHR$(34)+CHR$(13)
160 COLOR1,3,0:MONITOR
170 COLOR0,2,7:COLOR1,1
180 POKE763,RCLR(0):POKE764,RLUM(0)
190 POKE765,1:GRAPHIC3
```

When the computer arrives at line140 you will see that MONITOR is entered. You can then load a previously saved graphic such as L'GREAT GRAPHIC 1',8 and to get passed this point in the program just type X and RETURN followed by RUN RETURN and the program will run again but because location 762 now contains a 1, the program jumps to line 170 and we are found on GRAPHIC 3. Change that to GRAPHIC 1 or GRAPHIC 0 as suits your purposes.

When you are satisfied with your graphic you will want to print it - especially if you have a printer! The next program converts all the data on the GRAPHIC SCREEN to the correct form for the Star LC-10 printer.

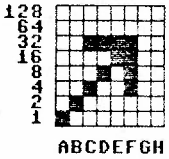
```
10 REM ***********************************
20 REM *         SCREEN PRINTER          *
30 REM *                                 *
40 REM *    PRINT THE GRAPHIC SCREEN     *
50 REM *                                 *
60 REM *      BASIC BY ROB MARSHALL      *
70 REM *  MACHINE CODE BY PETER CRACK    *
80 REM *            4.8.90               *
90 REM ***********************************
100 FORI=0TO62:READI%:POKE1552+I,I%:NEXTI
110 SYS1552:TRAP590
120 COLOR0,3,0:COLOR1,2
130 GRAPHIC0:PRINT"(YEL)(CLR)(DOWN)(DOWN)(DOWN)(DOWN)"
140 PRINT"┌───────────────────────────────────────┐"
150 PRINT"│      THIS PROGRAM IS ONLY FOR USE ON  │"
160 PRINT"│      PRINTERS MODIFIED TO SCROLL THE  │"
170 PRINT"│      PAPER IN THE REVERSE DIRECTION.  │"
180 PRINT"│      BEFORE SWITCHING THE PRINTER ON  │"
190 PRINT"│      SET DIP SWITCHES 1,2 AND 4 OFF   │"
200 PRINT"│          PRESS A KEY WHEN READY       │"
210 PRINT"└───────────────────────────────────────┘":POKE239,0:GETKEYA$
220 PRINT"(CLR)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
)(RGHT)(RGHT)BLACK=0"
230 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RG
T)(RGHT)RED=1"
240 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RG
T)BLUE=2"
250 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)PURPLE=3"
260 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)YELLOW=4"
270 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)ORANGE=5"
280 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)GRE
N=6"
290 A$="":FORI=1TOPEEK(171):A$=A$+CHR$(PEEK(604+I)):NEXTI
300 IFA$=CHR$(16)+"SCREEN PRINTER"+CHR$(0)ORA$=CHR$(16)ORA$=CHR$(0)THENA$=""
310 IFPEEK(171)=0THENGOSUB580
320 PRINT"(RVS)(DOWN)(WHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)"A$"(OFF)":INPUT"(
YEL)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)WHICH COLOR FOR PRINTING";C%
330 FORI=1TO16:POKE999+I,0:NEXTI
340 FORI=1TOLEN(A$):POKE998+I,ASC(MID$(A$,I,1)):NEXTI:OPEN4,4
350 PRINT#4,CHR$(27)CHR$(64);:REM RESET
360 PRINT#4,CHR$(27)CHR$(65)CHR$(8);:  REM LINE SPACING 8/72 INCH FOR GRAPHICS
370 PRINT#4,CHR$(27)CHR$(108)CHR$(20);:REM SET LEFT MARGIN
380 PRINT#4,CHR$(27)CHR$(114)CHR$(C%);:REM CHOSEN COLOR
390 IFPEEK(144)=128THEN570
400 GRAPHIC1:FORF=0TO24:C$="":D$="":FORB=0TO39
410 SYS1568
420 FORI=0TO7:C%(I)=PEEK(224+I)
430 IFB>19THEND$=D$+CHR$(C%(I)):ELSEC$=C$+CHR$(C%(I))
440 NEXTI
450 D%=D%+8
460 NEXTB:FORT=0TO1
470 PRINT#4,CHR$(27)CHR$(42)CHR$(4)CHR$(160)CHR$(0)C$;:REM 160 DOTS WIDE
480 PRINT#4,CHR$(27)CHR$(42)CHR$(4)CHR$(160)CHR$(0)D$;:REM 160 DOTS WIDE
490 PRINT#4:NEXTT:PRINT#4,CHR$(10):NEXTF
500 FORI=0TO24:PRINT#4,CHR$(27)CHR$(10);:NEXTI
510 PRINT#4,CHR$(27)CHR$(64):CLOSE4
520 IFA$=""THENB$="(LEFT)(LEFT)(LEFT)":ELSEB$="  "
```

(Screen Printer continued)

```
530 GRAPHIC0:PRINT"(YEL)(CLR)PRINTING OF";B$;"(RVS)(WHT)";A$;"(YEL)(OFF) COMPLET
ED(DOWN)(DOWN)(DOWN)(WHT)":END
540 DATA162,23,169,0,149,208,202,16,251,169,32,133,209,133,210,96,166,210,208,1,
96
550 DATA164,211,162,7,138,72,177,208,162,0,10,54,224,232,224,8,208,248,200,104,1
70
560 DATA202,16,236,192,0,208,7,230,209,166,210,202,134,210,132,211,96,0,0,0,0
570 GRAPHIC0:PRINT"(DOWN)PRINTER NOT PRESENT":END
580 FORI=1TO16:A$=A$+CHR$(PEEK(999+I)):NEXTI:RETURN
590 GRAPHIC0:PRINT"(DOWN)(WHT)"ERR$(ER)"IN LINE"EL"(YEL)"
```

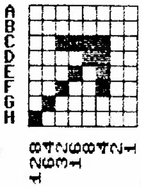
If you do not have a printer that can go backwards or do not want to modify your printer, you can still print the graphic screen. Remove lines 140 to 210 and also line 500

You will have to check with the Users Manual for your printer if it is not a Star LC-10 to make sure that the commands in lines 350 to 510 are appropriate to your printer. Line 410 converts the screen's 8 x 8 block of dots (a character) to the correct data as far as the printer is concerned. The screen and printer are programmed differently to give the same graphical representation.



PRINTER

1,2,36,40,48,
60,0,0



COMPUTER

0,0,124,12,20,
36,64,128

It is important to check if this corresponds with the way your printer works as this is for the Star printer and may be upside down for Commodore printers.

I see Datel Electronics of Stoke-on-Trent are now selling a package of a Star LC-10 color printer (parallel version), a serial to parallel converter and a Graphic Screen converter program for C16/64/+4 computer users for £300. Why they could not do this before now is a mystery, especially when their technical guy told me a year ago that it was not possible (they were selling the serial to parallel converter at that time). The printer can be bought cheaper than they are selling it for and readers of this club magazine do not need to buy the Graphic Screen converter program. Alternatively you can do as I have and buy the Serial version of the printer called LC-10C colour, but if you do that make sure that your mail order company knows exactly what you want and state clearly on your order that it is the Serial version for Commodore Computers that you want. Having had mine for over a year, I am pleased with it but realise that since it has a section dedicated (well nearly) to Commodore I may well be missing out on some facilities which the original version has. The LC-10 has not been in production for 6 months and a friend has the new LC-200 for his Amiga. The Amiga uses the parallel communication system. If anyone is using the LC-10 and has a problem with programming, perhaps you find it too complicated or haven't the time to give to programming, then I would be glad to tackle the problem for you.

If you have a Star LC-10 and would like to modify it to do multiscan graphic printing there is a switch which has to be disabled. The switch is operated by the 'release lever' on the right hand side near the back. The only use of this switch is to tell the microprocessor that we are using fanfold paper and the consequence of that is to permit the printer to move the paper backwards. We want to be able to move A4 cut paper backwards which we otherwise could not do. Either cut a wire going to the switch or remove the switch, put some insulating tape between the contacts and tape the wiring and switch to the base of the printer's case. It must be kept in mind that only three quarters of a sheet of paper can be used for printing on since if the paper end sensor has been passed the paper will be damaged when the printer roller goes backwards after the graphic has been finished and the next scan is prepared for. The code to make the printer go backwards one line is PRINT#4,chr$(27)chr$(10);

Best wishes to all club members,

Rob

Rob Marshall.

```
10 REM **********************************
20 REM *       EMPEROR PENGUIN          *
30 REM *       BY ROB MARSHALL          *
40 REM *          17.5.90               *
50 REM **********************************
60 KEY1,"GRAPHIC0"+CHR$(13)
70 COLOR0,2:COLOR1,1:GRAPHIC1,1
80 DRAW1,129,163TO124,161TO118,161TO112,159TO104,159
90 DRAWTO100,157TO108,147TO101,90TO109,48TO115,33TO124,18TO139,2TO154,0TO160,1
100 DRAWTO178,12TO184,24TO187,33TO188,42TO188,51TO180,70TO178,66TO172,57TO175,63
TO175,67
110 DRAWTO178,72TO181,87TO181,77TO180,70
120 DRAW1,183,63TO184,72TO182,84TO181,87
130 DRAW1,176,67TO160,48TO158,30TO162,28
140 DRAWTO160,24TO158,27TO148,18TO143,16TO138,18TO132,27TO130,40TO124,57TO127,72
150 DRAWTO123,96TO116,119TO114,120TO114,117TO113,99TO112,90TO112,72TO111,62TO109
,57
160 DRAWTO110,96TO109,117TO115,132TO116,147TO130,162TO132,163TO134,161
170 PAINT1,120,159:PAINT1,110,57
180 CIRCLE0,162,19,5,6:PAINT0,162,19:CIRCLE0,165,26,4,10:PAINT0,165,26
190 DRAW1,178,75TO175,93TO175,108TO177,129TO178,138TO172,153
200 CIRCLE1,172,159,1,3
210 CIRCLE1,169,158,1,3
220 CIRCLE1,164,159,1,3
230 CIRCLE1,154,159,1,3
240 CIRCLE1,150,159,1,3
250 CIRCLE1,145,160,1,3
260 CIRCLE1,144,157,2,2,200,440
270 CIRCLE1,142,158,3,5,180,390
280 CIRCLE1,139,156,4,6,180,410
290 CIRCLE1,149,157,2,3,200,440
300 CIRCLE1,148,155,3,3,280,440
310 CIRCLE1,147,154,4,4,300,400
320 CIRCLE1,154,158,2,3,260,440
330 CIRCLE1,153,156,3,3,340,440
340 CIRCLE1,152,155,4,4,280,400
350 CIRCLE1,164,158,2,3,200,410
360 CIRCLE1,163,156,3,3,190,430
370 CIRCLE1,162,155,4,4,200,430
380 CIRCLE1,170,158,2,3,280,410
390 CIRCLE1,169,156,3,3,200,430
400 CIRCLE1,168,155,4,4,310,430
410 CIRCLE1,172,155,2,3,340,490
420 CIRCLE1,138,140,4,10,30,220,16
430 DRAW1,132,147TO142,132
440 CIRCLE1,154,125,12,14,208,410,40
450 CIRCLE1,152,123,15,15,75,160
460 CIRCLE1,140,152,20,20,5,40
470 CIRCLE1,155,134,3,4,160,200
480 CIRCLE1,154,136,12,12,65,120
490 DRAW1TO165,153
500 CIRCLE1,154,132,12,12,120,240
510 CIRCLE1,154,132,16,16,142,230
520 PAINT1,152,123:PAINT1,135,148
530 CIRCLE0,151,126,3,4
540 CIRCLE0,162,126,3,4
```

```
550 PAINT0,151,126:PAINT0,162,125
560 CIRCLE0,148,139,10,10,358,430
570 CIRCLE0,146,143,10,10,370,430
580 CIRCLE0,164,139,10,10,280,360
590 CIRCLE0,154,127,10,10,80,180
600 DRAW0,148,132TO148,128
610 PAINT0,156,136
620 PAINT0,152,133:PAINT0,159,133
630 CIRCLE1,150,127,1:PAINT1,150,128
640 CIRCLE1,161,127,1:PAINT1,161,128
650 CIRCLE1,151,127,1:PAINT1,151,128
660 CIRCLE1,162,127,1:PAINT1,162,128
670 CIRCLE1,150,139,9,9,370,402
680 CIRCLE1,163,138,9,9,302,333,7
690 DRAW0,156,130TO156,128
700 DRAW1,156,131TO156,133
```

