Personal Computer World

CENTURY

BEST OF PCW

SOFTWARE FOR THE BBC MICRO

BEST OF
PERSONAL
COMPUTER WORLD

SOFTWARE FOR THE BBC MICRO

BEST OF
PERSONAL COMPUTER WORLD

SOFTWARE FOR THE BBC MICRO

Century Communications
London

First published in Great Britain in 1983 by
Century Communications Ltd. (a subsidiary of Century
Publishing Co. Ltd.), 76 Old Compton Street, London W1V 5PA.

ISBN 0 7126 0266 6

Printed in Great Britain in 1983 by Garden City Press Limited,
Letchworth, Hertfordshire SG6 1JS.

CONTENTS

Acknowledgements

Special thanks are due to all the original program writers, to our team of converters and referees, and to Ms. Jane Green who organised everybody so well. It has cost much midnight oil, and a few premature grey hairs, maybe, but without their efforts we would never have been able to produce a book as good as this.

INTRODUCTION

The Information Revolution is all about getting the information you want when you want it. How many times have you wished that you had all of PCW's programs and BBC hints in one volume?

For over five years PCW has been collecting up the best in programs and programming tips for every micro. Now we've selected the finest of these and tailor-made the programs to run on your BBC computer.

Each program was redesigned by a BBC expert to make full use of colour, sound and the other facilities of your machine. All the programs have been closely vetted by one of PCW's experienced team of referees to ensure that they are as bug-free as possible.

For a pinch of spice we've added all the best readers' tips and programming hints, providing you with a unique compendium of the Best of PCW for your machine.

Of course one day we'll be inviting you to dial up this information but for the time being enjoy a good, old-fashioned book!

Jane Bird, editor
Personal Computer World

BENNY NOTARIANNI
Original program by Richard Styll

GENERAL DESCRIPTION

The user plays a game of 'Patience' on the screen. The object is to place each suit in ascending order on its respective pile.

Instructions are displayed, then the main program is called. (It is not possible to look at the instructions again without rewinding the tape.) The single pack is laid out in the form of a 'Harp'. The cards may be moved from one column to another or new cards are dealt from the stack. As soon as an Ace becomes available, it may be used to start a file.

No false moves or cheating are allowed! Perhaps a 'Cheat' facility could be added for desperate players?

DETAILED DESCRIPTION

Program 1 Introduction
Lines 10-190 the introductory program.
170 resets PAGE to allow extra memory for disc users.
180 calls the actual game program.
210-420 instructions.
440-560 defines the pips and the back of the cards.
(Thus saving space in Program 2.)

```
LIST
    1  REM *INSTRUCTIONS/INITIALISE PROG
RAM*
    2 REM CHAINS MAIN PROGRAM
   10 MODE7
   20 PROC_pchars
   30 PRINT CHR$141;CHR$130;SPC(10);"PAT
IENCE"
   40 PRINT CHR$141;CHR$131;SPC(10);"PAT
IENCE"
   50 PRINTSPC(14);CHR$134;"BY"
   60 PRINTSPC(7);CHR$133;CHR$136;"B. NO
TARIANNI"
   70 A=INKEY(300)
   80 VDU 30
   90 FOR X%=1 TO 9
  100   VDU 11
  110   A=INKEY(25)
  120   NEXT
  130 VDU 31,3,15
  140 PRINT" Do You want Instructions."
  150 *FX15,1
  160 IF LEFT$(GET$,1)="Y" PROC_instruct
ions
  170 IF PAGE=&1900 THEN PAGE=&1100 : RE
M DISK USERS ONLY
  180 CHAIN"CARDS"
  190 END
  200
  210 DEF PROC_instructions
  220 CLS
  230 PRINT" This is a computerised vers
ion of"
  240 PRINT"the game of Patience."
  250 PRINT" Cards are dealt to a stack
3 at a"
  260 PRINT"time by pressing D. The play
er may"
  270 PRINT"move a card from the stack t
o the"
  280 PRINT"columns by specifying S as t
he"
```

```
  290 PRINT"source and the column number
 as"
  300 PRINT"destination."
  310 PRINT" Aces can be removed from th
e stack"
  320 PRINT"or columns and placed in the
ir own"
  330 PRINT"boxes by specifying P as the
 TO"
  340 PRINT"parameter."
  350 PRINT" The game is won by placing
all"
  360 PRINT"of a suit on it's respective
 ace."
  370 PRINT" All illegal moves result in
 a beep."
  380 PRINT '''CHR$133CHR$136"   Press S
pace To Begin"
  390 REPEAT
  400   A$=GET$
  410   UNTIL A$=" "
  420 ENDPROC
  430
  440 DEF PROC_pchars
  450 FOR character%=224 TO 227
  460   VDU 23,character%
  470   FOR row%=1 TO 8
  480     READ value%
  490     VDU value%
  500     NEXT
  510   NEXT
  520 ENDPROC
  530 DATA &00,&08,&1C,&3E,&7F,&3E,&1C,&
08
  540 DATA &00,&14,&3E,&7F,&7F,&3E,&1C,&
08
  550 DATA &00,&08,&1C,&3E,&7F,&7F,&2A,&
08
  560 DATA &00,&08,&1C,&2A,&7F,&2A,&08,&
08
  570
```