```
1 REM ***********************************
2 REM * CHANNEL 4 LOGO USING GRAPHIC 3 *
3 REM *                                 *
4 REM *    BY ROB MARSHALL  9.12.90     *
5 REM ***********************************
10 KEY1,"(WHT)GRAPHIC0"+CHR$(13)
20 COLOR1,3,2:COLOR0,1:COLOR2,7,3:COLOR3,16,3:COLOR4,1
30 GRAPHIC3,1
40 DRAW1,80,149TO80,47TO86,32TO86,149TO80,149:PAINT1,81,50:REM RED
50 BOX2,72,130,78,162:PAINT2,73,131:REM BLUE
60 DRAW2,72,115TO72,39TO78,23TO78,115TO72,115:PAINT2,73,50:REM BLUE
70 BOX3,44,117,78,128:PAINT3,45,118:REM GREEN
80 BOX3,88,117,100,128:PAINT3,89,118:REMGREEN
90 COLOR2,9,4:REM YELLOW
100 DRAW2,45,115TO70,45TO70,65TO52,115TO45,115:PAINT2,47,114
110 DRAW2,80,41TO80,20TO81,18TO86,26TO80,41:PAINT2,82,33
120 COLOR2,5,2:REM PURPLE
130 BOX2,62,151,70,162:PAINT2,63,152
140 BOX2,80,151,93,162:PAINT2,81,152
```

Eric Jones The Fold Bucknell Shropshire SY7 OAA
05474 348 90-11-29

Dear Editor Roy

SENDING MEMORY-DUMP, DISASSEMBLY, OR LISTING TO PRINTER

For printing out memory-dumps and disassemblies, for some time now I've been using a system that's very like the one described by Peter Crack in the 'Aug/Sept' issue just received, except that my system doesn't print the registers at the beginning or 'READY' at the end.

Before you start, decide which blocks of memory you want to disassemble or dump. Then in direct BASIC mode enter 'OPEN4,4 «return»'.

Next, ***as a single direct command***, enter

'CMD4,CHR$(14)"TITLE"CHR$(15);:SYS62613 «return»'

This sends the title to the printer as double-width characters — if your printer can't do them then just leave out the 'CHR$(14)' and 'CHR$(15)' — and puts you into MONITOR; the printer won't start printing yet, because of the semicolon after the title.

Now enter your memory-dump or disassembly commands using 'Dssss ffff «return»' and/or 'Mssss ffff «return»' in the usual way, as many of them as you like. When you enter the first one, the printer prints the title followed by one blank line and then the print-out — all but the last line. When you enter the next one, that last line is printed and followed by one blank line, and then the next print-out all but its last line; and so on.

Finally enter 'X «return»'. This prints out the last line of all, cancels the 'CMD4', and then returns you to BASIC. You can now enter 'CLOSE4 «return»' without having to enter 'PRINT#4' first — or of course you can leave File 4 open and do some more printing.

For BASIC listings, after opening the file, enter ***as a single direct command***

'CMD4,CHR$(14)"TITLE"CHR$(15):LISTss-ff:LISTss-ff:PRINT#4 «return»'

wherein you can put as many 'LISTs' (with or without their start and finish line numbers) as you like — provided they will all fit into a single command. (Note that there's no semicolon in this command. You get one blank line after the title, but you get two blank lines between 'LISTs'.)

Sorry I can't send you specimens — I'm right out of thermal-printer paper!

I enclose two copies of this for you to pass on, one to Peter Crack and one to Kevin Wheals who inspired Peter's letter.

Cheers!

Eric

PS: I hope to have some more for you before too long — thermal paper supplies permitting; or perhaps one day I'll get round to finding a way to couple the C-16 into the Amstrad's printer . . .

PPS: What's happening about QUIXAVER?

I'll be in touch shortly about Quixaver Eric. (ED, Roy).

Dear Roy,

As promised in our telephone conversation, here are the two programs, and I hope that they will be of interest to some of the members. I've put two copies of each on the disk for you,and put It into a sleeve so that you can see how it works out. I found that UHU glue stick for paper was ideal for sticking the flaps and it is'nt messy either.

The DISK SLEEVE PRINTER was published in YOUR COMMODORE in 1988 in a C16 SPECIAL, the auther was J.HOYLE.

The CRIBBAGE game was also printed in YOUR COMMODORE and this was in APRIL 1987 and the auther was HUGH MC GHEE.

Well thats all for now ROY, I hope that you have had some luck with that C16 chip I'm sorry that I could'nt have been more help in that respect.

Bye For Now

JIM NICHOL

DECEMBER 1990 & JANUARY 1991
Please send me Vol 2 Issue's 9&10 90/91 of 'C16/C116/+4 COMPUTING MONTHLY'.

I enclose a PO/CHEQUE for the value of £2.00.

NAME & ADDRESS:

SIGNED: ______________________________

Dear Member please send £2, because you have received the Dec '90 ish already hand have yet to pay for it, and the other £1 is for the Jan 1991, which is yet to come.

****
* cribbage *
****

Instructions

It is assumed that the player knows how to play cribbage. The rules are exactly those given in HOYLES RULES of GAMES. Since some aspects of scoring points vary from region to region, the points scored in this version will be reiterated in this introduction. The following points scored should thus be noted.

1) Due to the layout of the cards, X is used to represent a 10.

2) Input is handled by means of a joystick in port two. When asked to play or discard a card, simply move the marker which appears above the cards to the required position and press the fire button. When entering your points, moving the joystick < right > and < left > increases and decreases the points counter respectively. Once again pressing fire enters your response. The program is protected against attempts to discard or play the same card twice. Attempts to claim too many points produce an error message. The PLUS/4 always claims the correct number of points for itself but,if you don't claim all your points,the PLUS/4 will steal them.

3) Each player is dealt six cards and must discard two cards into the crib. The PLUS/4 plays a simple strategy choosing its best four cards after consideration of the two cards it is discarding. The cards are shuffled after every hand but only cards are shuffled out of the pack to save time. The program differentiates between dealer and non dealer. The player who does'nt have the crib always receives the first card.

4) The PLUS/4 cuts the cards for the initial crib. The lowest card wins. Note that in this game the Ace always counts as one. Illegal combinations are not allowed so two Aces of Spades, for example, will not be displayed. Ties however such as both players cutting a 2 of different suits, are permitted, in which case the cut will be performed again.

5) The first player to obtain 121 points or more wins.

6) Note that to the run up the PLUS/4 will check your cards if you say you can't go. This is the only time the PLUS/4 examines your cards but it does not cheat. Any discovery of an ability to play after entering 'can't' go will produce an error message,as will an attempt to play a card that would result in a total excess of 31. The PLUS/4 does not play a strategy in the run up except that it will always make the total of 31 if it can. It will not,however,always make the total 15 even if it can.

7) The ending of a go in the run up has proved the most tricky operation in practice. If the player plays last the PLUS/4 does not know that the player cannot go unless the total is 31 or the last card has been played. Otherwise the player must enter 'can't' go before entering the last point.

Points are scored as follows:

A) 2 points for any pairs, 6 points for three cards of a kind and 12 points for four cards of a kind.

B) 2 points for any combination of cards totalling 15, counting picture cards as 10.

C) For any run up of three or more cards, count 1 point for each card in the run.

D) For obtaining a total of 15 or 31 in the run up, score 2 points.

E) For playing the last card in any go in the run up, score 1 point.

F) For a flush of four cards in your hand, score 4 points. Note there are no points for a flush or four in the crib.

G) For a flush of five, either in your hand or in the crib, score 5 points.

H) If the starter card is a Jack, and it is in your crib, add 2 points when counting your hand in the main game.

I) If you have a Jack either in you hand or in the crib of the same suit as the starter card, count 1 point.

---

8) The game is played on the high resolution screen to facilitate the different background colours.

FREEZE FRAME

Dear Roy
As I was saying 'FREEZ FRAME' is now under way, although I have'nt had time to send you anything yet. It will cost 50p to each member (postage included), or 30p + S.A.E. a tape and a picture/photo/drawing, anything will do.

My strongest would be planes, but anything sent in to my address, marked 'FREEZE FRAME' (all photos/drawings etc will be returned!)

Anyway thats all for now, all the best

Simon Pollard, 7 Seavy Road, GOOLE, N.Humberside, DN14 6TA. Phone 0405-768898

My apologies go out to Simon, because I lost this letter and have just found it, please contact Simon for more details on this Service.

DISK SLEEVE PRINTER

FOR PLUS/4,C16 and 64 WITH 1541 OR 1551 DISK DRIVE and COMMODORE COMPATABLE PRINTER

Knowing which files a particular disk contains, without having to load in the directory, can be rather a messy buisiness of sqeezing the relevent information into the small space allowed on the labels supplied with the disk. An alternative method is to write the disk information on the corresponding disk's paper sleeve, in a similar manner as done with records and tapes. This simple BASIC program allows the directory of a 5.25 floppy disk to be listed to any Commodore compatible printer, in the foremat pattern of a disk sleeve.

The program may be run with either single or double sided disks, but the number of files contained on any side must not be more than 42. Once the front of the disk sleeve has been filled, the listing continues on the reverse side of the sleeve. Once the program has run, the result will be similar to the demonstration one submitted. All that remains to be done is to cut around the dotted lines, fold and glue to form a sleeve. Putting the original sleeve inside this makes the sleeve a little stiffer.

PROGRAM NOTES

Type in the program as listed - the REM statements may be omitted if desired - and then save.

The program was developed for a BROTHER HR5-C 80 colomn dot matrix printer, it should run other COMMODORE compatible dot matrix printers, The line spacing should be set to 1/16 inch if possible,although this is not essential.

USING THE PROGRAM

Connect the printer to your computer and load with paper - the pattern is printed in the center of a piece ofA4. Load in the SLEEVE PRINTER program, and RUN it. Select single or double sided disk when prompted, and insert the disk to be directoried into the drive (side A if you are using a double sided disk). After pressing any key, the program loads the disk directory and exracts the file names and file types, sorting them into a format ready for printing. If the double sided disk option is being used then the disk should be reversed, when prompted,so that the operation can be repeated for the other side. If the single sided disk option is chosen, then the program will go direct to the print routine.

Before printing commences, one final prompt to load paper is given after which any key should be pressed to continue. Once printing has ended, remove the paper from the printer. Cut out the pattern along the dotted lines and then fold along the solid line. The flaps should then be folded over the back of the sleeve and glued-checking that the disk fits properly. After the glue has dried, insert the disk into its new home.

SLEEVE SAMPLE

```
 -----------------------------------------------------------------------
|    * *********************************************** *
|    * * "DISK2                 " 01 2A *           * *
|    * *********************************************** *
|    * *                                            * *
|    * *   "+4 DISK MENUS"      PRG                 * *
|    * *   "+4 ARTIST 2"        PRG                 * *
|    * *   "HEAD BASHER"        PRG                 * *
|    * *   "DEMOLITION"         PRG                 * *
|    * *   "@SCROLL ROUTINES"   PRG                 * *
|    * *   "SPACE SHIP"         PRG                 * *
|    * *   "JACKPOT"            PRG                 * *
|    * *   "PYRAMIDE"           PRG                 * *
|    * *   "STACK 16"           PRG                 * *
|    * *   "RANGE BALL"         PRG                 * *
|    * *   "BUGGY"              PRG                 * *
|    * *   "BACKGAMMON"         PRG                 * *
|    * *   "OTHELLO"            PRG                 * *
|    * *   "DRAUGHTS"           PRG                 * *
|    * *   "TOPPER"             PRG                 * *
|    * *   "SPACEMAZE"          PRG                 * *
|    * *   "HIGHER/LOWER"       PRG                 * *
|    * *   "FROG +4"            PRG                 * *
|    * *                                            * *
|****  *********************************************  ****|
 -----------------------------------------------------------------------
        |***********************************************|
        |* "DISK2                 " 01 2A *            *|
        |***********************************************|
        |*                                             *|
        |*  "DRIVER"            PRG                    *|
        |*  "QUEST"             PRG                    *|
        |*  "DEATH SHIP"        PRG                    *|
        |*  "DOUBLE"            PRG                    *|
        |*  "PUZZLER"           PRG                    *|
        |*  "NUMBERS"           PRG                    *|
        |*  "SUPERBALL"         PRG                    *|
        |*  "CRAZY JUMPMAN"     PRG                    *|
        |*  "MOVING BALL2"      PRG                    *|
        |*  "+4 ARTIST"         PRG                    *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |*                                             *|
        |***********************************************|
         -----------------------------------------------
```

SLEEVE SAMPLE

```
* ******************************************************* *
* * "DISK4            " 01 2A *                          * *
* ******************************************************* *
* *                                                      * *
* *  "WORD SPLASH"         PRG                           * *
* *  "WORD GAME"           PRG                           * *
* *  "DICE PROGRAM"        PRG                           * *
* *  "SKETCHPAD"           PRG                           * *
* *  "COUNTING"            PRG                           * *
* *  "NUMBERS GAME"        PRG                           * *
* *  "NUMBERS GAME2"       PRG                           * *
* *  "GUSTAVUS"            PRG                           * *
* *  "GUSTAVUS2"           PRG                           * *
* *  "HI-LO"               PRG                           * *
* *  "HI-LO2"              PRG                           * *
* *  "DUO DRAUGHTS"        PRG                           * *
* *  "DUO DRAUGHTS2"       PRG                           * *
* *  "O'GRADY SAYS"        PRG                           * *
* *  "O'GRADY SAYS2"       PRG                           * *
* *  "P1 DICE"             PRG                           * *
* *  "MAGNIFIED CHAR"      PRG                           * *
* *  "P53 CIRCLES"         PRG                           * *
* *                                                      * *
**** ******************************************************* ****
```