Program 2 The Game

Lines 30-173 sets up the cards and the screen.
180-350 the main program loop. Accepts and validates keyboard entries. (Note the use of INSTR.)
370-610 moves card(s) from one column to another.
630-770 moves a card from the Stack.
781-785 checks the colour of 2 cards.
800-880 displays the next card.
900-980 deals from the Stack.
1020-1140 shuffles the pack initially.
1160-1260 lays out the cards in a 'Harp'.
1266-1272 reveals the next card on the Stack.
1280-1370 initialises arrays.
1410-1560 draws a card on the Stack.
(Note the use of a Graphics Window - VDU 24.)
1590-1650 describes a number card.
1680-1730 describes a picture card.
1810-1890 finds the suit of a card for display on a column.
1910-2040 moves a card to its correct Pile.
3010-3250 draws the screen layout.
4010-4060 prints 'shadow-effect' letters.
6010-6090 win routine.
20010 this DATA statement is a single string defining a pack of cards.
30010-30100 DATA statements defining the pip positions of each number card.

```
>LLIST
   30 DIM board$(7),suits$(4)
   40 DIM c_view%(7)
   50 shuffled$=STRING$(104," ")
   55 stack$=shuffled$
   60 stack$="" : pack$="" : shuffled$="

   70 v$="/A23456789TJQK*"
   73 RESTORE 20010 : READ pack$
   75 MODE 1
   76 A%=RND(-TIME)
   85 VDU 23;8202;0;0;0;
  100 VDU 19,3,6,0,0,0
```

```
  110 VDU 19,2,4,0,0,0
  111 REPEAT
  112   PROC_init
  113   VDU 26,4
  120   GCOL 0,130 : COLOUR 130 : CLG
  130   PROC_draw_screen
  140   PROC_shuffle
  150   PROC_lay_board
  173   won%=FALSE
  180   REPEAT
  183     COLOUR 3
  190     PRINT TAB(24,13) "From  "
  192     *FX 15,1
  200     REPEAT
  210       A$=GET$
  220       UNTIL INSTR("SD1234567",A$)<
>0
  230     IF A$="D" PROC_deal : GOTO 330
  240     PRINT TAB(33,13) A$
  250     from%=VAL(A$)
  260     PRINT TAB(24,14) "TO    "
  266     *FX 15,1
  270     REPEAT
  280       B$=GET$
  290       UNTIL INSTR("P1234567",B$)<>
0
  300     PRINT TAB(33,14) B$
  310     to%=VAL(B$)
  315     IF B$="P" PROC_to_pile : GOTO
330
  320     IF A$="S" A%=FN_from_stack ELS
E A%=FN_move_cards(from%,to%)
  325     IF NOT A% PRINT CHR$7
  330     UNTIL won%=TRUE
  340   PROC_won
  350   UNTIL won%=FALSE
  355 MODE 7
  356 END
  360
  370 DEF FN_move_cards(source%,destinat
ion%)
  380 lead$=RIGHT$(board$(destination%),2)
```

```
  390 IF lead$="" lead$="**"
  400 next$=MID$(v$,INSTR(v$,LEFT$(lead$
,1))-1,1)
  420 string$=board$(source%)
  430 position%=c_view%(source%)*2-1
  435 IF position%<1=FALSE
  480 card$=MID$(board$(source%),positio
n%,2)
  485 IF INSTR(card$,next$)=0 =FALSE
  490 IF NOT FN_test1 THEN=FALSE
  520 sequence$=MID$(string$,position%,L
EN(string$)-position%+1)
  530 FOR X%=1 TO LEN(sequence$) STEP 2
  540   board$(destination%)=board$(dest
ination%)+MID$(sequence$,X%,2)
  550   board$(source%)=LEFT$(board$(sou
rce%),LEN(board$(source%))-2)
  560   PROC_flip_card(destination%)
  570   PROC_flip_card(source%)
  580   VDU 31,source%*3-2,position% DIV
 2 + 3+X% DIV 2
  590   PRINT "  "
  600   NEXT
  603 c_view%(source%)=c_view%(source%)-
1
  610 =TRUE
  620
  630 DEF FN_from_stack
  640 IF LEN(stack$)<1 =FALSE
  650 card$=LEFT$(stack$,2)
  660 lead$=RIGHT$(board$(to%),2)
  670 IF lead$="" lead$="**"
  680 next$=MID$(v$,INSTR(v$,LEFT$(lead$
,1))-1,1)
  690 IF next$="" = FALSE
  700 IF INSTR(card$,next$)=0 =FALSE
  710 IF NOT FN_test1 THEN=FALSE
  740 board$(to%)=board$(to%)+card$
  750 PROC_flip_card(to%)
  760 stack$=RIGHT$(stack$,LEN(stack$)-2
  765 PROC_show_stack
  770 =TRUE
```

```
  780
  781 DEF FN_test1
  782 c11$=RIGHT$(lead$,1) : c12$=RIGHT$
(card$,1)
  783 IF INSTR("SC",c11$)<>0 AND INSTR("
SC",c12$)<>0 =FALSE
  784 IF INSTR("DH",c11$)<>0 AND INSTR("
DH",c12$)<>0 =FALSE
  785 =TRUE
  790
  800 DEF PROC_flip_card(column%)
  810 y%=LEN(board$(column%))
  830 IF y%=0 VDU 31,3*column%-2,y%/2+3
: PRINT "  " : ENDPROC
  840 card$=RIGHT$(board$(column%),2)
  850 IF INSTR("SC",RIGHT$(card$,1)) COL
OUR 0 ELSE COLOUR 1
  860 VDU 31,3*column%-2,y%/2+2
  870 PRINT card$
  871 VDU 31,3*column%-2,y%/2+3
  873 PRINT "  "
  880 ENDPROC
  890
  900 DEF PROC_deal
  910 FOR card%=1 TO 3
  920   IF LEN(shuffled$)=0 THEN shuffle
d$=stack$ : stack$=""
  930   stack$=RIGHT$(shuffled$,2)+stack
$
  940   shuffled$=LEFT$(shuffled$,LEN(sh
uffled$)-2)
  950   NEXT
  960 card$=LEFT$(stack$,2)
  966 PROC_show_stack
  980 ENDPROC
 1010
 1020 DEF PROC_shuffle
 1030 shuffled$=""
 1050 C%=52
 1060 FOR I%=1 TO 52
 1070   A%=INT(RND(1)*C%+1)
```

```
 1080   shuffled$=shuffled$+MID$(pack$,2
*A%-1,2)
 1090   L$=LEFT$(pack$,(A%-1)*2)
 1100   R$=RIGHT$(pack$,(LEN(pack$)/2-A%
)*2)
 1110   pack$=L$+R$
 1120   C%=C%-1
 1130   NEXT
 1135 pack$=shuffled$
 1140 ENDPROC
 1150
 1160 DEF PROC_lay_board
 1165 VDU 31,1,1
 1166 COLOUR 3
 1170 PRINT TAB(1) "1  2  3  4  5  6  7"
 1180 FOR row%=7 TO 1 STEP -1
 1190   PRINT
 1200   FOR column%=1 TO row%
 1210     board$(column%)=board$(column%
)+RIGHT$(shuffled$,2)
 1220     shuffled$=LEFT$(shuffled$,LEN(
shuffled$)-2)
 1223     VDU 31,column%*3-2,10-row%
 1230     PRINT "##" ;
 1240     NEXT
 1250   NEXT
 1255 PROC_show_stack
 1256 FOR I%=1 TO 7 : PROC_flip_card(I%)
 : NEXT
 1260 ENDPROC
 1263
 1266 DEF PROC_show_stack
 1269 PROC_print_card(LEFT$(stack$,2),44
,32)
 1272 ENDPROC
 1275
 1280 DEF PROC_init
 1290 FOR I%=1 TO 7
 1300   board$(I%)=STRING$(38," ")
 1310   NEXT
 1313 FOR I%=1 TO 4
 1316   suits$(I%)="/"
```

```
 1319   NEXT
 1350 FOR I%=1 TO 7
 1352   board$(I%)=""
 1353   c_view%(I%)=8-I%
 1359   NEXT
 1370 ENDPROC
 1380
 1410 DEF PROC_print_card(card$,x%,y%)
 1413 IF card$="/" card$=""
 1420 VDU 29,x%;y%;
 1430 VDU 24,-4;0;192;227;
 1440 GCOL 0,131
 1450 CLG
 1460 VDU 5
 1470 IF card$="" GCOL 0,1 : MOVE 16,128
 : PRINT "EMPTY" : VDU 4 : ENDPROC
 1480 value$=LEFT$(card$,1)
 1490 value%=INSTR("A23456789TJQK",value
$)
 1500 suit$=CHR$(223+INSTR("DHSC",RIGHT$
(card$,1)))
 1510 IF INSTR("SC",RIGHT$(card$,1)) GCO
L 0,0 ELSE GCOL 0,1
 1520 MOVE 0,223 : PRINT value$
 1530 MOVE 160,32 : PRINT value$
 1540 IF value%<11 PROC_number_cards ELS
E PROC_picture_cards
 1550 VDU 4
 1560 ENDPROC
 1590 DEF PROC_number_cards
 1600 RESTORE value%*10+30000
 1610 REPEAT
 1620   READ x%,y%
 1630   IF x%<>0 MOVE x%*4-16,y%*4+16 :
PRINT suit$
 1640   UNTIL x%=0
 1650 ENDPROC
 1660
 1680 DEF PROC_picture_cards
 1690 MOVE 80,124
 1700 PRINT value$
 1710 MOVE 160,223 : PRINT suit$
```

```
 1720 MOVE 0,32 : PRINT suit$
 1730 ENDPROC
 1800
 1810 DEF PROC_show_suit(suit%)
 1820 number%=0
 1825 PROC_print_card(card$,suit%*224+14
8,32)
 1830 FOR suit%=1 TO 4
 1840   card$=RIGHT$(suits$(suit%),2)
 1860   IF LEFT$(card$,1)="K" number%=nu
mber%+1
 1870   NEXT
 1880 IF number%=4 won%=TRUE ELSE won%=F
ALSE
 1890 ENDPROC
 1900
 1910 DEF PROC_to_pile
 1920 IF A$="S" card$=LEFT$(stack$,2) EL
SE card$=RIGHT$(board$(from%),2)
 1930 IF card$="" ENDPROC
 1940 suit%=INSTR("SCDH",RIGHT$(card$,1))
 1950 next$=MID$(v$,INSTR(v$,LEFT$(card$
,1))-1,1)
 1960 IF LEFT$(suits$(suit%),1 )<>next$
ENDPROC
 1970 suits$(suit%)=card$
 1980 PROC_show_suit(suit%)
 1990 IF A$="S" stack$=RIGHT$(stack$,LEN
(stack$)-2) : PROC_show_stack : ENDPROC
 2000 card$=board$(from%)
 2010 board$(from%)=LEFT$(card$,LEN(card
$)-2)
 2011 IF LEN(card$)/2-1 <c_view%(from%)
c_view%(from%)=c_view%(from%)-1
 2013 PROC_flip_card(from%)
 2040 ENDPROC
 3000
 3010 DEF PROC_draw_screen
 3020 GCOL 0,3
 3030 MOVE 12,12 : DRAW 12,1011
 3040 DRAW 1267,1011 : DRAW 1267,12
 3050 DRAW 12,12
```

```
 3060 MOVE 24,24 : DRAW 24,318
 3070 DRAW 256,318 : DRAW 256,24
 3080 DRAW 24,24
 3081 MOVE 348,24 : DRAW 348,318
 3082 DRAW 1255,318 : DRAW 1255,24
 3083 DRAW 348,24
 3090 VDU 5
 3100 MOVE 56,302
 3110 PRINT "Stack " ;
 3120 GCOL 0,0
 3130 PRINT "    Spades Clubs" ;
 3140 GCOL 0,1
 3150 PRINT "  Diams  Hearts"
 3160 GCOL 0,3
 3170 MOVE 690,992 : DRAW 1255,992
 3180 DRAW 1255,662 : DRAW 690,662
 3190 DRAW 690,992
 3200 PROC_double_print( "B E E B",866,9
70,1,3 )
 3210 PROC_double_print( "P A T I E N C
E",736,900,3,0 )
 3220 PROC_double_print( "     b y",772,
836,1,0 )
 3230 PROC_double_print( "B. NOTARIANNI.
",770,756,3,1 )
 3233 VDU 4
 3239 GCOL 0,3
 3240 VDU 4
 3241 MOVE 690,630 : DRAW 1255,630
 3242 DRAW 1255,348 : DRAW 690,348
 3243 DRAW 690,630
 3250 ENDPROC
 4000
 4010 DEF PROC_double_print( a$,x%,y%,B%
,F% )
 4020 GCOL 0,B%
 4030 MOVE x%,y% : PRINT a$
 4040 GCOL 0,F%
 4050 MOVE  x%+4,y%+4 : PRINT a$
 4060 ENDPROC
 6000
 6010 DEF PROC_won
```

```
 6020 VDU 5,26
 6030 PROC_double_print( "Well done !!!"
,732,532,0,1)
 6035 PROC_double_print( "Y O U  W I N "
,764,490,3,0)
 6040 PROC_double_print( "Play again ? "
,732,422,3,1)
 6045 VDU 4
 6049 *FX 15,1
 6050 REPEAT
 6060   A$=GET$
 6070   UNTIL A$="Y" OR A$="N"
 6080 IF A$="N" won%=FALSE
 6090 ENDPROC
20000
20010 DATA AHACADAS2H2C2D2S3H3C3D3S4H4C4
D4S5H5C5D5S6H6C6D6S7H7C7D7S8H8C8D8S9H9C9
D9STHTCTDTSJHJCJDJSQHQCQDQSKHKCKDKS
20020
30000
30010   DATA 24,27,0,0
30020   DATA 24,48,24,8,0,0
30030   DATA 24,48,24,27,24,8,0,0
30040   DATA 13,48,35,48,13,8,35,8,0,0
30050   DATA 13,48,35,48,13,8,35,8,24,27
,0,0
30060   DATA 13,48,35,48,13,8,35,8,13,27
,35,27,0,0
30070   DATA 13,48,35,48,13,8,35,8,13,27
,35,27,24,38,0,0
30080   DATA 13,48,35,48,13,8,35,8,13,27
,35,27,24,38,24,18,0,0
30090   DATA 13,48,35,48,13,8,35,8,24,27
,13,35,35,35,13,21,35,21,0,0
30100   DATA 13,48,35,48,13,8,35,8,13,35
,35,35,13,21,35,21,24,38,24,18,0,0
```

GOLF

RAY DERBY
Original program by Jeff Aughton

GENERAL DESCRIPTION

This program provides a simulated round of golf. You must use your skill and judgement to decide which club to use in each situation and how hard to hit the ball. Remember to use club 19, your putter, when you are on the green. Other instructions are displayed when the game is run. See you in the clubhouse!

DETAILED DESCRIPTION

Lines 10-100 introduction to game and set up variables.
110-260 play each hole then finish game.
270-400 PROC INTRO - Logo and welcome message.
410-730 PROC INIT - instructions and initialising round.
740-810 PROC HOLE - play a hole.
820-1020 play approach with possible hazards.
1030-1150 hole ball and print appropriate message.
1160-1390 PROC FIN - end of game message.
1400-1720 PROC GREEN - deal with putting and display of green.
1730-1860 PROC PIC - display of hole.
1870-2130 PROC HAZARD - generate and deal with random hazard including, if necessary playing out of hazard.

```
LIST
   10 *KEY8RUN|M
   20 *KEY9CLS|M
   30 *KEY7MODE7|M
   40 MODE 7
   50 DATA367,4,445,4,314,4,189,3,520,5
   60 DATA331,4,468,4,375,4,177,3,424,4
```

```
   70 DATA562,5,336,4,208,3,447,4,298,4
   80 DATA543,5,385,4,478,4
   90 PROC_intro
  100 PROC_init
  110 MODE 5
  120 VDU28,0,31,19,25 : VDU 19,1,2,0,0,
0,0
  130 c=0
  140 FORH%=1TO18
  150   YV=0:RV=0
  160   GCOL0,1
  170   PROC_hole
  180   c=c+h
  190   NEXT
  200 MODE 7
  210 PROC_fin
  220 CLS:PRINT''"You went round in ";c
  230 PRINT"Would you like to play anoth
er round"'" 'Y' for Yes"'" 'N' for No"
  240 REPEAT:A$=GET$:UNTIL A$="Y" OR A$=
"N"
  250  IF A$="Y"THENRESTORE:GOTO100
  260 STOP
  270 DEFPROC_intro
  280 CLS
  290 FORC%=130TO132:PRINT'''
  300   PRINTCHR$(C%);CHR$(141);TAB(14)"
PRO _ GOLF"
  310   PRINTCHR$(C%);CHR$(141);TAB(14)"
PRO _ GOLF"
  320   NEXT
  330 PRINTCHR$(135)
  340 ZZ=INKEY(500)
  350 CLS
  360 PRINT"Welcome to the";CHR$(136);"P
.C.W. Links"
  370 PRINT"This is an 18 hole course wi
th a course record of 2 under."
  380 PRINT"Have a good round be courteo
us to your"'"fellow golfers and see you
in the club_ house."
  390 ZZ=INKEY(1000)
```

```
  400 ENDPROC
  410 DEFPROC_init
  420 CLS
  430 REPEAT
  440   PRINT"What is your handicap ";
  450   INPUT H
  460   UNTIL H=INT(H) AND H>-1 AND H<31
  470 PRINT'" DIFFICULTIES AT GOLF "'
  480 PRINT"1..Hooking"'
  490 PRINT"2..Slicing"'
  500 PRINT"3..Poor distance"'
  510 PRINT"4..Trap shots"'
  520 PRINT"5..Putting"'
  530 REPEAT
  540   PRINT"Which is your worst"'"Choo
se the number coresponding to one   only
";
  550   F=GET-48:PRINTF
  560   UNTIL F=INT(F) AND F>0 AND F<6
  570 ZZ=INKEY(500)
  580 CLS
  590 PRINT'"   SELECTION OF CLUBS"
  600 PRINT"YARDAGE      CLUBS     REF No.
"
  610 PRINT'" 280-200  1_4 WOODS   1_4"
  620 PRINT'" 200_100  1_5 IRONS   11_15
"
  630 PRINT'" 100_ 0   6_9 IRONS   16_19
"
  640 ZZ=INKEY(2500)
  650 CLS:FORCH=1TO18:READD,P:PRINT" Hol
e no. ";CH;"  ";D;" yds. Par ";P:NEXT
  660 RESTORE
  670 ZZ=INKEY(2500)
  680 CLS
  690 PRINT'"You control the strength o
f shot by the %age of total swing"'"0%
= weak"'"100% = strong"
  700 VDU19,1,2,0,0,0
  710 haz=FALSE
  720 ZZ=INKEY(2500)
  730 ENDPROC
```

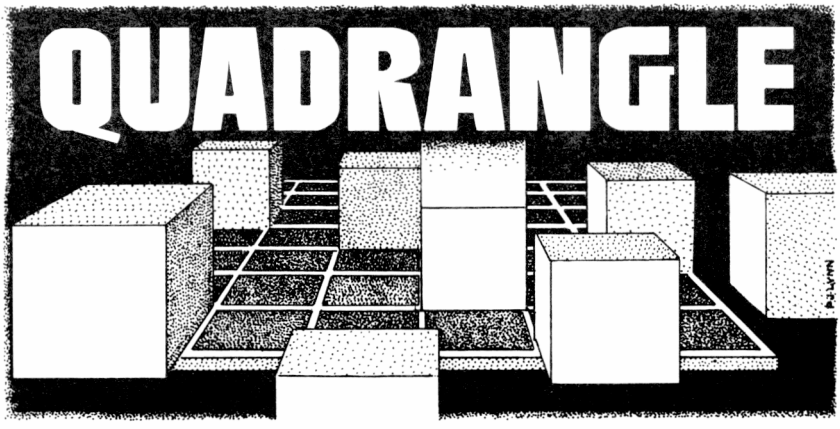
```
  740 DEFPROC_hole
  750 CLS:CLG
  760 READD%,P%
  770 PROC_PIC
  780 PLOT69,TX,TY
  790 PRINT"Hole No. ";H%;'"it's ";D%;"
yards"'"and it's a par ";P%
  800 D=D%
  810 h=0
  820 REPEAT
  830    h=h+1
  840    INPUT"What club ",C
  850    IFC<>INT(C)THEN840
  860    IFC<0 OR C>19 THEN 840
  870    IFC>4 AND C<11 THEN 840
  880    IFC>4THENC=C-6
  890    CLS
  900    INPUT"What %age swing ",B
  910    IF RND(B)>7.5 THEN haz = TRUE
  920    HD=B*(14-C)/13*2.8*(100+(30-H))/
(125+RND(30))
  930    IF F=3 THENHD=HD*.95
  940    SOUND0,-10,4,2
  950    HDP=INT(HD*10+.5)/10
  960    PRINT"You have hit the"'"ball ";
HDP;" yards and"
  970    IFD>0THEN D=D-HD ELSE D=D+HD
  980    IF ABS(D)>=.25 THEN PRINT"you ar
e ";INT(D*10+.5)/10;" yards  from the pi
n" ELSE PRINT"You holed it in ";h
  990    PLOT22,TX+(D%-D)*(HX-TX)/D%,TY+(
D%-D)*(HY-TY)/D%
 1000    IFRND(40)<H OR haz=TRUE THEN PRO
C_HAZARD
 1010    haz=FALSE
 1020     UNTIL ABS(D)<20
 1030 PROC_GREEN
 1040 IF(h-P%+4)<8THENON (h-P%+4) GOTO 1
050,1060,1070,1080,1090,1100,1110 ELSE G
OTO1130
 1050 PRINT"That's an ALBATROSS"'"3 unde
r par":GOTO1120
```

```
 106Ø PRINT"That's an EAGLE"'"2 under pa
r":GOTO1120
 107Ø PRINT"That's a BIRDIE"'"1 under pa
r":GOTO1120
 108Ø PRINT"That's a PAR":GOTO 1120
 109Ø PRINT"That's BOGIE"'"1 over par":G
OTO 1120
 110Ø PRINT"That's DOUBLE_BOGIE"'"2 over
 par":GOTO 1120
 111Ø PRINT"That's TRIPLE_BOGIE"'"3 over
 par":GOTO 1120
 112Ø GOTO114Ø
 113Ø PRINT"That was a disaster"'"but do
n't give up"
 114Ø ZZ=INKEY(5ØØ)
 115Ø ENDPROC
 116Ø DEFPROC_fin
 117Ø diff=c-72
 118Ø diff=diff DIV 2
 119Ø IF diff > 4 THEN 135Ø
 12ØØ IF diff <-1 THEN 137Ø
 121Ø diff=diff+2
 122Ø ON diff GOTO 123Ø,125Ø,127Ø,129Ø,1
31Ø,133Ø
 123Ø PRINT"Great round next time you mi
ght get the record"
 124Ø ENDPROC
 125Ø PRINT"That was pretty good,a nice
round by anystandards"
 126Ø ENDPROC
 127Ø PRINT"Not bad you have talent but
still have  some tricks to learn"
 128Ø ENDPROC
 129Ø PRINT"Look up the club PRO some ti
me he can   help you sort out your game"
 13ØØ ENDPROC
 131Ø PRINT"I've seen worse rounds,but n
ot often .  Ever considered TIDDLYWINKS?"
 132Ø ENDPROC
 133Ø PRINT"When did you learn to play?I
 know it wasthis week but was it YESTERD
AY or the   day before?"
```

```
 1340 ENDPROC
 1350 PRINT"You are a CLOD , a BUFFOON!"
'"Before you despoil this course again"'
"it is required that you learn to play"'
"the game of GOLF!!!"
 1360 ENDPROC
 1370 PRINT"MAGIC! That's a new course r
ecord you   ought to enter for the OPEN"
 1380 ENDPROC
 1390 ENDPROC
 1400 DEFPROC_GREEN
 1410 CLG:CLS:PRINT'TAB(5);"On the green"
 1420 MOVE960,640
 1430 FORd%=0TO360STEP10
 1440    x=640+320*COS(d%*PI/180)
 1450    y=640+320*SIN(d%*PI/180)
 1460    DRAWx,y
 1470    NEXT
 1480 MOVE650,640
 1490 FORd%=0TO360STEP10
 1500    x=640+10*COS(d%*PI/180)
 1510    y=640+10*SIN(d%*PI/180)
 1520    DRAWx,y
 1530    NEXT
 1540 MOVE640,640+D/20*320
 1550 REPEAT
 1560    PLOT22,640,640+D/20*320
 1570    PRINT"You are now ";INT(ABS(D)+.
5);" yards"'"from the pin"
 1580    INPUT"What  CLUB ",C
 1590    IFC<>19THENPRINT"BAD choice of c
lub":GOTO1580
 1600    INPUT"%age swing ",B
 1610    h=h+1
 1620    HD=40*(B+RND(3))/100
 1630    IFF=5THENHD=HD+2*(RND(1)-.5)
 1640    IFABS(D)<1THEN1670
 1650    IFD>0 THEN D=D-HD ELSE D=D+HD
 1660    PRINT"That went ";INT(ABS(HD)+.5
);" yards"
 1670    UNTILABS(D)<1
 1680 FORZZ%=1TO5
```

```
 1690    PLOT22,640,640
 1700    ZZ=INKEY(10)
 1710    NEXT
 1720 ENDPROC
 1730 DEFPROC_PIC
 1740 VDU5
 1750 MOVE640,768:DRAW240,768:DRAW160,64
0:DRAW240,512:DRAW640,512
 1760 IFP%=3THEN DRAW880,512:DRAW960,640
:DRAW880,768:DRAW640,768:HX=900:HY=630
 1770 RV=RND(3)
 1780 IFP%>3ANDRV=1THENDRAW960,384:DRAW1
120,512:DRAW1120,640:DRAW960,768:DRAW640
,768:HX=950:HY=450
 1790 IFP%>3ANDRV=2THENDRAW960,512:DRAW1
120,640:DRAW1120,768:DRAW960,896:DRAW800
,796:DRAW640,768:HX=1000:HY=820
 1800 IFP%>3ANDRV=3THENDRAW1120,512:DRAW
1200,640:DRAW1120,768:DRAW640,768:HX=115
0:HY=650
 1810 MOVEHX,HY:PRINT"O":TX=180:TY=640
 1820 FORF%=1TO11
 1830    MOVE180,640:PLOT70,180,640:ZZ=IN
KEY(10)
 1840    NEXT
 1850 VDU4
 1860 ENDPROC
 1870 DEFPROC_HAZARD
 1880 IFRND(20)=20THEN1890 ELSE 1920
 1890 CLS:PRINT'"Out of bounds !"
 1900 PRINT"drop ball,penalty"'"one stro
ke":h=h+1
 1910 GOTO2130
 1920 IF F<3 THEN 1990
 1930 CLS:PRINT"   B U N K E R"
 1940 PRINT"Your best chance is"'"your 7 iron"
 1950 INPUT"%age of swing ",B
 1960 IFB<0 OR B>100 THEN 1950
 1970 IFRND(B)>75 THEN haz =TRUE:CLS:PRI
NT"FAILED to get out":h=h+1:GOTO2130
 1980 HD=B:D=D-HD:haz=FALSE:PLOT22,TX+(D
%-D)*(HX-TX)/D%,TY+(D%-D)*(HY-TY)/D%
```

```
 1990 CLS:PRINT"In the R O U G H"
 2000 INPUT"Which club ",C
 2010 IFC<>INT(C) OR C<0 THEN 2000
 2020 IFC>4THENC=C-6
 2030 IFC>13THEN2000
 2040 INPUT"%age of swing ",B
 2050 IFB<0 OR B>100 THEN 2040
 2060 IFRND(B)>75 THEN haz =TRUE
 2070 HD=B*(14-C)/13*2.8*(100+(30-H))/(1
25+RND(30))
 2080 D=D-HD
 2090 HDP=INT(HD*10+.5)/10
 2100 PRINT"You have hit the"'"ball ";HD
P;" yards and"
 2110 PRINT"you are ";INT(D*10+.5)/10;"
yards  from the pin"
 2120 PLOT22,TX+(D%-D)*(HX-TX)/D%,TY+(D%
-D)*(HY-TY)/D%
 2130 ENDPROC
```