```
|*******************************************************|
|* "DISK4            " 01 2A *                        *|
|*******************************************************|
|*                                                     *|
|*  "P89 RESISTORS"     PRG                            *|
|*  "P47 WORM"          PRG                            *|
|*  "COLOR/LUMINANCE"   PRG                            *|
|*  "TEXTMODE EXAMPLE"  PRG                            *|
|*  "HI-RES EXAMPLE"    PRG                            *|
|*  "MULTICOLOR EX"     PRG                            *|
|*  "HI-RES CHARS"      PRG                            *|
|*  "MULTI CUST CHAR"   PRG                            *|
|*  "EXTD COL MODE"     PRG                            *|
|*  "SCROLLING EX"      PRG                            *|
|*  "CHAR ANIMATION"    PRG                            *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*                                                     *|
|*******************************************************|
100 REM************************
110 REM*                      *
120 REM* DISK SLEEVE PRINTER  *
130 REM*                      *
140 REM*      BY J. HOYLE     *
150 REM*                      *
160 REM*          1987        *
170 REM*                      *
180 REM************************
190 REM*                      *
200 REM*     FOR USE WITH     *
210 REM*                      *
220 REM*   C64, PLUS4 C16     *
230 REM*                      *
240 REM*     AND COMMODORE    *
250 REM*                      *
260 REM* COMPATABLE PRINTERS  *
270 REM*                      *
280 REM************************
290 REM
300 REM
310 REM ******** MAIN LOOP ********
320 REM
330 DIMT$(50,1),A$(50,1)
340 S=0
350 PRINT"■■■SINGLE OR DOUBLE SIDED (S/D) "
360 GETS$
370 IFS$<>"S"ANDS$<>"D"THENGOTO360
380 IFS$="S"THENGOTO410
390 GOSUB550
400 GOSUB740
410 GOSUB550
420 GOSUB740
430 GOSUB900
440 PRINT"■■■CUT OUT AND GLUE PATTERN, AND THATS IT"
450 PRINT"■■■■■■■■DO YOU WANT TO RUN AGAIN ■■ (Y/N) ?"
460 GETA$
470 IFA$<>"Y"ANDA$<>"N"THEN460
480 IFA$="Y"THENRUN
490 END
550 PRINT"■■■■■■■■■■■■PLACE SIDE '"CHR$(65+S)"' OF THE DISK IN DRIVE"
560 PRINTTAB(11) "■■■■THEN PRESS ANY KEY"
570 GETK$
580 IFK$=""THEN570
590 PRINT"■■■■■■■■■■■■■■■PLEASE WAIT WHILST LOADING.....■"
600 C=0
610 OPEN15,8,15:OPEN1,8,0,"$0:*
620 FORCO=1TO32
630 GET#1,T$
640 T$(C,S)=T$(C,S)+T$
650 NEXTCO
660 C=C+1
670 IFST=0THEN620
680 CLOSE1
690 CLOSE15
700 RETURN
740 PRINT"■■■■■■■■■■■■■■■PLEASE WAIT WHILST SORTING.....■"
750 A(S)=C
760 A$(0,S)=T$(0,S)
770 FORCO=1TOC-1
780 A$(CO,S)=" "
790 FORL=1TOLEN(T$(CO,S))
800 T$=MID$(T$(CO,S),L,1)
810 IFASC(T$)<>34THEN840
```

```
830 L=LEN(T$(CO,3))
840 NEXTL:NEXTCO
850 S=1
860 RETURN
900 PRINT"████"TAB(18)"O.K...."
910 PRINT"█████████POSITION THE PAPER IN THE PRINTER"
920 PRINT"████AND THEN PRESS ANY KEY TO EXECUTE"
930 GETK$
940 IFK$=""THEN930
950 OPEN4,4:PRINT#4:P$=CHR$(16)
960 PRINT#4,P$"05 ┌────┬──────────────────────────────────────────────────────┬
────┐"
970 PRINT#4,P$"05|   * ******************************************************* *
    |"
980 PRINT#4,P$"05|   * *";:PRINT#4,P$"13"A$(0,0);:PRINT#4,P$"38█*";:PRINT#4,P$"4
0"A$(0,1);
990 PRINT#4,P$"65█* *    |"
1000 PRINT#4,P$"05|   * *******************************************************
*    |"
1010 PRINT#4,P$"05|   * *                                                       *
*    |"
1020 FORL=1TO18
1030 PRINT#4,P$"05|   * *";:PRINT#4,P$"13"A$(L,0);:PRINT#4,P$"40"A$(L,1);:PRINT#
4,P$"65* *    |"
1040 NEXTL
1050 PRINT#4,P$"05|   * *                                                       *
*    |
1060 PRINT#4,P$"05|**** *******************************************************
****|
1070 PRINT#4,P$"05 └────┼──────────────────────────────────────────────────────┼
────┘
1080 PRINT#4,P$"05     |*******************************************************|
1090 IFA(0)<13ANDA(1)<19THEN1190
1100 PRINT#4,P$"05     |*";:PRINT#4,P$"13"A$(0,0);:PRINT#4,P$"38█*";:PRINT#4,P$"
40"A$(0,1);
1110 PRINT#4,P$"65█*|"
1120 PRINT#4,P$"05     |*******************************************************|
1130 PRINT#4,P$"05     |*                                                     *|
1140 FORL=19TO42
1150 PRINT#4,P$"05     |*";:PRINT#4,P$"13"A$(L,0);:PRINT#4,P$"40"A$(L,1);:PRINT#
4,P$"65*|"
1160 NEXTL
1170 PRINT#4,P$"05     |*                                                     *|
"
1180 GOTO1220
1190 FORL=1TO27
1200 PRINT#4,P$"05     |*                                                     *|
"
1210 NEXTL
1220 PRINT#4,P$"05     |*******************************************************|
"
1230 PRINT#4,P$"05     └───────────────────────────────────────────────────────┘
"
1240 PRINT#4
1250 CLOSE4
1260 RETURN
1270 CLOSE4
```

36

```
50 REM *********** CRIBBAGE ***********
60 REM ................................
70 REM
1000 DEF FNA(Z)=INT(RND(1)*Z)+1
1010 GOSUB3360:GOSUB4390
1020 GOSUB3710:GOSUB3470
1030 DO
1040 :GOSUB3770
1050 :GOSUB4060:GOSUB3860:GOSUB1900
1060 :IFMP>120ORYP>120THEN1080
1070 :GOSUB1100:GOSUB3470
1080 LOOPUNTILMP>120ORYP>120
1090 GOTO4530
1100 TP=4:R=2:W=9:Q$=SD$(0,12):S$=SD$(1,12):GOSUB2470
1110 FORA=0TO4STEP2
1120 :F$(A,4)=Q$:F$(A+1,4)=S$:V%(A,4)=P%(0,12):V%(A+1,4)=P%(1,12)
1130 NEXT
1140 IFWC=2THEN1210
1150 A=2:PO=0:GOSUB1320:IFYP>120THEN1300
1160 B=40:R=18:S=24:C=0:GOSUB3330
1170 S$="FOR ME":A=0:PO=0:GOSUB1430:IFMP>120THEN1300
1180 A=0:W=0
1190 FORR=16TO37STEP7:S$=F$(5,A):Q$=F$(4,A):GOSUB2470:A=A+1:NEXT
1200 S$="IN THE CRIB FOR ME":A=4:PO=0:GOSUB1490:WC=2:GOTO1270
1210 S$="FOR ME":A=0:PO=0:GOSUB1440:IFMP>120THEN1300
1220 B=40:C=0:R=0:S=6:GOSUB3330
1230 A=2:PO=0:GOSUB1310:IFYP>120THEN1300
1240 A=0:W=18
1250 FORR=16TO37STEP7:S$=F$(5,A):Q$=F$(4,A):GOSUB2470:A=A+1:NEXT
1260 EP=9:A=4:PO=0:GOSUB1370:WC=1
1270 IFWC=1THENR=18:S=24:ELSER=0:S=6
1280 B=40:C=0:GOSUB3330:B=14:R=9:S=9:GOSUB3330
1290 B=5:R=10:S=14:GOSUB3330:B=16:R=15:S=R:GOSUB3330
1300 RETURN
1310 IFF$(A,4)="J"THENPO=PO+2
1320 B=1:Q$=F$(A+1,0)
1330 FORC=1TO4
1340 IFF$(A+1,C)=Q$THENB=B+1
1350 NEXT
1360 IFB=4ANDF$(A+1,4)<>Q$THENPO=PO+4:ELSEIFB=5THENPO=PO+5
1370 FORB=0TO3
1380 IFF$(A,B)="J"ANDF$(A+1,B)=F$(A+1,4)THENPO=PO+1:B=4
1390 NEXT
1400 FORB=0TO4:R%(0,B)=V%(A,B):R%(1,B)=V%(A+1,B):NEXT
1410 GOSUB1600
1420 PO=PO+T:GOTO2710
1430 IFF$(0,4)="J"THENPO=PO+2
1440 B=1:Q$=F$(A+1,0)
1450 FORC=1TO4
1460 IFF$(A,C)=Q$THENB=B+1
1470 NEXT
1480 IFB=4ANDF$(A+1,4)<>Q$THENPO=PO+4:ELSEIFB=5THENPO=PO+5
1490 FORB=0TO3
1500 IFF$(A,B)="J"ANDF$(A+1,B)=F$(A+1,4)THENPO=PO+1
1510 NEXT
1520 FORB=0TO4:R%(0,B)=V%(A,B):R%(1,B)=V%(A+1,B):NEXT
1530 GOSUB1600:PO=PO+T
1540 IFPO=0THENQ$="NO POINTS "+S$:GOTO1560
1550 Q$=STR$(PO):IFPO>1THENQ$=Q$+" POINTS "+S$:ELSEQ$=Q$+" POINT "+S$
1560 CHAR,7,12,Q$
1570 FORA=1TO2000:NEXT
1580 IFPO>0THENSOUND1,430,13:MP=MP+PO:R=36:S=9:P=MP:GOSUB3350
```

```
1620 IFA=5THENT=5:GOTO1750
1630 IFR%(0,0)+1=R%(0,1)ANDR%(0,1)+1=R%(0,2)ANDR%(0,2)+1=R%(0,3)THENT=T+4
1640 IFR%(0,0)+1=R%(0,1)ANDR%(0,1)+1=R%(0,2)ANDR%(0,2)+1=R%(0,4)THENT=T+4
1650 IFR%(0,0)+1=R%(0,1)ANDR%(0,1)+1=R%(0,3)ANDR%(0,3)+1=R%(0,4)THENT=T+4
1660 IFR%(0,0)+1=R%(0,2)ANDR%(0,2)+1=R%(0,3)ANDR%(0,3)+1=R%(0,4)THENT=T+4
1670 IFR%(0,1)+1=R%(0,2)ANDR%(0,2)+1=R%(0,3)ANDR%(0,3)+1=R%(0,4)THENT=T+4
1680 IFT=4THEN1750:ELSEIFT>4THEN1720
1690 FORB=0TOTP-2:FORC=B+1TOTP-1:FORD=C+1TOTP
1700 IFR%(0,B)+1=R%(0,C)ANDR%(0,C)+1=R%(0,D)THENT=T+3
1710 NEXTD,C,B
1720 FORA=0TOTP-1:FORB=A+1TOTP
1730 IFR%(0,A)=R%(0,B)THENT=T+2
1740 NEXTB,A
1750 A=0
1760 FORB=0TOTP:A=A+R%(1,B):NEXT
1770 IFA=15THENT=T+2:GOTO1890:ELSEIFA<15THEN1890
1780 IFTP<4THEN1830
1790 FORA=0TO1:FORB=A+1TO2
1800 FORC=B+1TO3:FORD=C+1TO4
1810 IFR%(1,A)+R%(1,B)+R%(1,C)+R%(1,D)=15THENT=T+2
1820 NEXTD,C,B,A
1830 FORA=0TOTP-2:FORB=A+1TOTP-1:FORC=B+1TOTP
1840 IFR%(1,A)+R%(1,B)+R%(1,C)=15THENT=T+2
1850 NEXTC,B,A
1860 FORA=0TOTP-1:FORB=A+1TOTP
1870 IFR%(1,A)+R%(1,B)=15THENT=T+2
1880 NEXTB,A
1890 RETURN
1900 CHAR,10,12,"          RUN UP...":FORA=1TO1645:NEXT:CHAR,18,12,"                    "
1910 CHAR,30,12,"TOTAL:"
1920 EP=0:T=0:MG=0:YG=0:IC=1:YC=1:TP=0:ML=WC
1930 R=37:S=12:P=T:GOSUB 3350:C=0
1940 R=18:S=R:Q$=" ┌───┐":GOSUB3340:R=19:S=21:Q$="|   |":GOSUB3340
1950 R=22:S=R:Q$=" └───┘":GOSUB3340:R=1:S=20:P=0:GOSUB3350
1960 R=23:S=R:Q$="PTS.":GOSUB3340:C=6:R=18:S=R:Q$="CAN'T":GOSUB3340
1970 C=7:R=19:S=R:Q$="GO":GOSUB3340
1980 DO
1990 PO=0
2000 IFML=1ANDYC=1THENGOSUB2540:GOTO2060
2010 IFML=2ANDIC=1THENGOSUB2120:GOTO2060
2020 IFML=1ANDYC=0ANDIC=1THENGOSUB2120:GOTO2060
2030 IFML=2ANDIC=0ANDYC=1THENGOSUB2540:GOTO2060
2040 IFIC=0ANDYC=0ANDEP=0THENPO=1:ONMLGOSUB2260,2710
2050 EP=0:T=0:TP=0:IC=1:YC=1:CHAR,37,12,"  ":P=T:R=37:S=12:GOSUB 3350
2060 LOOPUNTILMG+YG=8ORMP>120ORYP>120
2070 C=6:R=18:S=19:B=5:GOSUB3330
2080 C=16:R=7:S=7:B=23:GOSUB3330
2090 R=16:S=R:GOSUB3330
2100 B=32:C=7:R=12:S=R:GOSUB3330
2110 RETURN
2120 IFMG=4THEN2450
2130 IFT>4ANDT<15THEN2310:ELSEIFT>20THEN2360
2140 DO:A=FNA(4)-1:LOOPUNTILM%(A)=0
2150 W=0:S$=F$(1,A):Q$=F$(0,A)
2160 R=(A+2)*5+(A*2)+6:GOSUB2470:S=7:P=MG+1:GOSUB3350
2170 TP=TP+1:MG=MG+1:M%(A)=1
2180 T=T+V%(1,A):R=37:S=12:P=T:GOSUB3350
2190 R%(1,TP)=V%(0,A)
2200 ML=1
2210 IFMG+YG=8OR(MG=4ANDYC=0)THENPO=PO+1:IC=0:YC=0:EP=1
2220 IFT=15THENPO=PO+2
2230 IFT=31THENPO=2:IC=0:YC=0:EP=1
2240 IFTP>1THENGOSUB2970:ELSEIFPO=0THEN2290
2250 IFPO=0THENCHAR,8,12,"NO POINTS FOR ME":GOTO2290
2260 IFPO>0THENSOUND1,330,18:MP=MP+PO:P=MP:R=36:S=3:GOSUB3350
2270 Q$=STR$(PO):IFPO=1THENQ$=Q$+" POINT FOR ME":ELSEQ$=Q$+" POINTS FOR ME"
2280 CHAR,7,12,Q$
2290 FORA=1TO1645:NEXT
1600 B=0:GOSUB3240:T=0:A=1
```