PAUL BORRETT
Original program by Eileen Baghoomians
and Steven Fawthrop

GENERAL DESCRIPTION

This is an original, colourful game of the Connect 4 and Othello ilk.

The game is played on a 7 by 6 grid with coloured blocks, the object being to place four of your blocks at the corners of a square before your opponent. Blocks may only be placed on top of each other; they cannot hang in mid-air, and the squares formed may be as small or as large as the grid allows.

The program allows for one player to try and beat the computer, or for two players to play each other. If you require advice at any time the computer is very kindly programmed to help you – just press 'H'.

Users with a 32K BBC B model with disc unit will need to alter the memory position by entering: PAGE=&E00 (RETURN) before typing in the listing.

DETAILED DESCRIPTION

Lines 10-80 initial set-up on screen including colour mode.
90-280 introduction to game, input of players' names, skill level required and choice of first move.

290-310 error and help messages.
320-550 analyses game and displays appropriate end-of-game messages.
560-590 offers choice of another game.
600-610 message showing that computer is thinking out next move.
620-770 analyses moves made and decides on computer's next move.
780-800 PROC G - procedure used whilst computer thinks out next move.
810-830 PROC B - shows winning state.
840-860 PROC C - grid outline displayed on screen.
870-940 lines accessed from gosub routines.
950-1030 data listing.
1040-1060 PROC A - procedure analysing number of blocks in columns.
1070-1130 PROC D - procedure used if a column is full.
1140-1150 PROC E - resets variables.
1160-1180 PROC F - displays blocks.
1190-1210 PROC I - inputs player's name.
1220-1230 PROC L - displays introduction and instructions.

```
>LIDST
   1Ø VDU23,224,-1,-1,-1,-1,-1,-1,-1,-1:
*TV255
   2Ø MODE1:VDU19,2,2,Ø,Ø,Ø
   3Ø DIMA$(1),D$(1),S(1),F(7Ø,2),G(12),
M(6,6,2)
   4Ø CLS:PRINT"QUADRANGLE":VDU28,Ø,31,Ø
,Ø:PRINT"UADRANGLE":VDU28,2,31,39,2
   5Ø FORK=1TO22:READL,M:D$(Ø)=D$(Ø)+CHR
$L:D$(1)=D$(1)+CHR$M:NEXT:T$=CHR$31+CHR$
Ø+CHR$25
   6Ø FORK=1TO18:READL:C$=C$+CHR$L:NEXT
   7Ø FORK=1TO7Ø:READF(K,Ø):NEXT
   8Ø FORK=ØTO12:G(K)=2^K:NEXT
   9Ø PROCL
  1ØØ CLS:PRINTTAB(5,5);"ONE OR TWO PLAY
ERS ?";
  11Ø B$=GET$:IFB$<"1"ORB$>"2" THEN11Ø E
LSEP=VALB$
```

```
  12Ø CLS:IFP=1 PRINT"YOUR NAME PLEASE "
;:PROCI:A$(Ø)=A$:A$(1)="COMPUTER":PRINT'
'"YOUR PIECES WILL BE ";CHR$11;D$(Ø)''''"
THE COMPUTER'S WILL BE ";CHR$11;D$(1)
  13Ø IFP=2 PRINT"FIRST PLAYER'S NAME ";
:PROCI:A$(Ø)=A$:PRINT''"YOUR PIECES ARE
";CHR$11;D$(Ø)'''''':PRINT"SECOND PLAYER'
S NAME ";:PROCI:A$(1)=A$:PRINT''"YOUR PI
ECES ARE ";CHR$11;D$(1):P=Ø
  14Ø IFP=1 PRINT''"LEVEL OF PLAY(1-EASY
,2-AVERAGE,3-HARD) ?";CHR$8; ELSEFORK=1T
O2ØØØ:NEXT:E$="Ø":GOTO16Ø
  15Ø E$=GET$:IFE$<"1"ORE$>"3"THEN15Ø
  16Ø T=VALE$:TS=T:S(Ø)=Ø:S(1)=Ø
  17Ø A=RND(1)*2.5+3.5:B=RND(1)*2.5+2.5:
C=5Ø:D=Ø:CLS
  18Ø IFA$(1)<>"COMPUTER"THEN23Ø
  19Ø IFT<3 P=1:GOTO23Ø
  2ØØ PRINT"DO YOU WANT TO GO FIRST ?";
  21Ø B$=GET$:IFB$="Y" P=Ø ELSEIFB$="N"
P=1 ELSE21Ø
  22Ø CLS:IFB$="Y" D=1
  23Ø FORK=ØTO6:FORL=1TO6:M(K,L,Ø)=-1:NE
XT:M(K,Ø,Ø)=Ø:FORM=1TO2:FORL=ØTO6:M(K,L,
M)=Ø:NEXT,,:PROCC
  24Ø      PRINTT$;SPC8Ø;T$;A$(P);"'S MOV
E";:C=C+1
  25Ø      IFP=Ø ORA$(P)<>"COMPUTER"THEN27Ø
  26Ø      IFD=1 Q=P:GOTO6ØØ ELSEIFD=1:G=
RND(4)+2:IFG<6 THEN28Ø ELSEG=4+(RND(2)+1
)*SGN(RND(1)-.5):GOTO28Ø
  27Ø      B$=GET$:G=VALB$
  28Ø      IFB$="H" Q=Ø:T=3:GOTO6ØØ
  29Ø      IFG<1ORG>7 PRINTTAB(Ø,25)"YOU
HAVE MADE AN UNACCEPTABLE MOVE       PLEA
SE TRY AGAIN.":K=INKEY(2ØØ):GOTO24Ø
  3ØØ      G=G-1:IFM(G,Ø,Ø)=6 PRINTTAB(Ø,
25)"THE COLUMN CHOSEN IS OCCUPIED
    PLEASE TRY AGAIN":K=INKEY(2ØØ):GOTO24Ø
  31Ø      T=TS:IFB$="H" PRINTT$;SPC8Ø;T$
;"I RECOMMEND ";(G+1);".";CHR$1Ø;CHR$13;
A$(Ø);"'S MOVE":GOTO27Ø
```

```
  320      M(G,Ø,Ø)=M(G,Ø,Ø)+1:M(G,M(G,Ø,
Ø),Ø)=P:PROCF:PROCE:IFE$<>"Y"GOSUB33Ø:GO
TO49Ø ELSEGOTO49Ø
  330      E=(2^X)OR2^(Y+6):FORJ=1TO7Ø:H=
E ANDF(J,Ø):IFH=E GOSUB35Ø
  340        NEXT:RETURN
  350      IFP=1 THEN42Ø ELSEIFABS(F(J,1)
)>5 THENRETURN ELSEIFABS(F(J,1))<Ø THEN4
5Ø
  360      F(J,1)=F(J,1)+1:IFF(J,1)=1 THE
NFØ=1 ELSEIFF(J,1)=2 THENFØ=2 ELSEFØ=7
  370      U5=1
  380      FORU=ØTO12:F=F(J,Ø)ANDG(U):IFF
=Ø THEN4ØØ
  390        IFU<7ANDU<>X U1=U ELSEIFU>6
ANDU<>Y U2=U-6
  4ØØ        NEXT:M(X,U2,U5)=M(X,U2,U5)+F
Ø:M(U1,Y,U5)=M(U1,Y,U5)+FØ:M(U1,U2,U5)=M
(U1,U2,U5)+FØ
  41Ø      RETURN
  42Ø      IFABS(F(J,1))>5 RETURN ELSEIFF
(J,1)>Ø THEN47Ø
  43Ø      F(J,1)=F(J,1)-1:IFF(J,1)=-1 FØ
=1 ELSE IFF(J,1)=-2 FØ=2 ELSE FØ=7
  44Ø      U5=2:GOTO38Ø
  45Ø      IFF(J,1)=-1 FØ=-1 ELSEIFF(J,1)
=-2 FØ=-2 ELSEFØ=-7
  46Ø      GOSUB44Ø:F(J,1)=1Ø:RETURN
  47Ø      IFF(J,1)=1 FØ=-1 ELSEIFF(J,1)=
2 FØ=-2 ELSEFØ=-7
  48Ø      GOSUB44Ø:F(J,1)=-1Ø:RETURN
  49Ø      FORG=ØTO6:IFM(G,Ø,Ø)=6 NEXT EL
SEG=6:NEXT:GOTO51Ø
  5ØØ    IFE$<>"Y" PRINTT$;SPC8Ø;T$;"THE
GAME IS A DRAW":GOTO56Ø
  51Ø    IFE$<>"Y"P=(P+1)AND1:GOTO24Ø ELS
ES(P)=S(P)+1:PRINTT$;SPC8Ø;T$;A$(P);" WI
NS":PRINTTAB(2Ø*P,Ø);A$(P);" ";S(P)
  52Ø    FORL=1TO16:PRINTTAB(2+X*5,21-(3*
Y));C$;TAB(2+X*5,21-(3*I));C$;TAB(2+(X+W
*(Y-I))*5,21-(3*Y));C$;TAB(2+(X+(Y-I)*W)
*5,21-(3*I));C$
```

```
  530      FORM=1TO100:NEXT
  540      PRINTTAB(2+X*5,21-(3*Y));D$(P)
;TAB(2+X*5,21-(3*I));D$(P);TAB(2+(X+W*(Y
-I))*5,21-(3*Y));D$(P);TAB(2+(X+(Y-I)*W)
*5,21-(3*I));D$(P)
  550      FORM=1TO100:NEXT,:PROCB
  560      PRINTT$;SPC80;T$;"DO YOU WANT
TO PLAY ANOTHER GAME ?";
  570      E$=GET$:IFE$="N"END ELSEIFE$<>
"Y"THEN570
  580      PRINTT$;SPC80;T$;"ARE THE PLAY
ERS THE SAME ?";
  590      E$=GET$:IFE$="N"CLS:GOTO90 ELS
E170
  600      PRINTT$;SPC80;T$;"THINKING";:F
ORK=0TO6:PRINT".";
  610        IFM(K,0,0)=6 M(K,0,1)=-1000:
GOTO630
  620        R=K:S=M(K,0,0)+1:GOSUB870:M(
K,0,1)=F
  630        NEXT:PROCG:IFM(G,0,1)>998 TH
EN310
  640      FORK=0TO6:IFM(K,0,0)<5 THENQ=(
P+1)AND1:R=K:S=M(K,0,0)+2:GOSUB870:IFF=1
000 M(K,0,1)=-999
  650        IFM(K,0,1)>-999 PRINT".";:Q=
P:R=K:S=M(K,0,0)+1:IFM(K,0,0)<6 PROCA:M(
K,0,1)=F
  660        NEXT:IFT=1 THEN770 ELSEIFT=2
 GOSUB890:GOTO770
  670      FORK=0TO6:IFM(K,0,1)<-997 THEN
760 ELSEM(K,M(K,0,0)+1,0)=P:M(K,0,0)=M(K
,0,0)+1
  680        FORM=0TO6:IFM(M,0,0)=6 M(M,0
,2)=-1000:GOTO690 ELSEQ=(P+1)AND1:R=M:S=
M(M,0,0)+1:IFM(K,0,0)<6 PROCA:M(M,0,2)=F
  690          NEXT:PRINT".";
  700        FORLP=1TO4:L=-100:FORM=0TO6:
IFM(M,0,2)>L L=M(M,0,2):LQ=M
  710            NEXT:M(LQ,0,2)=-1000:M(K
,0,1)=M(K,0,1)-L*B
  720          NEXT
```

```
  730        Q=P:R=K:S=M(K,Ø,Ø)+1:GOSUB87
Ø:IFF=1ØØØ M(K,Ø,1)=M(K,Ø,1)-B*A
  740        M(K,Ø,Ø)=M(K,Ø,Ø)-1:M(K,M(K,
Ø,Ø)+1,Ø)=-1
  750        PRINT".";
  760        NEXT:Q=P
  770      PRINTT$;SPC8Ø:PROCG:GOTO31Ø
  780      DEFPROCG
  790      G=-1:L=-1ØØØ:FORK=ØTO6:IFM(K,Ø
,1)>L L=M(K,Ø,1):G=K ELSEIFM(K,Ø,1)=L AN
DABS(3-K)<(3-G) L=M(K,Ø,1):G=K
  8ØØ        NEXT:ENDPROC
  81Ø      DEFPROCB
  82Ø      CLS:PROCC:FORK=ØTO6:FORL=1TO6:
IFM(K,L,Ø)<>-1 PRINTTAB(2+K*5,21-(3*L));
D$(M(K,L,Ø))
  83Ø          NEXT,:ENDPROC
  84Ø        DEFPROCC
  85Ø        FORK=ØTO7:FORL=ØTO6:PRINTTAB
(K*5+1,L*3+2);".":NEXT,:PRINTCHR$3Ø;A$(Ø
);" ";S(Ø);TAB(2Ø,Ø);A$(1);" ";S(1)
  86Ø          FORK=1TO7:PRINTTAB(K*5-2,2
2);K:NEXT:ENDPROC
  87Ø          F=Ø:X=R:Y=S:PROCD:IFE$="Y"
 F=1ØØØ:RETURN
  88Ø          Q=(Q+1)AND1:X=R:Y=S:PROCD:
Q=(Q+1)AND1:IFE$="Y"F=999:RETURN ELSERET
URN
  89Ø          FORK=ØTO6:IFM(K,Ø,1)<-997
THEN93Ø ELSEM(K,M(K,Ø,Ø)+1,Ø)=1:M(K,Ø,Ø)
=M(K,Ø,Ø)+1
  9ØØ            IFM(K,Ø,Ø)<6 Q=(P+1)AND1
:R=K:S=M(K,Ø,Ø):PROCA:M(K,Ø,1)=M(K,Ø,1)-
A*F
  91Ø            M(K,Ø,Ø)=M(K,Ø,Ø)-1:M(K,
M(K,Ø,Ø)+1,Ø)=-1
  92Ø            Q=(P+1)+1:R=K:S=M(K,Ø,Ø)
+1:IFM(K,Ø,Ø)<6 PROCA:M(K,Ø,1)=M(K,Ø,1)+
A*F:M(K,Ø,1)=M(K,Ø,1)*5+RND(4)
  93Ø            PRINT".";:NEXT:Q=P:RETUR
N
  94Ø          RETURN
```

```
  95Ø          DATA17,17,1,2,224,224,224,
224,224,224
  96Ø          DATA224,224,8,8,8,8,8,8,8,
8,1Ø,1Ø
  97Ø          DATA224,224,224,224,224,22
4,224,224
  98Ø          DATA8,8,8,8,8,8,8,8,1Ø,1Ø
  99Ø          DATA17,17,3,3
 1ØØØ          DATA32,32,32,32,8,8,8,8,1Ø
 1Ø1Ø          DATA32,32,32,32,8,8,8,8,1Ø
 1Ø2Ø          DATA387,39Ø,396,4Ø8,432,48
Ø,771,774,78Ø,792,816,864,1539,1542,1548
,1560,1584,1632,3075,3078,3084,3096,312Ø
,3168,6147,6150,6156,6168,6192,624Ø,645,
650,66Ø,68Ø,72Ø,1285,129Ø,13ØØ,132Ø,136Ø
,2565,257Ø
 1Ø3Ø          DATA258Ø,26ØØ,264Ø,5125,51
3Ø,514Ø,516Ø,52ØØ,1161,117Ø,1188,1224,23
13,2322,234Ø,2376,4617,4626,4644,468Ø,21
93,221Ø,2244,4369,4386,442Ø,4257,429Ø
 1Ø4Ø          DEFPROCA
 1Ø5Ø          IFP=Ø F=M(K,M(K,Ø,Ø)+1,1)
ELSEF=M(K,M(K,Ø,Ø)+1,2)
 1Ø6Ø          ENDPROC
 1Ø7Ø          DEFPROCD
 1Ø8Ø          E$="N":FORI=1TO6:IFI=Y I=6
:GOTO113Ø
 1Ø9Ø            IFX+Y-I<Ø ORX+Y-I>6 ORY<
Ø ORY>6 THEN111Ø
 11ØØ            IFM(X,I,Ø)=Q ANDM(X+Y-I,
Y,Ø)=Q ANDM(X+Y-I,I,Ø)=Q E$="Y":W=1:T=I:
I=6:NEXT:I=T:ENDPROC
 111Ø          IFX-Y+I<Ø ORX-Y+I>6 ORY<Ø
ORY>6 THEN113Ø
 112Ø          IFM(X,I,Ø)=Q ANDM(X-Y+I,Y,
Ø)=Q ANDM(X-Y+I,I,Ø)=Q E$="Y":W=-1:T=I:I
=6:NEXT:I=T:ENDPROC
 113Ø        NEXT:ENDPROC
 114Ø      DEFPROCE
 115Ø      Q=P:X=G:Y=M(G,Ø,Ø):PROCD:ENDPR
OC
 116Ø      DEFPROCF
```

```
 1170      IFP=1 ANDA$(P)="COMPUTER" FORK
=1TO9:PRINTTAB(2+G*5,21-(3*M(G,0,0)));D$
(P):FORL=1TO100:NEXT:PRINTTAB(2+G*5,21-(
3*M(G,0,0)));C$:FORL=1TO100:NEXT,
 1180        PRINTTAB(2+G*5,21-(3*M(G,0,0
)));D$(P):ENDPROC
 1190        DEFPROCI
 1200        INPUTA$:IFA$=""ORLENA$>16 PR
INT'"RE-ENTER"':GOTO1200




 1210        ENDPROC
 1220        DEFPROCL
 1230        PRINT''"  This game is simil
ar to Connect 4    and Othello but the o
bject is to       place four of your piec
es at the       corners of a square. The
first to do  so is the  winner."
 1240        PRINT''''"  To move, enter t
he number underneath the column in which
 you wish to play. You can compete with
the computer, or another human. If you n
eed help at    anytime, simply press 'H'
."
 1250        PRINT''''''"       PRESS A KE
Y TO PLAY";:A$=GET$:VDU26:ENDPROC
```