```
2300 RETURN
2310 A=5:B=3
2320 DOWHILEB>=0
2330 IFT+V%(1,B)=15ANDM%(B)=0THENA=B:EXIT:ELSEB=B-1
2340 LOOP
2350 IFA<5ANDRND(1)>.5THEN2150:ELSE2140
2360 A=5:B=0:C=5
2370 DOWHILEB<4
2380 IFT+V%(1,B)=31ANDM%(B)=0THENA=B:EXIT
2390 IFT+V%(1,B)<32ANDM%(B)=0THENC=B
2400 B=B+1
2410 LOOP
2420 IFA<5THEN2150:ELSEIFC<5THENA=C:GOTO2150
2430 CHAR,8,12,"I CAN'T GO       "
2440 FORA=1TO1645:NEXT
2450 IC=0:IFYC=0ANDML=1ANDEP=0THENPO=1:EP=1:GOTO2260
2460 RETURN
2470 COLOR0,2
2480 CHAR,R-2,W,"┌─────┐":CHAR,R-2,W+6,"└─────┘"
2490 FORY=W+1TOW+5:CHAR,R-2,Y,"|     |":NEXT
2500 IFS$="♥"ORS$="♦"THENCOLOR1,3,5
2510 CHAR,R-1,W+1,S$:CHAR,R+1,W+5,S$:CHAR,R,W+3,Q$
2520 IFS$="♥"ORS$="♦"THENCOLOR1,1
2530 COLOR0,6,5:RETURN
2540 IFYG=4THEN2950
2550 CHAR,8,12,"YOU TO PLAY       "
2560 Q$="Y":X=9:Y=17:C=9:D=37:S=7
2570 GOSUB4630:IFX=9THEN2900:ELSEC=(X-2)/7-2
2580 IFY%(C)<>0THEN2890
2590 IFT+V%(3,C)>31THEN2880
2600 R=X:S=16:P=YG+1:GOSUB3350
2610 TP=TP+1:Y%(C)=1:YG=YG+1
2620 R%(1,TP)=V%(2,C)
2630 T=T+V%(3,C):R=37:S=12:P=T:GOSUB3350
2640 ML=2
2650 IFYG+MG=8OR(YG=4ANDIC=0)THENPO=PO+1:EP=1
2660 IFT=15THENPO=PO+2
2670 IFT=31THENPO=2:IC=0:YC=0:EP=1
2680 IFTP>1THENGOSUB2970
2690 IFTP>1ORPO>0THENGOSUB2710
2700 RETURN
2710 CHAR,8,12,"ENTER NO. OF POINTS ":IFEP=9THENCHAR,28,12,"IN CRIB"
2720 F=99:R=1:S=20:P=0:GOSUB4630:F=0
2730 CHAR,8,12,"                            ":IFEP=9THENCHAR,28,12,"         "
2740 IFP=POTHEN2830:ELSEIFP>POTHEN2820
2750 R=PO-P:Q$="I CLAIM"+STR$(R)+" ":IFR>1THENQ$=Q$+"POINTS":ELSEQ$=Q$+"POINT"
2760 CHAR,8,12,Q$
2770 FORA=1TO1645:NEXT
2780 CHAR,8,12,"                    "
2790 YP=YP+P:MP=MP+R
2800 P=MP:R=36:S=9:GOSUB3350:SOUND1,930,18:IFMP>120THENYP=0:RETURN
2810 GOTO2840
2820 CHAR,8,12,"I MAKE IT LESS!":SOUND1,20,70:FORA=1TO1645:NEXT:GOTO2710
2830 YP=YP+PO
2840 IFPO>0THENSOUND1,930,18:R=36:S=15:P=YP:GOSUB3350
2850 FORA=1TO500:NEXT
2860 CHAR,1,20,"0 "
2870 RETURN
2880 CHAR,8,12,"TOTAL OVER 31!":FORA=1TO1645:NEXT:GOTO2550
2890 CHAR,8,12,"THAT'S CHEATING!!":FORA=1TO1645:NEXT:GOTO2550
2900 A=0:B=5:DOWHILEA<4
2910 IFY%(A)=0ANDT+V%(3,A)<32THENB=A:EXIT:ELSEA=A+1
2920 LOOP
2930 IFB=5THEN2950
2940 CHAR,8,12,"OH YES YOU CAN!!":FORA=1TO1645:NEXT:GOTO2550
2950 YC=0:IFIC=0ANDML=2ANDEP=0THENPO=1:EP=1:GOTO2710
2960 RETURN
2970 IFTP=2THEN3140
2980 B=1:GOSUB3230:L=0:D=1
```

```
3010 IFR%(0,C)<>R%(0,C+1)THEND=D+1
3020 IFD>2THENL=D
3030 NEXT
3040 IFL=TPTHEND=L:GOTO3130
3050 IFL=0THEN3140
3060 DO
3070 B=TP-L+1:GOSUB3230:D=1
3080 FORC=BTOTP-1
3090 IFR%(0,C)+1=R%(0,C+1)THEND=D+1
3100 NEXT
3110 IFD>2ANDD<>LTHENL=D:GOTO3070
3120 LOOPUNTILD<3ORD=LORB>TP-2
3130 IFD=LTHENPO=PO+L:RETURN
3140 L=1:C=TP:B=TP-1
3150 DO
3160 IFR%(1,B)=R%(1,C)THENL=L+1:B=B-1
3170 LOOPUNTILB<1ORB<TP-3ORR%(1,B)<>R%(1,C)
3180 IFL=1THEN3220
3190 IFL=2THENPO=PO+2
3200 IFL=3THENPO=PO+6
3210 IFL=4THENPO=PO+12
3220 RETURN
3230 FORC=BTOTP:R%(0,C)=R%(1,C):NEXT
3240 DO
3250 D=0
3260 FORC=BTOTP-1
3270 IFR%(0,C)<=R%(0,C+1)THEN3300
3280 R=R%(0,C):R%(0,C)=R%(0,C+1)
3290 R%(0,C+1)=R:D=D+1
3300 NEXTC
3310 LOOPUNTILD=0
3320 RETURN
3330 Q$="":FORA=1TOB:Q$=Q$+"  ":NEXT
3340 FORA=RTO5:CHAR,C,A,Q$:NEXT:RETURN
3350 CHAR,R,5,RIGHT$(STR$(P),LEN(STR$(P))-1):RETURN
3360 DIM F$(5,4),C$(3,5),C%(3,5),R%(1,8),M%(3),Y%(3)
3370 DIM V%(5,4),CA$(3,13),SD$(1,12),P%(1,12):VOL7
3380 CA$(0,0)="♠":CA$(1,0)="♥"
3390 CA$(2,0)="♣":CA$(3,0)="♦"
3400 FORA=0TO3
3410 RESTORE4870
3420 FORB=1TO13
3430 READCA$(A,B)
3440 NEXTB,A
3450 FORA=0TO3:Y%(A)=0:M%(A)=0:NEXT
3460 RETURN
3470 FORC=0TO12
3480 DO:A=INT(RND(1)*4):B=INT(RND(1)*13)+1:LOOPUNTILCA$(A,B)<>""
3490 SD$(0,C)=CA$(A,B):SD$(1,C)=CA$(A,0)
3500 IFB<10THENP%(1,C)=B:ELSEP%(1,C)=10
3510 P%(0,C)=B:CA$(A,B)=""
3520 NEXT
3530 GOSUB3400:A=0
3540 IFWC=1THEN3630
3550 FORC=0TO10STEP2
3560 C$(0,A)=SD$(0,C):C$(1,A)=SD$(1,C)
3570 C%(0,A)=P%(0,C):C%(1,A)=P%(1,C)
3580 C$(2,A)=SD$(0,C+1):C$(3,A)=SD$(1,C+1)
3590 C%(2,A)=P%(0,C+1):C%(3,A)=P%(1,C+1)
3600 A=A+1
3610 NEXT
3620 GOTO3700
3630 FORC=0TO10STEP2
3640 C$(2,A)=SD$(0,C):C$(3,A)=SD$(1,C)
3650 C%(2,A)=P%(0,C):C%(3,A)=P%(1,C)
3660 C$(0,A)=SD$(0,C+1):C$(1,A)=SD$(1,C+1)
3670 C%(0,A)=P%(0,C+1):C%(1,A)=P%(1,C+1)
3680 A=A+1
3690 NEXT
```

```
3700 RETURN
3710 CHAR,6,10,"┌──────────────────────────────────────┐"
3720 FORY=11TO13:CHAR,6,Y,"│                                      │":NEXT
3730 CHAR,6,14,"└──────────────────────────────────────┘"
3740 CHAR,31,9,"ME  :"+STR$(MP)
3750 CHAR,30,15,"YOU  :"+STR$(YP)
3760 RETURN
3770 A=0:W=18
3780 FORR=2TO37STEP7
3790 Q$=C$(2,A):S$=C$(3,A):GOSUB2470
3800 FORY=0TO6:CHAR,R-2,Y,"▒▒▒▒▒▒":NEXT
3810 A=A+1
3820 NEXT
3830 FORY=9TO15:CHAR,0,Y,"▒▒▒▒▒▒":NEXT
3840 IFWC=1THENCHAR,7,9,"MY CRIB":ELSECHAR,7,15,"YOUR CRIB"
3850 RETURN
3860 CHAR,10,12,"DISCARD TWO CARDS          ":L=0:P=2
3870 Q$="Y":X=2:Y=17:C=2:D=37:S=7
3880 GOSUB4630:C=(X-2)/7
3890 IFC$(2,C)<>""THEN3920
3900 SOUND1,20,70:CHAR,8,12,"YOU CAN'T DISCARD IT TWICE"
3910 FORR=1TO1645:NEXT:CHAR,8,12,"                          ":GOTO3870
3920 F$(4,P)=C$(2,C):F$(5,P)=C$(3,C):C$(2,C)="":SOUND 1,930,12
3930 V%(4,P)=C%(2,C):V%(5,P)=C%(3,C)
3940 L=L+1:P=P+1:IFL<2THEN3870
3950 A=0
3960 FORB=0TO5
3970 IFC$(2,B)=""THEN4010
3980 F$(2,A)=C$(2,B):F$(3,A)=C$(3,B)
3990 V%(2,A)=C%(2,B):V%(3,A)=C%(3,B)
4000 A=A+1
4010 NEXT
4020 B=14:R=18:S=24:C=0:GOSUB3330:W=18:A=0
4030 IFC$(2,0)=""ANDC$(2,1)=""THEN4050
4040 FORR=16TO37STEP7:Q$=F$(2,A):S$=F$(3,A):GOSUB2470:A=A+1:NEXT
4050 RETURN
4060 CHAR,10,12,"PLEASE WAIT I'M THINKING"
4070 B=0:TP=3:BT=0
4080 FORX=0TO2:FORY=X+1TO3
4090 FORP=Y+1TO4:FORL=P+1TO5
4100 R%(0,0)=C%(0,X):R%(0,1)=C%(0,Y):R%(0,2)=C%(0,P):R%(0,3)=C%(0,L)
4110 R%(1,0)=C%(1,X):R%(1,1)=C%(1,Y):R%(1,2)=C%(1,P):R%(1,3)=C%(1,L)
4120 GOSUB3240
4130 IFR%(0,0)+1=R%(0,1)ANDR%(0,1)+1=R%(0,2)ANDR%(0,2)+1=R%(0,3)THENT=4:ELSET=
4140 IFT>0THENGOSUB1750:ELSEGOSUB1690
4150 F=0:MG=1:FORPO=0TO5
4160 IFPO=XORPO=YORPO=PORPO=LTHEN4180
4170 IFMG=1THENR=PO:MG=2:ELSES=PO
4180 NEXTPO
4190 IFC%(0,R)=C%(0,S)THENF=2:GOTO4230
4200 IFABS(C%(0,R)-C%(0,S))<3THENF=1
4210 IFC%(1,R)+C%(1,S)=15THENF=F+2
4220 IFC%(1,R)+C%(1,S)=5THENF=F+1
4230 IFWC=1THENT=T+F:ELSET=T-F
4240 IFT>BTTHENBT=T:POKE208,R:POKE209,S
4250 NEXTL,P,Y,X
4260 IFBT=0THENR=0:S=1:ELSER=PEEK(208):S=PEEK(209)
4270 F$(4,0)=C$(0,R):F$(5,0)=C$(1,R)
4280 F$(4,1)=C$(0,S):F$(5,1)=C$(1,S)
4290 V%(4,0)=C%(0,R):V%(5,0)=C%(1,R)
4300 V%(4,1)=C%(0,S):V%(5,1)=C%(1,S)
4310 A=0:FORB=0TO5
4320 IFB=RORB=STHEN4360
4330 F$(0,A)=C$(0,B):F$(1,A)=C$(1,B)
4340 V%(0,A)=C%(0,B):V%(1,A)=C%(1,B)
4350 A=A+1
4360 NEXT
4370 B=10:C=0:R=0:S=6:GOSUB3330
4380 RETURN
```