VIRUS

BRIAN JONES
Original program by Allan and Sue Vining

GENERAL DESCRIPTION

A light cycle by any other name would shine as bright!

This fast-moving game casts you in the role of avenging antibody, protecting your pound of flesh from an invading virus.

Both you and the virus move around the screen leaving trails, and you must use the cursor keys to trap the virus without running into anything yourself, but beware, if you succeed then you are faced with two viruses.

The game uses MODE 7, and has excellent sound effects, but only runs on a model B.

DETAILED DESCRIPTION

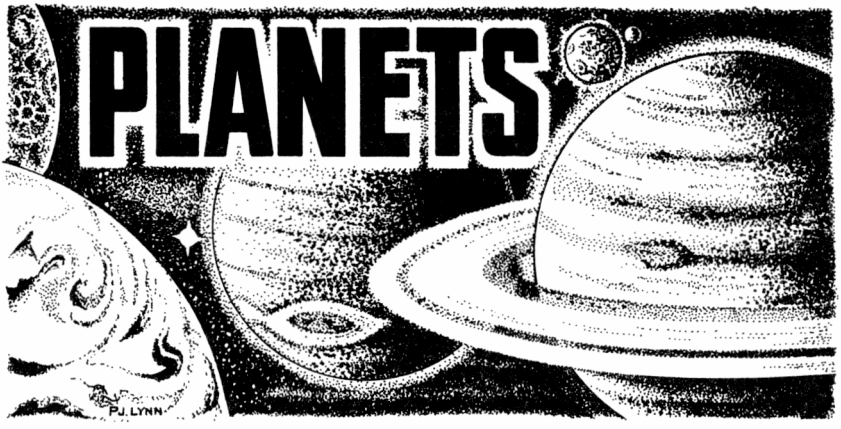
Lines 10-140 set things up, print the instructions, and wait until the space bar is pressed.
150-370 set up the screen.
380-450 move antibody and see if it hits anything.
460-530 check how many viruses are still alive, move them, and go round again.
550-570 you lose, start again.
580-630 you win, end game.
650-820 move virus or turn right or left.
840-870 delay for T% hundredths of a second.
890-940 print a string in double height anywhere on the screen in any colour.

```
LIST
   10 MODE 7 : VDU 23,1,0;0;0;0; : *FX4,1
   20 *FX11,0
   30 ON ERROR REPORT : PRINT " at line
";ERL : GOTO 590
```

```
   40 ENVELOPE 1,1,5,0,-10,10,5,5,30,-2,
0,-10,126,110
   50 PROCdouble(18,1,"VIRUS")
   60 PRINT TAB(16,3);CHR$(148);",,,,,,,"
   70 PRINT 'CHR$(130);"YOU CONTROL AN A
NTIBODY";
   80 PRINT CHR$(135);"o";CHR$(130);
   90 PRINT "THAT MUST"
  100 PRINT CHR$(130);"TRAP A VIRUS. YOU
 START OFF WITH ONE &"
  110 PRINT CHR$(130);"AS YOU ELIMINATE
THEM THE STRAIN GROWS."
  120 PRINT CHR$(131);"CONTROL IS BY THE
 CURSOR KEYS"
  130 PRINT 'SPC(8);CHR$(157);CHR$(132);
"PRESS SPACE TO START  ";CHR$(156)
  140 REPEAT : UNTIL INKEY(-99)
  150 SOUND 1,1,100,20 : CORNER=&7C00
  160 DIM M(4) : FOR I=1 TO 4 : READ M(I
) : NEXT I
  170 DATA190,205,230,255
  180 DIM V(4) : FOR I=1 TO 4 : READ V(I
) : NEXT I
  190 DATA CORNER+207,CORNER+636,CORNER+
180,CORNER+562
  200 DIM B(4)
  210 N=0 : U=0
  220 N=N+1 : IF N=5 THEN 580
  230 CLS
  240 Z=CORNER+500
  250 FOR I=1 TO 4 : B(I)=RND(4) : NEXT I
  260 B=40 : C=1 : D=32 : X=RND(4)
  270 IF X=1 THEN X=B
  280 IF X=2 THEN X=-B
  290 IF X=3 THEN X=C
  300 IF X=4 THEN X=-C
  310 PRINT TAB(0,0);"XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXX"
  320 FOR I=1 TO 23
  330   PRINT TAB(0,I);"X
                       X";
  340   NEXT I
```

```
  350 PRINT TAB(0,24);"XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX";
  360 CORNER?999=ASC("X")
  370 SOUND 0,-10,4,2 : S%=TIME
  380 K=INKEY(0) : IF K=-1 THEN 430
  390 IF K=138 AND X<>-B THEN X=B
  400 IF K=137 AND X<>-C THEN X=C
  410 IF K=136 AND X<>C THEN X=-C
  420 IF K=139 AND X<>B THEN X=-B
  430 Z=Z+X
  440 IF ?Z<>32 THEN 550
  450 ?Z=ASC("o")
  460 FOR T=1 TO N : IF B(T)=0 THEN U=U+
1
  470   NEXT
  480 IF U=N THEN U=0 : GOTO 220
  490 U=0
  500 FOR T=1 TO N : IF B(T)<>0 THEN PRO
Cvirus
  510   NEXT
  520 PROCdelay(30-(TIME-S%))
  530 GOTO 370
  540
  550 FOR I=-15 TO -1 STEP 2 : SOUND 0,I
,RND(3)+3,3 : NEXT I
  560 SOUND 1,1,100,20
  570 PROCdouble(13,12,"YOU CRASHED") :
PROCdelay(400) : RUN
  580 PROCdouble(15,12,"YOU WIN")
  590 *FX15,1
  600 *FX12,0
  610 *FX4,0
  620 VDU 23,1,1;0;0;0;
  630 END
  640
  650 DEF PROCvirus
  660 SOUND &13,1,M(T),3 : G=0 : F=0
  670 IF ?V(T)=D THEN 770
  680 ON T GOTO 690,700,710,720
  690 ON B(T) GOTO 730,740,750,760
  700 ON B(T) GOTO 730,760,750,740
  710 ON B(T) GOTO 740,760,730,750
```

```
  720 ON B(T) GOTO 760,730,740,750
  730 G=B : GOTO 770
  740 G=-C : GOTO 770
  750 G=-B : GOTO 770
  760 G=C
  770 F=F+1 : V(T)=V(T)+G
  780 IF ?V(T)<>D THEN V(T)=V(T)-G : B(T
)=(B(T) MOD 4)+1 : IF F<5 THEN 680
  790 W=V(T)
  800 IF ?(W-B)<>D AND ?(W+B)<>D AND ?(W
-C)<>D AND ?(W+C)<>D THEN B(T)=0
  810 ?V(T)=ASC("*")
  820 ENDPROC
  830
  840 DEF PROCdelay(T%)
  850 TT%=TIME
  860 REPEAT UNTIL TIME>TT%+T%
  870 ENDPROC
  880
  890 DEF PROCdouble(X,Y,A$)
  900 FOR I=0 TO 1
  910   PRINT TAB(0,Y+I);SPC(40)
  920   PRINT TAB(X-2,Y+I);CHR$(141);CHR
$(129);A$
  930   NEXT
  940 ENDPROC
```



PAUL BORRETT
Original program by C. Ward

GENERAL DESCRIPTION

This is a program which simulates, using the high-resolution graphics of MODE 2, the movement of the four inner planets about the sun. Using different colours for each, the orbits of Mercury, Venus, Mars and good old Mother Earth can be viewed in relation to each other. The month and year is also displayed and updated, so as to give an indication of when the planets will be in the displayed positions - just the thing for star-gazers and budding astrologers!

There is little user interaction with the program, once the Solar System has been set into motion. The SHIFT key pauses the program whilst it is depressed but, apart from that, there is little else to do other than sit back and watch the world(s) go by!

DETAILED DESCRIPTION

Lines 10-160 display heading and instructions.
170-280 display colour codes for planets, initialise date arrays and planet positions, display the Sun.
290-520 within an infinite loop - calculate X and Y co-ordinates of each of the four planets and PLOT them so as

to produce a green trace whenever the planet moves on. Display date at bottom of screen. If SHIFT is depressed, do nothing until the situation changes.
530-550 procedure to display 'planet' at its given X,Y position.
560 date DATA statement.

```
   1Ø *TV255
   2Ø MODE7:PRINT''CHR$141;CHR$13Ø;"Planets"
   3Ø PRINTCHR$141;CHR$13Ø;"Planets"
   4Ø PRINT''"This is a simulation of the fou
r inner"
   5Ø PRINT"planets:- Mercury,Venus,Earth and
 Mars."
   6Ø PRINT'"You will be shown their orbits a
round"
   7Ø PRINT"the sun with green trails left be
hind"
   8Ø PRINT"them,along with the date when the
"
   9Ø PRINT"planets will be in that position.
"
  1ØØ PRINT'"The planets are represented by f
our"
  11Ø PRINT"flashing blocks in the same colou
rs as"
  12Ø PRINT"their names,shown at the top of t
he"
  13Ø PRINT"screen."
  14Ø PRINT'"Your may pause the program at an
y time"
  15Ø PRINT"by pressing the SHIFT key."
  16Ø PRINTTAB(Ø,23);CHR$133;"Press RETURN to
 begin";:REPEAT UNTILGET=13
  17Ø MODE2
  18Ø VDU23,224,Ø,6Ø,126,126,126,126,6Ø,Ø,23;
8202;Ø;Ø;Ø;
  19Ø COLOUR5:PRINTTAB(Ø,Ø);"Mercury":COLOUR7
:PRINTTAB(14,Ø);"Venus"
  2ØØ COLOUR6:PRINTTAB(Ø,2);"Earth":COLOUR1:P
RINTTAB(15,2);"Mars":COLOUR2
  21Ø DIMMonth$(12),Monthlength(12)
  22Ø Month=8:Day=1:Year=1983
```

```
  230 FORI=1TO12:READMonth$(I),Monthlength(I)
:NEXT
  240 Marspos=-356.9
  250 Earthpos=-210.2
  260 Venuspos=-194.5
  270 Mercurypos=-87.5
  280 GCOL0,3:MOVE616,528:VDU5,224
  290 REPEAT:VDU5
  300    Marsx=640+SIN(Marspos/180*PI)*440:Mar
sy=512+COS(Marspos/180*PI)*440
  310    GCOL3,1:PROCPlanet(Marsx,Marsy)
  320    Earthx=640+SIN(Earthpos/180*PI)*285:E
arthy=512+COS(Earthpos/180*PI)*285
  330    GCOL3,6:PROCPlanet(Earthx,Earthy)
  340    Venusx=640+SIN(Venuspos/180*PI)*210:V
enusy=512+COS(Venuspos/180*PI)*210
  350    GCOL3,7:PROCPlanet(Venusx,Venusy)
  360    Mercuryx=640+SIN(Mercurypos/180*PI)*1
10:Mercuryy=512+COS(Mercurypos/180*PI)*110
  370    GCOL3,5:PROCPlanet(Mercuryx,Mercuryy)
  380    REPEAT:UNTILNOTINKEY(-1)
  390    GCOL3,1:PROCPlanet(Marsx,Marsy)
  400    GCOL3,6:PROCPlanet(Earthx,Earthy)
  410    GCOL3,7:PROCPlanet(Venusx,Venusy)
  420    GCOL3,5:PROCPlanet(Mercuryx,Mercuryy)
  430    GCOL0,2:PLOT69,Marsx+8,Marsy+8:PLOT69
,Earthx+8,Earthy+8
  440    PLOT69,Venusx+8,Venusy+8:PLOT69,Mercu
ryx+8,Mercuryy+8
  450    Marspos=Marspos-0.52405
  460    Earthpos=Earthpos-0.9856
  470    Venuspos=Venuspos-1.6021
  480    Mercurypos=Mercurypos-4.0924
  490    VDU4
  500    PRINTTAB(0,31);Day;" ";Month$(Month);
" ";Year;STRING$(18-POS," ");
  510    Day=Day+1:IFDay>Monthlength(Month) TH
ENDay=1:Month=Month+1:IFMonth=13 THENMonth=1:
Year=Year+1:IFYear/4=INT(Year/4) ANDYear/100<
>INT(Year/100) THENMonthlength(2)=29 ELSEMont
hlength(2)=28
  520    UNTILFALSE
```

```
  530 DEFPROCPlanet(X,Y)
  540 MOVEX,Y:DRAWX+16,Y:MOVEX,Y+4:DRAWX+16,Y
+4:MOVEX,Y+8:DRAWX+16,Y+8
  550 ENDPROC
  560 DATAJanuary,31,February,29,March,31,Apr
il,30,May,31,June,30,July,31,August,31,Septem
ber,30,October,31,November,30,December,31
```

TARGET PRACTICE

LEON GOODFRIEND
Original program by Gordon Mills

GENERAL DESCRIPTION

This is a game for two players, in which the object is to obliterate 10 randomly placed targets with 15 missiles, the missile launcher moving across the bottom of the screen at a variable speed.

After providing their names, the players have to select the speed at which their missile launcher is to move: this can be anything from a gentle plod to a definite whoo-oosh! Missiles are launched by pressing a key; their journey towards the target and, with luck and skill, the eventual explosion, are accompanied by some realistic sound effects. A count of how many missiles each player has left is shown throughout.

DETAILED DESCRIPTION

Lines 10-110 introduction to game and instructions.
120-210 input names of players and speeds for their bases.
220-340 set up characters and sounds and initialise arrays.
350-460 set up screen display with targets.
479-990 alternate goes between players until both are out of shots.
Block split up as follows:
490-530 move player 2 base across screen at appropriate speed.
540-690 fire shot player 1 if key pressed, and explode target if hit.
720-730 keep moving base 1 until boundary crossed.
740-780 move player 1 across screen at appropriate speed.
790-940 fire shot player 2.
980-990 keep moving base 2 until boundary crossed.
1000-1090 game over, print winner, option to play again.

1100-1280 procedure for blowing up target with explosion and flashing colour.

```
   10 MODE 7
   20 PRINTTAB(10,5)CHR$(141)CHR$(131)"TARGET
 PRACTICE"
   30 PRINTTAB(10)CHR$(141)CHR$(131)"TARGET P
RACTICE"
   40 PRINTTAB(11)CHR$(147)"~~~~~~~~~~~~~~~~"
   50 PRINT'"This is a two player game."
   60 PRINT"The object is to destroy as many
targets";
   70 PRINT"as possible with your fifteen sho
ts."
   80 PRINT"The player on the left uses the '
Z' key"
   90 PRINT"and the player on the right uses
'/' to"
  100 PRINT"fire a missile.  The speed of the
"
  110 PRINT"missile launcher is variable"
  120 PRINT'"Please enter your names"
  130 INPUT'"Player one ",A$
  140 IF LEN(A$)>14THEN LETA$=LEFT$(A$,14)
  150 INPUT"Player two ",B$
  160 IF LEN(B$)>14THEN LETB$=LEFT$(B$,14)
  170 PRINT'"Enter speed (1 fast - 10 slow)"
  180 PRINT'A$" ";:INPUTA
  190 IFA<1 OR A>10THEN 180
  200 PRINTB$" ";:INPUTB
  210 IFB<1 OR B>10THEN 200
  220 MODE 1
  230 COLOUR 2
  240 VDU23;8202;0;0;0;
  250 VDU23,224,28,34,73,93,73,34,28,0
  260 T$=CHR$(224)
  270 VDU23,225,0,24,24,24,24,60,126,231
  280 S$=CHR$(225)
  290 ENVELOPE1,2,7,-1,0,15,45,0,5,-1,0,-127,
100,50
  300 DIM A%(18)
  310 FORA%=1TO18:A%(A%)=A%:NEXT A%
  320 FORA%=1TO10
```

```
  330    M%=RND(18)
  340    IFA%(M%)=ØTHEN33Ø
  350    PRINTTAB(M%,A%*2+2)T$
  360    PRINTTAB(M%+2Ø,A%*2+2)T$
  370    A%(M%)=Ø
  380    NEXT A%
  390 PRINTTAB(8-LEN A$/2,Ø)A$;TAB(28-LEN B$/
2,Ø)B$
  4ØØ COLOUR 3
  410 PRINTTAB(Ø,31)"15 Shots left";TAB(2Ø)"1
5 Shots left";
  42Ø MOVEØ,96Ø:DRAW128Ø,96Ø
  43Ø MOVE636,7Ø:DRAW636,1Ø24
  44Ø MOVE64Ø,7Ø:DRAW64Ø,1Ø24
  45Ø SA=15:SB=15:S1%=Ø:S2%=Ø
  46Ø VDU5
  47Ø FORD=1TO2ØØØ:NEXT D
  48Ø FORC=1TO15Ø
  49Ø   IFSA=ØORS1%=1ØTHEN74Ø
  5ØØ   FORX=8TO584STEP16
  51Ø     MOVEX,72:PRINTS$;
  52Ø     FORD=1TO2Ø*A:NEXT D
  53Ø     IF NOT INKEY(-98)THEN72Ø
  54Ø     SOUND1,1,12Ø,24
  55Ø     SA=SA-1
  56Ø     VDU4
  57Ø     PRINTTAB(Ø,31);SA" ";
  58Ø     CX%=X+12
  59Ø     FORY%=76TO95ØSTEP4
  6ØØ       PLOT69,CX%,Y%:PLOT69,CX%+4,Y%
  61Ø       IF POINT(CX%,Y%+4)=Ø THEN 66Ø
  62Ø       SOUNDØ,-15,4,2Ø
  63Ø       PLOT71,CX%,Y%:PLOT71,CX%+4,Y%
  64Ø       PROCEXPLO(CX%DIV32,31-Y%DIV32)
  65Ø       Y%=95Ø
  66Ø       PLOT71,CX%,Y%:PLOT71,CX%+4,Y%
  67Ø       NEXT Y%
  68Ø     VDU5
  69Ø     MOVEX+32,7Ø
  7ØØ     VDU127
  71Ø     X=584
  72Ø     VDU127
```

```
  730      NEXT X
  740    IF SB=0ORS2%=10THEN990
  750    FORX=648TO1224STEP16
  760      MOVEX,72:PRINTS$;
  770      FORD=1TO20*B:NEXT D
  780      IF NOT INKEY(-105)THEN970
  790      SOUND1,1,120,24
  800      SB=SB-1
  810      VDU4
  820      PRINTTAB(20,31);SB" ";
  830      CX%=X+12
  840      FORY%=76TO950STEP4
  850        PLOT69,CX%,Y%:PLOT69,CX%+4,Y%
  860        IF POINT(CX%,Y%+4)=0 THEN 910
  870        SOUND0,-15,4,20
  880        PROCEXPLO(CX%DIV32,31-Y%DIV32)
  890        PLOT71,CX%,Y%:PLOT71,CX%+4,Y%
  900        Y%=950
  910        PLOT71,CX%,Y%:PLOT71,CX%+4,Y%
  920        NEXT Y%
  930      VDU5
  940      MOVEX+32,70
  950      VDU127
  960      X=1216
  970      VDU127
  980      NEXT X
  990    NEXT C
 1000 VDU4
 1010 VDU28,1,29,38,22
 1020 COLOUR129:CLS
 1030 PRINTTAB(14)"GAME OVER"
 1031 PRINT'A$;TAB(16);S1%'B$;TAB(16);S2%
 1032 IFS1%>S2%PRINT'A$" is the winner!":GOTO
1040
 1033 IFS1%<S2%PRINT'B$" is the winner!":GOTO
1040
 1034 IFS1%=S2%THENIFSA=SB PRINT'"A draw!":GO
TO1040
 1035 IFSA>SB PRINT'A$; ELSEPRINT'B$;
 1036 PRINT" wins - less shots used."
 1040 PRINT'TAB(9);"Another game -Y/N ?";
 1050 *FX15,1
```

```
 1060 A$=GET$
 1070 IF INSTR("Yy",A$)THEN RUN
 1080 IF INSTR("Nn",A$)THENMODE7:END
 1090 GOTO1060
 1100 DEFPROCEXPLO(X,Y)
 1110 IFX<20 S1%=S1%+1:PRINTTAB(17,0);S1%ELSE
S2%=S2%+1:PRINTTAB(37,0);S2%
 1120 LOCAL A,B
 1130 VDU19;4;0;
 1140 FORA=1TO24:NEXT A
 1150 VDU19;0;0;
 1160 VDU4
 1170 FORA=1TO7
 1180   COLOUR2
 1190   PRINTTAB(X,Y)"#"
 1200   FORB=1TO35:NEXT B
 1210   COLOUR1
 1220   PRINTTAB(X,Y)T$
 1230   FORB=1TO120:NEXT B
 1240   NEXT A
 1250 PRINTTAB(X,Y)" "
 1260 VDU5
 1270 COLOUR3
 1280 ENDPROC
```