```
4410 CHAR,1,1,Q$:CHAR,28,1,Q$
4420 CHAR,1,23,Q$:CHAR,28,23,Q$
4430 CHAR,19,1,"ME":CHAR,19,23,"YOU"
4440 DO:A=INT(RND(1)*4):B=INT(RND(1)*13)+1
4450 :C=INT(RND(1)*4):D=INT(RND(1)*13)+1:LOOPUNTILA<>CORB<>D
4460 R=20:W=4:Q$=CA$(A,B):S$=CA$(A,0):GOSUB2470
4470 W=14:Q$=CA$(C,D):S$=CA$(C,0):GOSUB2470
4480 IFB<DTHEN4500:ELSEIFB>DTHEN4510
4490 CHAR,19,12,"TIE":FORA=1TO3000:NEXT:CHAR,19,12,"    ":GOTO4440
4500 CHAR,17,12,"MY CRIB":WC=1:GOTO4520
4510 CHAR,16,12,"YOUR CRIB":WC=2
4520 GOSUB4720:SCNCLR:RETURN
4530 COLOR0,3,1:COLOR4,3,1:COLOR1,3,5:GRAPHIC0,1
4540 IFMP>120THENCHAR,7,6,"▮YOU HAVE LOST...OF COURSE!▦"
4550 IFYP>120THENCHAR,8,6,"▮BEATEN...AND BY A HUMAN!▦"
4560 GOSUB4720
4570 CHAR,8,12,"A N O T H E R   G A M E ?"
4580 CHAR,8,14,"YES                         NO"
4590 Q$="↑":X=9:Y=15:C=9:D=32:S=23
4600 GOSUB4630
4610 IFX=9THENSOUND1,930,12:RUN
4620 SOUND1,930,12:FORA=1TO250:NEXT:GRAPHIC0,1:NEW:END
4630 IFF<99THENCHAR,X,Y,Q$:ELSEGOSUB3350:IFP<10THENCHAR,2,20," "
4640 DO:A=JOY(2):LOOPUNTILA=3ORA=7ORA=128
4650 IFF<99THENCHAR,X,Y," "
4660 IFA=128THENRETURN
4670 IFA=3ANDF<99THENX=X+5:IFX>DTHENX=C
4680 IFA=7ANDF<99THENX=X-5:IFX<CTHENX=D
4690 IFA=3ANDF=99THENP=P+1:IFP>30THENP=0
4700 IFA=7ANDF=99THENP=P-1:IFP<0THENP=30
4710 GOTO4630
4720 RESTORE
4730 FORA=1TO76
4740 IFA=19ORA=37THENRESTORE4800
4750 IFA=27ORA=55THENFORB=1TO30:READC:NEXT
4760 READB,C,D:SOUND1,B,C:SOUND2,B-2,C
4770 FORP=1TOD:NEXT
4780 NEXT
4790 RETURN
4800 DATA739,12,1,755,12,1,770,12,1,864,22,1,770,12,1,864,22,1,770,12,1,864,42
4810 DATA245,864,12,1,881,12,1,889,12,1,897,12,1,864,12,1,881,12,1,897,22,1,854
4820 DATA12,1,881,22,1,864,67,1,834,12,1,810,12,1,798,12,1,834,12,1,864,12,1,897
4830 DATA26,1,881,12,1,854,12,1,834,12,1,881,52,1,864,12,1,881,12,1,889,12,1,897
4840 DATA12,1,864,12,1,881,12,1,897,22,1,864,12,1,881,22,1,897,12,1,864,12,1,881
4850 DATA12,1,897,22,1,864,12,1,881,22,1,897,12,1,864,12,1,881,12,1,897,22,1
4860 DATA854,12,1,881,22,1,864,98,1645
4870 DATAA,2,3,4,5,6,7,8,9,X,J,Q,K
```

```
******************************************** BLOOPING BUG *********************
********************************** by PETER and MELANIE CRACK ***************
* This month I will fill in some more missing routines. so, as before load
* both parts of the programme, then enter D41Ø6 and press enter and carry on
* from there. this is the explanation.
41Ø6-41ØF Every time the 'YOU' or 'BUGBUSTER' sprite is being moved and $D8
*          is unequal a collision has been detected between it and another
*          sprite. so this routine is activated to take the 'HITS NOT YET
*          ADDED' to your score pointer at $4E3C and increase it by one. then
*          checks this new number to make sure it does not exceed dec127 or
*          hex #$7F and then return it to $4E3C and return from gosub.
* The next routine evaluates the joystick return and sets the 'YOU' sprite
* pointers accordingly, first type in D4A32 and press return then position
* the cursor to the left of the first address ($4A32) and carry on.
4A32-4A33 Stop interrupts and switch out rom.
4A36-4A3A Load 'A' register with $E8 (joystick return register) and check to
*          see if it is equal.if not then branch to $4A57 else...........
4A3B-4A4B Increase each direction delay reload pointer in turn by one, at
*          the same time checking that they do not exceed three. these
*          pointers are located at $478Ø to $4783.
4A4D-4A5Ø Reduce speed pointer ($D6) by one and check to see if it has
*          reached #FF.dec-1 if yes then branch to $4A54 else store it in $D6
4A54       Jump to next routine at $4AD5.
4A57-4A5A If the programme goes to here then $E8 has some value. so, save a
*          copy on the stack. clear 'X' reg. discard the four leftmost bits
*          of $E8 ('A' reg. loaded with value of $E8 at $4A36).
4A5C-4A6F And compare it in turn, with the four main values, up,down,left
*          and right that is Ø1,Ø5,Ø7 and Ø3. of joystick returns, increasing
*          'X' register by one after each check, if any of the checks are
*          true then a branch is made to $4A73, the 'X' register well now
*          contain ØØ,Ø1,Ø2,Ø3 depending on which comparison was true, this
*          is why you cannot make a diagonal joystick move.
4A7Ø       If none of the above tests were true then jump to $4ACA.
4A73-4A7F Load 'A' register with direction delay register offset by 'X' reg.
*          if the value returned is zero then branch to $4A82 else decrease
*          it by one. store it back and jump to $4ACA.
4A82-4A8D Load the corresponding direction delay reload into 'A' register,
*          transfer it to 'Y' reg. decrease it by one and transfer it back
*          then store it into delay reload and delay registers.
4A9Ø-4A99 Load 'A' reg. with the value in $4788 offset by 'X' reg. (the four
*          addresses at $4788 to $478B contain the low byte of the four
*          movement routines). now we compare the 'X' reg. with #$ØØ and #$Ø2
*          to establish wether the joystick was pushed up or down if yes then
*          branch to $4ABØ else....................
4A9B-4AAD The joystick was pushed left or right!!. The 'A' reg. contains the
*          low byte of the address of the routine which will move thw sprite
*          in that direction, by comparing this value with the one in $43ØA
*          (which was the direction in which the sprite moved last) we can
*          detrmine wether or not the sprite and the joystick movement are in
*          the same direction. if the two values are the same we increase $D6
*          (speed reg.), and increase $DC (this is the delay reg. that
*          reduces the frequency of up or down moves, if you check the move
*          evaluation routine at $43ØØ to $432Ø, you will see that the sprite
*          will move left or right every time, but will only move up or down
*          after it has counted $DD to zero, $DC is the delay reload for this
*          action). then jump to $4ACA, if the two are not the same then we
*          simply store the 'A' register in $43ØA and jump to $4ACA (this
*          unfortunately results in some jerky movement if you hold the
*          joystick left and then quickly push it right).
4ACA-4AD1 Pull the original value of $E8 back off the stack clear the four
*          rightmost bits and check to see if it is zero if not (you have
*          pressed the fire button) branch to $4BØØ else..................
4AD5-4AD8 return to $4AD6 (main get sprite routine), the fact that I have
* used two commands the same, one after the other is the result of creating
* a programme at the keyboard and not writing it all out therefore planning
* it first. remember this routine is only carried out by 'you' sprite and
* only when the delay routine at $4A73 allows it. this results in a form of
* inertia but I admit it is a hapnazard.
```

PART 10/4

```
*****************************************************************************
4B00-4BFF This routine creates the 'YOU' sprite with the hole in the middle
*         and the 'SHOT' sprite, there are two 'YOU' sprites one before you
*         shoot and one after, the second one is only on the screen when the
*         'SHOT' sprite is also on the screen, the 'SHOT' sprite remains on
*         screen for sixteen moves.
4B00-4B04 Load 'A' reg. with the value stored in $12 (note the missing hash
*         sign this means it is an address not a number), this register is
*         to #$10 when you press the fire button and then counted down to
*         zero thus giving sixteen moves, compare it with zero if it is then
*         branch to $4B07 else jump to $4AD5.
4B07-4B0A Load 'A' reg. with $E5 (present sprite number), save a copy on
*         stack and gosub store sprite registers.
4B0D-4B1D Clear $D0 and $D4, store #$B5 (this is the page number where
*         'YOU' sprite with hole is stored) in $D1, pull a copy of present
*         sprite number off stack and push it back to keep the stack pointer
*         tidy, transfer it to 'Y' reg. load the present sprite 'original'
*         data page number into $D5 and gosub transfer new 'YOU' sprite data
*         into present 'YOU' sprite 'original' data page.
4B22-4B2C Load #$0F into $E5 ('SHOT' sprite number) transfer 'X' to 'A' reg.
*         and store it on stack, load #$B6 into $D1 and gosub create but do
*         not print 'SHOT' sprite.
4B2F-4B33 Pull 'SHOT' sprite number off stack and store it in $E5, do the
*         same for 'YOU' sprite and store it in $E4.
4B35      Gosub transfer 'YOU' sprite position and move pointers.
4B38      Gosub position 'SHOT' sprite on screen.
4B3B-4B41 Transfer 'SHOT' sprite speed reg. from static data to programme
*         data and jump to next part of programme.
4B48-4B56 Set data transfer pointers $D0,$D1,$D4,$D5.
4B58-4B63 Load 'X' reg. with #$01 and change 'YOU' and 'SHOT' sprite numbers
*         into data addresses (low byte in each case) to enable screen
*         address registers to be transfered from 'YOU' to 'SHOT' sprite.
4B65-4B67 Load 'X' reg. with #$02 and gosub data transfer (three bytes will
*         be exchanged) the last (speed reg.) will be discarded ('SHOT'
*         sprite always moves fast).
4B6A-4B73 Add #$05 to $D0, load 'X' reg. with #$03 and gosub data transfer
*         (four bytes will be exchanged from 'YOU' sprite to 'SHOT' sprite
*         these are, delay reload, delay, position x and position y reg.)
4B76-4BFF To this point we have positioned 'SHOT' and 'YOU' sprites top left
*         hand corners over each other, now we must position the 'SHOT'
*         sprite over the centre of the 'YOU' sprite, the following routine
*         works but is not very accurate and as I have improved it in my
*         sprite editor I do not think it worth an explanation, except to
*         say that the gosub at $4BDB prints and saves the 'SHOT'sprite data
4CB3-4CBF Creates a sprite without printing it on screen.
4CD0-4CE8 Load 'A' reg. with #$10 (dec 16 the number of moves the 'SHOT'
*         sprite gets before it dissapears off the screen)and save it in $12
*         load 'A' reg. with 'SHOT' sprite number and store it in $E5, gosub
*         get sprite data, stop interrupts, switch out ROM and jump back to
*         main routine, we have now created and positioned the 'SHOT' sprite
*         and as this will be the next sprite to be moved there is very
*         little delay between you pressing the fire button and a shot
*         appearing on screen (remember 'YOU' sprite number is #$0E and
*         'SHOT' sprite number is #$0F).
* And now we come to the final few routines that will complete the programme
* these are the the routines that evaluate the 'SHOT' sprite actions, remove
* and replace the 'BLOOPING BUG' sprites and the end routines.
4360-4362 Checks $D8 (collision register) and branches to $43D3 if there are
*         none.
4364-4375 Loads position accross the screen of first sprite.deducts #$04 and
*         compares it with $DE (position across screen of 'SHOT' sprite) if
*         the 'SHOT' sprite is to the left of this then branch to $4389
*         (goto next sprite), next add #$08 and again compare it with $DE.
*         if the 'SHOT' sprite is to the right of this new value then branch
*         to $4389
4377-4384 At this point we know that the shot sprite is within four pixel
*         points of one of the 'BLOOPING BUG' sprites (across the screen)
****************************************************** CONTINUED *******
```

```
*******************************************************************************
*         now we must establish if it is within four pixel points up or down
*         the screen, so we load the vertical reg. for this 'BLOOPING BUG'
*         sprite, deduct #$04 and check it against $DF (vertical reg. for
*         'SHOT' sprite), if $DF is lower than this (further up the screen)
*         then branch to $4389, else add #$08 and compare it with $DF again
*         if $DF is higher than this (further down the screen), then branch
*         to $4389, else gosub remove and replace this 'BLOOPING BUG' sprite
*         (centre of 'SHOT' sprite is within four pixel points of centre of
*         this 'BLOOPING BUG' sprite), and jump to next part of programme.
4389-438C Increase 'X' reg. by one, this reg. holds 'BLOOPINB BUG' sprite
*         number and compare it with #$0E (there can never be more than 14
*         'BLOOPING BUG' sprites on the screen at any one time #$00 to #$0D
*         #$0E='YOU' sprite and #$0F='SHOT' sprite),branch to $43D3 if equal
438E-4394 Transfer 'Y' to 'A' reg. add #$10 and transfer 'A' back to 'Y' reg
*         (this is simpler than increasing 'Y' reg sixteen times), this sets
*         the 'Y' reg. to point to the next set of registers and force
*         branch to $4368.
4264-42B3 This routine erases the 'BLOOPING BUG' sprite which has been hit.
4264-4269 Transfer 'X' to 'A' reg. shift all bits left four times, transfer
*         'A' to 'Y' reg, thus creating the right offset from the begining
*         of $4600 (page 46, which is sprite data area held in groups of
*         sixteen pointers, hence four shifts left, ASL*4 or multiply by 16)
426A-426D Store rotation pointer for this sprite in temp $33.
427A-427D Store sprite page number into temp reg $68.
427F-4281 Store 'SHOT' sprite number in 'A' reg. and save on stack.
4282-4286 Store 'BLOOPING BUG' which has been hit, sprite number in temp reg
*         $11, load 'Y' reg. with same and gosub erase this sprite.
4289-428B Pull 'SHOT' sprite number back off stack, transfer 'A' to 'Y' reg.
*         and gosub erase this sprite.
428E-4290 Pull last gosub address back off stack and discard, then jump to
*         next routine ($4D00).
4294-429F Stop interrupts, switch out ROM, load #$FF into sprite active list
*         store sprite number in 'Y' reg. and gosub get sprite pointers.
42A2-42B3 Load #$60 (hex code for RTS or return commandd) into $4532 ($4500
*         to $4531 is the erase part of the sprite erase/print routine at
*         $4500 to $45BB), gosub sprite erase, load #$A9 (original value)
*         back into $4532, switch in ROM, allow interrupts and return from
*         gosub.
4CF0-4D96 This routine sets two new sprites to replace the one just removed.
4CF0-4CF4 This is done when the sprite list is full, so pull the last two
*         return addresses off the stack and jump back to the main routine.
*         it is also done if the sprite just erased was the smallest
*         possible (a B4 sprite).
4D00-4D06 Increase $68 and load it into 'A' reg. compare it with #$B5 (there
*         are four 'BLOOPING BUG' sprites #$B0 is the largest 5*5 chars, B1,
*         B2.B3 are 4*4.3*3,2*2 chars and B4 is the smallest 1*1  B0 to B4
*         are the page numbers of the original sprite data areas, if the
*         sprite just erased was a B4 or smallest sprite then it will
*         disapear so branch to $4D79 else.................
4D08-4D09 Store a copy of 'A' reg. onto stack and gosub get new sprite no.
4D0C      Branch if carry clear to $41F0, this means there is no room in
*         sprite list for another pair of of sprites so go back to main
*         routine, if the carry is set there is room and 'A' reg. holds new
*         sprite number.
4D0E-4D15 Store 'A' reg. (sprite number) in $E5 and save a copy on stack,
*         store new sprite page number in $D1 and gosub create new sprite
*         without printing it.
4D18-4D76 Load old sprite number into $E4 and save a copy on stack, now
*         transfer all data to the two newly created sprites and move them
*         into the correct positions on the screen but do not print them.
4D79-4D7D Stop interrupts. switch out ROM and jump to main sprite routine.
4D80-4D96 Checks through sprite active list, returns first unused number
*         in 'A' reg. clearing the carry flag if it finds one and setting
*         carry flag if not.
************************************************** CONTINUED ************
```

```
****************************************************************************
41C6-41EA This routine checks to see if 'SHOT' sprite is on screen and if it
*         should be removed.
41C6-41CF Load 'Y' reg. with $12 ('SHOT' sprite number of turns on screen
*         reg.) is it minus?, if yes then 'SHOT' sprite is not on screen so
*         branch to $41D1, else decrease 'Y' reg. by one an check it again
*         branch to $41D1 if it is now minus, else store 'Y' reg. in $12 and
*         check to see if it is zero, if yes then branch to $41D4 else jump
*         to $41E7.
41D4-41DF Load 'X' reg. with 'YOU' sprite number and store 'X' reg. in $E5,
*         (the present (YOU' sprite on screen is the one with the hole in
*         the middle and a 'SHOT' sprite is running), load 'A' reg. with
*         #$B7 that is the data page number which contains the 'SHOT' and
*         'YOU' sprite comdined, store it in $D1, the next two gosubs
*         transfer this data to 'YOU' sprite area.
41E2-41EA Load 'Y' reg. with 'SHOT' sprite number and gosub erase sprite,
*         gosub check end of game and return from this gosub.
40EA-40FF This routine is only carried out at the end of play, clear $83
*         gosub return to lo-res. reset interrupt vectors, reset stack
*         pointer to #$FF and break programme.
4F50-4F73 Checks each part of score counter to ensure it has all been added
*         to the display, $4E38 is the hundreds counter, $4E42 is the tens
*         counter, $4E3C is the units counter and $4E3B is the lines counter
*         for the units display and if not zero it means that the last
*         number is not yet fully in the display, if all the above registers
*         are zero then the score display is up to date, so the routine then
*         checks to see if any 'BLOOPING BUG' sprites are still active if
*         yes then return from gosub, else..................
4F74-4F90 Load 'Y' reg. with 'YOU' sprite number and gosub sprite erase,
*         load 'X' reg. with zero and store in $E5, load 'A' reg. with #$C2
*         (page number where last sprite data is to be found), store 'A' reg
*         in $D1 and gosub create and print sprite, reset interrupt vectors
*         to point to $4FD0.
4FD0-4FE4 This routine checks the keyboard at each interrupt for a 'Y' or an
*         'N' (capitals) key return (meaning either yes Iwant to try again
*         or no I do not), if no keypress is made the  programme jumps to
*         $CE0E. a 'N' keypress sends the programme to $40EA (end of game
*         routine). a'Y' keypress sends the programme to $4FE1. there it
*         loads 'X' reg. with #$FF, transfers it to stack pointer thus
*         reseting it and jumps to $4000 (start of game without instrucions)
* Once you have typed in the last part of this programme, save both parts
* and then 'RUN' the game with G5100 and press return, first you will see
* the instructions, follow these, and be amazed at what your +4 is capable.
* and to those who had the patience to type it all in my THANKS.
* As always any problems or ommisions on my part let me know or write in to
* the magazine. And now for next issue I have a new programme called
* 'LUNAR LANDER' yes folks, this is the 'almost as good as the coi-op'
* version that ATARI brought out in 1980 so slowly but surely we are
* catching them up!!!!!....................PETER and MELANIE CRACK..........
******************** call me on 081-367-3152 with our problems.*************
****************************************************************************
```

```
*******************************************************************************************
. 41Ø6  AE 3C 4E LDX $4E3C
. 41Ø9  E8       INX
. 41ØA  3Ø Ø3    BMI $41ØF
. 41ØC  8E 3C 4E STX $4E3C
. 41ØF  6Ø       RTS
. 411Ø  EA       NOP

. 4A32  EA       NOP
. 4A33  8D 3F FF STA $FF3F
. 4A36  A5 E8    LDA $E8
. 4A38  DØ 1D    BNE $4A57
. 4A3A  EA       NOP
. 4A3B  A2 Ø3    LDX #$Ø3
. 4A3D  BD 8Ø 47 LDA $478Ø.X
. 4A4Ø  A8       TAY
. 4A41  C8       INY
. 4A42  CØ Ø4    CPY #$Ø4
. 4A44  BØ Ø4    BCS $4A4A
. 4A46  98       TYA
. 4A47  9D 8Ø 47 STA $478Ø.X
. 4A4A  CA       DEX
. 4A4B  1Ø FØ    BPL $4A3D
. 4A4D  A4 D6    LDY $D6
. 4A4F  88       DEY
. 4A5Ø  3Ø Ø2    BMI $4A54
. 4A52  84 D6    STY $D6
. 4A54  4C D5 4A JMP $4AD5
. 4A57  48       PHA
. 4A58  A2 ØØ    LDX #$ØØ
. 4A5A  29 ØF    AND #$ØF
. 4A5C  C9 Ø1    CMP #$Ø1
. 4A5E  FØ 13    BEQ $4A73
. 4A6Ø  E8       INX
. 4A61  C9 Ø3    CMP #$Ø3
. 4A63  FØ ØE    BEQ $4A73
. 4A65  E8       INX
. 4A66  C9 Ø5    CMP #$Ø5
. 4A68  FØ Ø9    BEQ $4A73
. 4A6A  E8       INX
. 4A6B  C9 Ø7    CMP #$Ø7
. 4A6D  FØ Ø4    BEQ $4A73
. 4A6F  EA       NOP
. 4A7Ø  4C CA 4A JMP $4ACA
. 4A73  EA       NOP
. 4A74  BD 9Ø 47 LDA $479Ø.X
. 4A77  FØ Ø9    BEQ $4A82
. 4A79  A8       TAY
. 4A7A  88       DEY
. 4A7B  98       TYA
. 4A7C  9D 9Ø 47 STA $479Ø.X
. 4A7F  4C CA 4A JMP $4ACA
. 4A82  BD 8Ø 47 LDA $478Ø.X
. 4A85  FØ Ø6    BEQ $4A8D
. 4A87  A8       TAY
. 4A88  88       DEY
. 4A89  98       TYA
. 4A8A  9D 8Ø 47 STA $478Ø.X
. 4A8D  9D 9Ø 47 STA $479Ø.X
. 4A9Ø  BD 88 47 LDA $4788.X
. 4A93  EØ Ø2    CPX #$Ø2
. 4A95  FØ 19    BEQ $4ABØ
. 4A97  EØ ØØ    CPX #$ØØ
. 4A99  FØ 15    BEQ $4ABØ
. 4A9B  CD ØA 43 CMP $43ØA
. 4A9E  DØ ØA    BNE $4AAA
. 4AAØ  8D ØA 43 STA $43ØA
```

```
. 4AA3  E6 D6    INC $D6
. 4AA5  E6 DC    INC $DC
. 4AA7  4C CA 4A JMP $4ACA
. 4AAA  8D ØA 43 STA $43ØA
. 4AAD  4C CA 4A JMP $4ACA
. 4ABØ  CD 18 43 CMP $4318
. 4AB3  DØ 12    BNE $4AC7
. 4AB5  8D 18 43 STA $4318
. 4AB8  E6 D6    INC $D6
. 4ABA  A4 DC    LDY $DC
. 4ABC  88       DEY
. 4ABD  3Ø Ø5    BMI $4AC4
. 4ABF  84 DC    STY $DC
. 4AC1  EA       NOP
. 4AC2  EA       NOP
. 4AC3  EA       NOP
. 4AC4  4C CA 4A JMP $4ACA
. 4AC7  8D 18 43 STA $4318
. 4ACA  68       PLA
. 4ACB  29 FØ    AND #$FØ
. 4ACD  EA       NOP
. 4ACE  EA       NOP
. 4ACF  EA       NOP
. 4ADØ  EA       NOP
. 4AD1  DØ 2D    BNE $4BØØ
. 4AD3  DØ Ø3    BNE $4AD8
. 4AD5  4C E6 42 JMP $42E6
. 4AD8  4C E6 42 JMP $42E6
. 4ADB  4C ØØ 4A JMP $4AØØ
. 4ADE  ØØ       BRK
. 4ADF  ØØ       BRK
. 4AEØ  48       PHA
. 4AE1  A4 E5    LDY $E5
. 4AE3  B9 Ø3 46 LDA $46Ø3.Y
. 4AE6  85 5A    STA $5A
. 4AE8  68       PLA
. 4AE9  4C FØ 41 JMP $41FØ
. 4AEC  FF       ???
. 4AED  FF       ???

. 4BØØ  A5 12    LDA $12
. 4BØ2  FØ Ø3    BEQ $4BØ7
. 4BØ4  4C D5 4A JMP $4AD5
. 4BØ7  A5 E5    LDA $E5
. 4BØ9  48       PHA
. 4BØA  2Ø AØ 41 JSR $41AØ
. 4BØD  A9 ØØ    LDA #$ØØ
. 4BØF  85 DØ    STA $DØ
. 4B11  85 D4    STA $D4
. 4B13  A9 B5    LDA #$B5
. 4B15  85 D1    STA $D1
. 4B17  68       PLA
. 4B18  48       PHA
. 4B19  A8       TAY
. 4B1A  B9 DØ 49 LDA $49DØ.Y
. 4B1D  85 D5    STA $D5
. 4B1F  2Ø 28 4C JSR $4C28
. 4B22  A2 ØF    LDX #$ØF
. 4B24  86 E5    STX $E5
. 4B26  8A       TXA
. 4B27  48       PHA
. 4B28  A9 B6    LDA #$B6
. 4B2A  85 D1    STA $D1
. 4B2C  2Ø B3 4C JSR $4CB3
. 4B2F  68       PLA
. 4B3Ø  85 E5    STA $E5
. 4B32  68        PLA
. 4B33  85 E4     STA $E4
. 4B35  20 48 4B  JSR $4B48
. 4B38  20 98 4B  JSR $4B98
. 4B3B  AD F2 48  LDA $48F2
. 4B3E  8D F2 46  STA $46F2
. 4B41  4C D0 4C  JMP $4CD0
. 4B44  4C D0 4C  JMP $4CD0
. 4B47  EA        NOP
. 4B48  A9 00     LDA #$00
. 4B4A  85 D0     STA $D0
. 4B4C  A9 0A     LDA #$0A
. 4B4E  85 D4     STA $D4
. 4B50  A9 46     LDA #$46
. 4B52  85 D1     STA $D1
. 4B54  A9 48     LDA #$48
. 4B56  85 D5     STA $D5
. 4B58  A2 01     LDX #$01
. 4B5A  B5 E4     LDA $E4,X
. 4B5C  0A        ASL
. 4B5D  0A        ASL
. 4B5E  0A        ASL
. 4B5F  0A        ASL
. 4B60  95 E4     STA $E4,X
. 4B62  CA        DEX
. 4B63  10 F5     BPL $4B5A
. 4B65  A2 02     LDX #$02
. 4B67  20 E0 4B  JSR $4BE0
. 4B6A  A5 D0     LDA $D0
. 4B6C  18        CLC
. 4B6D  69 05     ADC #$05
. 4B6F  85 D0     STA $D0
. 4B71  A2 03     LDX #$03
. 4B73  20 E0 4B  JSR $4BE0
. 4B76  A2 01     LDX #$01
. 4B78  A5 E5     LDA $E5
. 4B7A  20 C5 4B  JSR $4BC5
. 4B7D  85 E0     STA $E0
. 4B7F  A5 E4     LDA $E4
. 4B81  20 C5 4B  JSR $4BC5
. 4B84  38        SEC
. 4B85  E5 E0     SBC $E0
. 4B87  95 64     STA $64,X
. 4B89  E6 D4     INC $D4
. 4B8B  CA        DEX
. 4B8C  10 EA     BPL $4B78
. 4B8E  E6 D0     INC $D0
. 4B90  E6 D0     INC $D0
. 4B92  A2 01     LDX #$01
. 4B94  20 E0 4B  JSR $4BE0
. 4B97  60        RTS
. 4B98  A9 0F     LDA #$0F
. 4B9A  85 E5     STA $E5
. 4B9C  78        SEI
. 4B9D  8D 3F FF  STA $FF3F
. 4BA0  20 E0 4A  JSR $4AE0
. 4BA3  20 3F 44  JSR $443F
. 4BA6  A4 64     LDY $64
. 4BA8  98        TYA
. 4BA9  48        PHA
. 4BAA  20 D8 44  JSR $44D8
. 4BAD  68        PLA
. 4BAE  A8        TAY
. 4BAF  88        DEY
. 4BB0  D0 F6     BNE $4BA8
. 4BB2  A4 65     LDY $65
```

```
. 4BB4  98        TYA
. 4BB5  48        PHA
. 4BB6  20 00 44  JSR $4400
. 4BB9  68        PLA
. 4BBA  A8        TAY
. 4BBB  88        DEY
. 4BBC  D0 F6     BNE $4BB4
. 4BBE  20 AE 45  JSR $45AE
. 4BC1  20 EE 4B  JSR $4BEE
. 4BC4  60        RTS
. 4BC5  18        CLC
. 4BC6  69 01     ADC #$01
. 4BC8  A8        TAY
. 4BC9  B1 D0     LDA ($D0),Y
. 4BCB  0A        ASL
. 4BCC  0A        ASL
. 4BCD  0A        ASL
. 4BCE  0A        ASL
. 4BCF  A8        TAY
. 4BD0  B1 D4     LDA ($D4),Y
. 4BD2  60        RTS
. 4BD3  EA        NOP
. 4BD4  A5 DE     LDA $DE
. 4BD6  38        SEC
. 4BD7  E5 33     SBC $33
. 4BD9  85 DE     STA $DE
. 4BDB  20 53 43  JSR $4353
. 4BDE  60        RTS
. 4BDF  EA        NOP
. 4BE0  A4 E4     LDY $E4
. 4BE2  B1 D0     LDA ($D0),Y
. 4BE4  A4 E5     LDY $E5
. 4BE6  91 D0     STA ($D0),Y
. 4BE8  E6 D0     INC $D0
. 4BEA  CA        DEX
. 4BEB  10 F3     BPL $4BE0
. 4BED  60        RTS
. 4BEE  A5 DE     LDA $DE
. 4BF0  38        SEC
. 4BF1  E5 65     SBC $65
. 4BF3  85 DE     STA $DE
. 4BF5  A5 DF     LDA $DF
. 4BF7  38        SEC
. 4BF8  E5 64     SBC $64
. 4BFA  85 DF     STA $DF
. 4BFC  4C D4 4B  JMP $4BD4
. 4BFF  60        RTS

. 4CB3  20 00 4C  JSR $4C00
. 4CB6  20 08 4C  JSR $4C08
. 4CB9  20 28 4C  JSR $4C28
. 4CBC  20 40 4C  JSR $4C40
. 4CBF  60        RTS
. 4CC0  EA        NOP

. 4CD0  A9 10     LDA #$10
. 4CD2  85 12     STA $12
. 4CD4  A9 0F     LDA #$0F
. 4CD6  85 E5     STA $E5
. 4CD8  20 F0 41  JSR $41F0
. 4CDB  EA        NOP
. 4CDC  EA        NOP
. 4CDD  EA        NOP
. 4CDE  EA        NOP
. 4CDF  EA        NOP
. 4CE0  EA        NOP
. 4CE1  78        SEI
. 4CE2  8D 3F FF  STA $FF3F
. 4CE5  4C E6 42  JMP $42E6
. 4CE8  EA        NOP

. 436Ø  A5 D8     LDA $D8
. 4362  FØ 6F     BEQ $43D3
. 4364  A2 ØØ     LDX #$ØØ
. 4366  AØ ØØ     LDY #$ØØ
. 4368  B9 ØA 46  LDA $46ØA.Y
. 436B  E9 Ø4     SBC #$Ø4
. 436D  C5 DE     CMP $DE
. 436F  BØ 18     BCS $4389
. 4371  69 Ø8     ADC #$Ø8
. 4373  C5 DE     CMP $DE
. 4375  9Ø 12     BCC $4389
. 4377  B9 ØB 46  LDA $46ØB.Y
. 437A  E9 Ø4     SBC #$Ø4
. 437C  C5 DF     CMP $DF
. 437E  BØ Ø9     BCS $4389
. 438Ø  69 Ø8     ADC #$Ø8
. 4382  C5 DF     CMP $DF
. 4384  9Ø Ø3     BCC $4389
. 4386  4C 64 42  JMP $4264
. 4389  E8        INX
. 438A  EØ ØE     CPX #$ØE
. 438C  FØ 45     BEQ $43D3
. 438E  98        TYA
. 438F  18        CLC
. 439Ø  69 1Ø     ADC #$1Ø
. 4392  A8        TAY
. 4393  38        SEC
. 4394  BØ D2     BCS $4368
. 4396  EA        NOP

. 4263  EA        NOP
. 4264  8A        TXA
. 4265  ØA        ASL
. 4266  ØA        ASL
. 4267  ØA        ASL
. 4268  ØA        ASL
. 4269  A8        TAY
. 426A  B9 Ø3 46  LDA $46Ø3.Y
. 426D  85 33     STA $33
. 426F  EA        NOP
. 427Ø  EA        NOP
. 4271  EA        NOP
. 4272  EA        NOP
. 4273  EA        NOP
. 4274  EA        NOP
. 4275  EA        NOP
. 4276  EA        NOP
. 4277  EA        NOP
. 4278  EA        NOP
. 4279  EA        NOP
. 427A  B9 ØD 46  LDA $46ØD.Y
. 427D  85 68     STA $68
. 427F  A5 E5     LDA $E5
. 4281  48        PHA
. 4282  86 11     STX $11
. 4284  A4 11     LDY $11
. 4286  2Ø 94 42  JSR $4294
. 4289  68        PLA
. 428A  A8        TAY
. 428B  2Ø 94 42  JSR $4294
. 428E  68        PLA
```

```
. 428F  68        PLA
. 429Ø  4C ØØ 4D  JMP $4DØØ
. 4293  EA        NOP
. 4294  78        SEI
. 4295  8D 3F FF  STA $FF3F
. 4298  A9 FF     LDA #$FF
. 429A  99 EØ 47  STA $47EØ.Y
. 429D  84 E5     STY $E5
. 429F  2Ø EF 41  JSR $41EF
. 42A2  A9 6Ø     LDA #$6Ø
. 42A4  8D 32 45  STA $4532
. 42A7  2Ø ØØ 45  JSR $45ØØ
. 42AA  A9 A9     LDA #$A9
. 42AC  8D 32 45  STA $4532
. 42AF  8D 3E FF  STA $FF3E
. 42B2  58        CLI
. 42B3  6Ø        RTS
. 42B4  EA        NOP

. 4CFØ  68        PLA
. 4CF1  68        PLA
. 4CF2  68        PLA
. 4CF3  68        PLA
. 4CF4  4C 2B 41  JMP $412B
. 4CF7  11 A8     ORA ($A8).Y
. 4CF9  99 EØ 47  STA $47EØ.Y
. 4CFC  4C 79 4D  JMP $4D79
. 4CFF  EA        NOP
. 4DØØ  E6 68     INC $68
. 4DØ2  A5 68     LDA $68
. 4DØ4  C9 B5     CMP #$B5
. 4DØ6  BØ 71     BCS $4D79
. 4DØ8  48        PHA
. 4DØ9  2Ø 8Ø 4D  JSR $4D8Ø
. 4DØC  9Ø E2     BCC $4CFØ
. 4DØE  85 E5     STA $E5
. 4D1Ø  48        PHA
. 4D11  A5 68     LDA $68
. 4D13  85 D1     STA $D1
. 4D15  2Ø B3 4C  JSR $4CB3
. 4D18  A5 11     LDA $11
. 4D1A  85 E4     STA $E4
. 4D1C  48        PHA
. 4D1D  A9 FØ     LDA #$FØ
. 4D1F  85 1Ø     STA $1Ø
. 4D21  2Ø 48 4B  JSR $4B48
. 4D24  68        PLA
. 4D25  85 E4     STA $E4
. 4D27  85 58     STA $58
. 4D29  68        PLA
. 4D2A  85 E5     STA $E5
. 4D2C  85 57     STA $57
. 4D2E  2Ø 9A 4B  JSR $4B9A
. 4D31  A2 Ø1     LDX #$Ø1
. 4D33  B5 57     LDA $57.X
. 4D35  ØA        ASL
. 4D36  ØA        ASL
. 4D37  ØA        ASL
. 4D38  ØA        ASL
. 4D39  95 E4     STA $E4.X
. 4D3B  CA        DEX
. 4D3C  1Ø F5     BPL $4D33
. 4D3E  A9 ØØ     LDA #$ØØ
. 4D4Ø  85 DØ     STA $DØ
. 4D42  A9 46     LDA #$46
. 4D44  85 D1     STA $D1
. 4D46  A2 0F     LDX #$0F
. 4D48  20 E0 4B  JSR $4BE0
. 4D4B  E6 D1     INC $D1
. 4D4D  A9 03     LDA #$03
. 4D4F  85 D0     STA $D0
. 4D51  46 E4     LSR $E4
. 4D53  46 E5     LSR $E5
. 4D55  A2 04     LDX #$04
. 4D57  20 E0 4B  JSR $4BE0
. 4D5A  A9 02     LDA #$02
. 4D5C  85 D0     STA $D0
. 4D5E  A4 E5     LDY $E5
. 4D60  B1 D0     LDA ($D0).Y
. 4D62  85 D5     STA $D5
. 4D64  A9 00     LDA #$00
. 4D66  85 D0     STA $D0
. 4D68  85 D4     STA $D4
. 4D6A  68        PLA
. 4D6B  85 D1     STA $D1
. 4D6D  20 28 4C  JSR $4C28
. 4D70  A5 E5     LDA $E5
. 4D72  4A        LSR
. 4D73  4A        LSR
. 4D74  4A        LSR
. 4D75  A8        TAY
. 4D76  99 E0 47  STA $47E0.Y
. 4D79  78        SEI
. 4D7A  8D 3F FF  STA $FF3F
. 4D7D  4C F2 42  JMP $42F2
. 4D80  A2 00     LDX #$00
. 4D82  BD E0 47  LDA $47E0.X
. 4D85  E4 11     CPX $11
. 4D87  F0 07     BEQ $4D90
. 4D89  C9 FF     CMP #$FF
. 4D8B  D0 03     BNE $4D90
. 4D8D  8A        TXA
. 4D8E  38        SEC
. 4D8F  60        RTS
. 4D90  E8        INX
. 4D91  E0 0E     CPX #$0E
. 4D93  D0 ED     BNE $4D82
. 4D95  18        CLC
. 4D96  60        RTS
. 4D97  10 EA     BPL $4D83

. 41C6  A4 12     LDY $12
. 41C8  30 07     BMI $41D1
. 41CA  88        DEY
. 41CB  30 04     BMI $41D1
. 41CD  84 12     STY $12
. 41CF  F0 03     BEQ $41D4
. 41D1  4C E7 41  JMP $41E7
. 41D4  A2 0E     LDX #$0E
. 41D6  86 E5     STX $E5
. 41D8  A9 B7     LDA #$B7
. 41DA  85 D1     STA $D1
. 41DC  20 08 4C  JSR $4C08
. 41DF  20 28 4C  JSR $4C28
. 41E2  A0 0F     LDY #$0F
. 41E4  20 94 42  JSR $4294
. 41E7  20 50 4F  JSR $4F50
. 41EA  60        RTS
```

```
. 40EA  A9 00     LDA #$00
. 40EC  85 83     STA $83
. 40EE  20 C9 C7  JSR $C7C9
. 40F1  A9 CE     LDA #$CE
. 40F3  8D 15 03  STA $0315
. 40F6  A9 0E     LDA #$0E
. 40F8  8D 14 03  STA $0314
. 40FB  A2 FF     LDX #$FF
. 40FD  9A        TXS
. 40FE  00        BRK
. 40FF  EA        NOP

. 4F50  AD 38 4E  LDA $4E38
. 4F53  D0 1E     BNE $4F73
. 4F55  AD 42 4E  LDA $4E42
. 4F58  D0 19     BNE $4F73
. 4F5A  AD 3C 4E  LDA $4E3C
. 4F5D  D0 14     BNE $4F73
. 4F5F  AD 3B 4E  LDA $4E3B
. 4F62  D0 0F     BNE $4F73
. 4F64  A2 0D     LDX #$0D
. 4F66  BD E0 47  LDA $47E0.X
. 4F69  C9 FF     CMP #$FF
. 4F6B  D0 06     BNE $4F73
. 4F6D  CA        DEX
. 4F6E  10 F6     BPL $4F66
. 4F70  4C 74 4F  JMP $4F74
. 4F73  60        RTS
. 4F74  A0 0E     LDY #$0E
. 4F76  20 94 42  JSR $4294
. 4F79  A2 00     LDX #$00
. 4F7B  86 E5     STX $E5
. 4F7D  A9 C2     LDA #$C2
. 4F7F  85 D1     STA $D1
. 4F81  20 A0 4C  JSR $4CA0
. 4F84  78        SEI
. 4F85  A9 4F     LDA #$4F
. 4F87  8D 15 03  STA $0315
. 4F8A  A9 D0     LDA #$D0
. 4F8C  8D 14 03  STA $0314
. 4F8F  58        CLI
. 4F90  60        RTS

>4F90 60 FF FF FF FF 13 13 13 :'...
>4F98 13 13 13 13 13 13 13 00 :........
>4FA0 00 00 00 00 10 EF 10 EF :.....o.o
>4FA8 10 EF 10 EF 10 EF 10 EF :.o.o.o.o
>4FB0 10 EF 10 EF 10 EF 10 EF :.o.o.o.o
>4FB8 10 EF 10 EF 10 EF 10 EF :.o.o.o.o
>4FC0 10 EF 10 EF 10 EF 10 EF :.o.o.o.o
>4FC8 10 EF 10 EF 10 EF 10 EF :.o.o.o.o

. 4FD0  20 E4 FF  JSR $FFE4
. 4FD3  C9 59     CMP #$59
. 4FD5  F0 0A     BEQ $4FE1
. 4FD7  C9 4E     CMP #$4E
. 4FD9  F0 03     BEQ $4FDE
. 4FDB  4C 0E CE  JMP $CE0E
. 4FDE  4C EA 40  JMP $40EA
. 4FE1  A2 FF     LDX #$FF
. 4FE3  9A        TXS
. 4FE4  4C 00 40  JMP $4000
. 4FE7  EF        ???
```

CON'T NEXT MONTH!

Kw21

FOLLOW UP ARTICLE FOR THE MAGAZINE USING GRAHAM'S BACKUP METHO

GAME	HEX SAVE FROM - TO	START CODE.
AIRWOLF 2	1ØØØ - 4ØØØ	1Ø1Ø
AURIGA	1ØØØ - 4ØØØ	1466
BABY BERKS	1ØØØ - 3ØØØ	144D
BUBBLE TROUBLE	1ØØØ - 4ØØØ	3ØØØ
COMMANDO	1ØØØ - 4ØØØ	1ØØD
DIAGON	1ØØØ - 4ØØØ	3258
DIZZY DICE	1ØØØ - 4ØØØ	11ØØ
DROID ONE	1ØØØ - 4ØØØ	18CE
EXORCIST	1ØØØ - 4ØØØ	3BØØ
FIRE ANT	1ØØØ - 4ØØØ	1AF4 or use 3F9
INVADERS	1ØØØ - 4ØØØ	1ØØE
LOCO CO-CO	1ØØØ - 4ØØØ	13ØØ
MAYHEM	1ØØØ - 4ØØØ	118C
'O' LEVEL MATHS	1ØØØ - 4ØØØ	X RUN
PIZZA PETE	1ØØØ - 4ØØØ	11FØ
PROSPECTOR PETE	1ØØØ - 4ØØØ	18ØØ
RUNNER	1ØØØ - 4ØØØ	1Ø18
SKYHAWK	1ØØØ - 4ØØØ	1Ø5Ø
SOLO	1ØØØ - 4ØØØ	2AA3
SPACE SWEEP	1ØØØ - 4ØØØ	2ØØØ
STREET OLYMPICS	1ØØØ - 4ØØØ	332F
VIDEO MEANIES	1ØØØ - 4ØØØ	1A18 or use 2A1
XADIUM	1ØØØ - 4ØØØ	15Ø5

AMMENDMENT FOR WOLF PACK WHICH WAS EARLIER FEATURED IN VOLUME ONE ISSUE 11/12 OF THE MAG.

It should be saved between 1ØØØ - 4ØØØ (and not 1ØØØ - 5ØØØ as stated)

The start code is 13ØØ (and not 44D1 as stated).

P.T.O.

ALTERNATIVE START CODES FOR GAMES PREVIOUSLY MENTIONED IN VARIOUS ARTICLES.

GAME	START CODE.
AIRWOLF	128Ø
BERKS 1, 2, 3	1Ø8D
ALIEN ATTACK	2ØØØ
BANDITS AT ZERO	3F13
DANGERZONE	16B8
DISASTA BLASTA	3ØØØ
GUNLAW	1Ø1A
INVASION 2000	23ØØ
JETBRIX	3ØØØ
KNOCK OUT	2Ø5Ø
LEGIONAIRE	18ØØ
MONKEY MAGIC	1Ø2Ø
ROBIN TO THE RESCUE	1Ø1Ø
ROCKMAN	1Ø2Ø
SPECTIPEDE.	1Ø7A
TIMESLIP	1Ø3D
TRAILBLAZER.	24CØ
VOID RUNNER	3ØØØ
WINNIE WITCH.	11EØ

by. Kevin wheals.

Dear Roy
I'm afraid I shall not be renewing my subscription to the magazine after the next issue. To be honest, I have found it something of a disappointment, in that I never saw anything appealing to my particular interest BASIC V13 programming.

To show the are no hard feelings, I enclose a small utility program I wrote TODAY 13.11.90 which you may use is you wish. It counts the words in the WP Memory ($C12C-$DF00), which is useful if you write things for publication. Simply exit the WP with a RESET and run the program.

Yours Alexander Hamilton, CHELTENHAM.

Right, now I'll get stuck in, firstly I only had four letters of you in the whole year that you were a member and this is the first contribution that you have sent to me during you duration of subscription, so don't give me a load of BALL CRAP about not supporting peoples needs, I DO TRY, but not everyone contributes to the mag, including yourself, so I have a lot of work to do in finding programs etc, I'm no whizz kid, so what more do you want, BLOOD. The only thing I can put it down to is that your a BLOODY moaner and not a DO-er. Anyway, anyone that wish to the prog its below:-

```
10 REM:PROGRAM WORDCOUNT:(C)A.W.HAMILTON 1990
20 FORI=8192TO8259:READA$:POKEI,DEC(A$):NEXTI:SYS8192:W=256*PEEK(216)+PEEK(217)
30 PRINT"WORD COUNT=";W:END
40 DATA A9,00,85,D8,85,D9,85,DA,A9,C1,85,DB,A0,2C,78,8D,3F,FF,20,39,20,D0,10,C8
50 DATA D0,F8,E6,DB,A5,DB,C9,DF,D0,F0,8D,3E,FF,58,60,E6,D9,D0,02,E6,D8,C8,D0,02
60 DATA E6,DB,20,39,20,F0,DB,D0,F4,B1,DA,30,04,C9,20,D0,02,A9,00,60
```

Dear Roy
It seems no ones sent a prog for your comp. I'm not very good at programming but I've sent this one in its not longenough really to put on tape and it's been done before.

```
10 PRINT CHR$(147):COLOR 0,1
20 C=1:R=3
30 H=1:V=1
40 CHAR 1,C,R,"#"
45 CO=C:RO=R
50 C=C+H:R=R+V
60 IF C=1 OR C=38 THEN H=-H
70 IF R=0 OR R=24 THEN V=-V
80 CHAR ,CO,RO," "
90 A=INT(16*RND(1))+1
95 B=INT(8*RND(1))
100 COLOR 1,A,B
110 GOTO 40
```

Yours faithfully, Peter Appleby, NOTTS.

Thanks Peter, for taking the time and effort for having a go, which is more than could be said for the 98% of the group that did'nt!!! Also in this mag you will find Kevin Wheals comp program, thanks to him aswell.

The Gothic Font
Comp Entry By Kev Wheals

Instructions

Once loaded, type run, the screen will blank to a plain black screen. After its sorted itself out, follow the on-screen instructions to experiment with the text. If you want to use the gothic set to type in your own programs with, instead of the usual commodore character set, just NEW the basic program and carry on.
Please note if your own program exceeds into location 15360 it will corrupt the gothic set, which sits in memory from location.
For those that are well used to switching the character set from ROM to RAM, will easily be able to change the set from 15360 to further up in RAM. Again please note the set can only be put into certain areas, because of the way the computer deals with this.

Cheers, Kevin Wheals.

```
2 REM *************************
3 REM * GOTHIC CHARACTER SET  *
4 REM *************************
5 REM *BY KEVIN WHEALS (C)1990*
6 REM *************************
10 COLOR0,1:COLOR4,1:COLOR1,2
20 POKE55,0:POKE56,60:CLR:POKE1177,62
30 FORI=0TO1023:POKE15360+I,PEEK(53248+I):NEXTI
40 POKE1177,63:POKE65299,60:POKE65298,192:SCNCLR
50 FORL=0TO25:CX=0:FORD=0TO7:READCH:CX=CX+CH:POKE15368+L*8+D,CH:NEXTD
60 READCH:IFCH< >CXTHENPOKE65299,208:POKE65298,196:PRINT"ERROR IN LINE";1000+(L*
0):STOP
70 NEXTL
100 PRINT"(DOWN)(DOWN)(DOWN)"
110 PRINTTAB(8);:PRINT"A B C D E F G H I J K L M"
120 PRINT
130 PRINTTAB(8);:PRINT"N O P Q R S T U V W X Y Z"
140 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"
150 PRINTTAB(5);:PRINT"ALL YOU HAVE TO DO IS CLEAR THE"
160 PRINTTAB(5);:PRINT"SCREEN, AND AWAY YOU GO IN THIS"
170 PRINTTAB(5);:PRINT"GOTHIC CHARACTER SET."
500 REM *** CHARACTER DATA FOLLOWS ***
1000 DATA 048,072,020,034,062,034,065,000, 0335
1010 DATA 092,034,066,124,066,034,092,000, 0508
1020 DATA 028,034,084,080,080,034,028,000, 0368
1030 DATA 088,100,066,066,066,100,088,000, 0574
1040 DATA 092,034,064,112,064,034,092,000, 0492
1050 DATA 092,034,032,120,032,032,064,000, 0406
1060 DATA 028,034,064,094,098,062,002,006, 0388
1070 DATA 028,034,032,060,034,034,036,000, 0258
1080 DATA 002,060,072,008,010,060,064,000, 0276
1090 DATA 001,002,002,002,034,068,056,000, 0165
1100 DATA 066,036,040,112,040,036,066,000, 0396
1110 DATA 024,036,032,032,032,033,094,000, 0283
1120 DATA 084,042,042,106,042,042,064,000, 0422
1130 DATA 066,050,042,106,042,042,068,000, 0416
1140 DATA 028,034,081,081,081,034,028,000, 0367
1150 DATA 092,034,034,124,032,032,064,000, 0412
1160 DATA 056,084,162,162,170,068,058,002, 0762
1170 DATA 092,034,034,120,036,034,066,000, 0416
1180 DATA 002,060,064,060,002,060,064,000, 0312
1190 DATA 001,126,048,080,080,033,030,000, 0398
1200 DATA 033,082,018,018,018,018,012,000, 0199
1210 DATA 076,178,034,034,034,020,008,000, 0384
1220 DATA 128,092,082,082,082,084,040,000, 0590
1230 DATA 034,084,012,008,024,037,066,000, 0265
1240 DATA 066,164,036,036,026,066,060,000, 0454
1250 DATA 126,002,004,008,016,032,064,126, 0378
1260 REM * END OF GOTHIC CHARACTER SET *
```

GAME REVIEW

Reviewer: Mark Lennon, W. SUSSEX
Game Reviewed: Indoor Soccer

Well its not often that I am going to do this, review a C16/+4 football game I mean. The fact is that the C16 is sadly lacking in the football department (don't blame the C16, its the programmers, ED) , but of the few, this one does'nt really deliver the goods. One good thing about the game is that there is an enhanced +4 version... is'nt there? Well there is, but enhanced???? its no different to the C16 version! 1 Or 2 Player, 3 Skill Levels, etc. The game is roughly the same as normal, apart from the fact that theres less players and you can bounce the ball off the wall's, and you can foul as much as you want! (a good option if you're a portsmouth fan!). So how does it play, you all ask, well not to good, to tell the truth, level 3 (the easy level) is way too hard, and god knows what level 1 is like.

The graphics are very average, sound is reduced to the crowd roar, and the ball bounce, and playability is very hard. Of the very few arcade football games on the C16/+4 this is'nt the worst, but a long way off the best. My advice would be, but it but don't expect to much.

Oh by the way for those who don't understand the Portsmouth Bit, they have had the worst disciplinary record for the past 2 seasons, but I don't care because I am a fanatical BRIGHTON supporter! (Who are BRIGHTON?? ED).

Marks out of	100%
GRAPHICS	52%
SOUND	18%
PLAYABILITY	42%
VFM	65%
OVERALL	48%

COMPANY - ALTERNATIVE
PRICE - £1.99

Merry Xmas and a
Happy New year to all C16/+4
Members.
All the best for '91

Roy Robinson

P.S 2 YEARS IN APRIL '91 !