MUSIC PLAYER

Original program by K. Hussey

GENERAL DESCRIPTION

This is an entertaining program which stores and plays musical sequences input by the user. These sequences can employ notes across three octaves, and can include notes, and rests, of different durations (from semibreve down to semiquaver). Repeats can be built in quite simply by defining the group of notes to be duplicated.

Notes are defined by using a simple notation which comprises [duration] [tone] where the former can range from 2 (semiquaver) to 16 (semibreve), and the latter can be selected from A-G (low octave), a-g (middle) or a'-g' (high). Sharps are defined by using the symbol #. Rests are indicated using the same approach for duration, followed by R or r.

The program is menu-driven, and provides a number of functions. A music sequence can be defined, or notes added to an existing sequence. The currently stored sequence can be listed, and individual notes changed. And, of course, the sequence can be played - in a manner that is rather lacking in soul perhaps, but with perfect pitch and timing!

DETAILED DESCRIPTION

Lines 10-60 initialisation, then jump to menu display.

70-260 menu item 1 - definition of note sequence. Outputs note number as a prompt and inputs user response. This is evaluated to determine note and duration, and both stored. The evaluated note is played.

270 menu item 2 - play stored note sequence.

280-300 procedure to play an individual note.

310-360 procedure to play a sequence of notes.

370-380 menu item 3 - change a particular note. Inputs number of the note in the sequence, and asks for new input for this note.

390-470 displays menu and asks for selection of menu function before jumping to appropriate place in the program.

480-570 procedure for generating repeat sequences from the currently stored note sequence.

580-620 menu item 4 - continue definition of note sequence. Displays last few notes of stored sequence, and then asks for more.

630-680 menu item 5 - list stored note sequence. Asks for start and end points, and then lists the notes between those points.

```
>L.
   10 DIM a%(120),f%(120),m%(120),entry$(120)
   20 ON ERROR GOTO 390
   30 C$="AA#BCC#DD#EFF#GG#"
   40 D$="aa#bcc#dd#eff#gg#"
   50 E$="a'a'#b'c'c'#d'd'#e'f'f'#g'g'#"
   60 GOTO 390
   70 CLS
   80  PRINT''TAB(4);"press escape to return
 to menu."'"ENTER r1-20 or R1-20 to repeat no
tes""1 to 20 and add them to the tune."
   90 K=1:temp%=K:flag%=0
  100 PRINT,K;:INPUT N$
  110 entry$(K)=N$
  120 IF ASC(N$)=&52 OR ASC(N$)=&72 PROCrpt:G
OTO 100
  130 test%=ASC(MID$(N$,2,1)):IF test%<&39 TH
EN k%=2 ELSE k%=1
  140 m%(K)=VAL(MID$(N$,1,k%))
  150 M$=MID$(N$,1+k%,LEN(N$)-k%)
  160 l%=INSTR(C$,M$):IF l%<>0 THEN N=36:GOTO
 210
  170 l%=INSTR(D$,M$):IF l%<>0 THEN N=84:GOTO
 210
  180 l%=INSTR(E$,M$):IF l%<>0 THEN N=134:GOT
O 240
  190 a%(K)=0:f%(K)=0:PROCtone
  200 K=K+1:temp%=K:GOTO 100
  210 a%(K)=-15:f%(K)=N+4*(l%+(l%>3)+(l%>7)+(
l%>10)+(l%>14))
```

```
  220 PROCtone:IF flag%=0 THEN K=K+1:temp%=K
ELSE K=temp%:flag%=0
  230 GOTO 100
  240 a%(K)=-15:f1%=1%+(1%>5)+(1%>12)+(1%>17)
+(1%>24)
  250 f%(K)=N+4*(f1%/2)
  260 GOTO 220
  270 PROCplay:goto 390
  280 DEFPROCtone
  290 SOUND 1,a%(K),f%(K),m%(K)
  300 ENDPROC
  310 DEF PROCplay
  320 FOR q%=1 TO temp%-1
  330   SOUND 1,a%(q%),f%(q%),2*m%(q%)-1
  340   SOUND 1,0,0,1
  350   NEXT q%
  360 ENDPROC
  370 CLS:PRINT''''''''"Enter the note number
of the change":INPUT K:CLS
  380 flag%=1:GOTO 100
  390 CLS:d$=CHR$(&8D):T$="MUSIC PLAYER"
  400 PRINT''TAB(8);d$;T$'TAB(8);d$;T$
  410 PRINT''"1)....Enter notes:16-Semibreve"
'TAB(18);"8-Minim"'TAB(18);"4-Crotchet"'TAB(1
8);"2-Quaver"'TAB(18);"1-Semiquaver"
  420 PRINT TAB(5);"For instance:4c# or 16A o
r 6d'#"'TAB(5);"Enter rests as 8r or 8R or 2r
 etc."
  430 PRINT'"2)....Play the phrase"'"3)....Ch
ange an entry"'"4)....Continue entries"'"5)..
..List entries to specified point"'"6)....Lea
ve the program"
  440 PRINT'STRING$(13,"@*@")
  450 PRINT'"Enter your choice and then RETUR
N"
  460 INPUT choice:ON choice GOTO 70,270,370,
580,630,470,
  470 END
  480 DEFPROCrpt
  490 pos%=INSTR(N$,"-")
  500 r1%=VAL(MID$(N$,2,pos%-2))
  510 r2%=VAL(MID$(N$,pos%+1,LEN(N$)-pos%))
```

```
  520 REPEAT
  530   entry$(K)=entry$(r1%)
  540   m%(K)=m%(r1%):a%(K)=a%(r1%):f%(K)=f%(
r1%)
  550   PROCtone:K=K+1:temp%=temp%+1:r1%=r1%+
1
  560   UNTIL r1%=r2%+1
  570 ENDPROC
  580 K=temp%:CLS
  590 FOR list%=K-5 TO K-1
  600   PRINT list%;"?";entry$(list%)
  610   NEXT list%
  620 GOTO 100
  630 CLS:INPUT''''''''"Listing required from
 note:"n1%
  640 INPUT"to note:"n2%
  650 FOR list%=n1% TO n2%
  660   PRINT list%;"?";entry$(list%)
  670   NEXT list%
  680 K=n2%+1:GOTO 100
```

JAMES BYRNE
Original program by Paul Bradshaw

GENERAL DESCRIPTION

This is a nice simple game in which you play the role of a sly cat. Your task is to prevent mice from eating a hunk of cheese. Instructions for moving, and for trapping mice, will appear on the screen when the program is run.

You score points for eating the mice as quickly as possible when they appear. If you don't kill them they will eventually run into the cheese and take a huge bite out of it. When all the cheese is gone the game is over. Happy mouse-hunting!

DETAILED DESCRIPTION

Lines 10-40 set up variables for game and call initialising procedures.
50-90 reset screen and cheese, start cat.
100-200 start new mouse with sound effects.
210-260 move mouse and cat until mouse hits something.
270-290 if all the cheese is gone give message for end of game, then restart from line 50.
300-460 procedure for moving mouse, checks for collisions.

Calls procedure for moving cat several times while doing this.
470-650 procedure to set up screen with border, cheese, and score display.
660-720 procedure to find position and direction for new mouse.
730-780 procedure to find random position for cat to start from.
790-870 procedure to make squeal and add to score when a mouse has been eaten.
880-980 procedure to make gulp and remove cheese when a mouse eats some.
990-1030 procedure to change direction of mouse when it hits border.
1040-1170 procedure to display messages at end of game and wait for a key to be pressed before restarting game.
1180-1370 procedure to display instructions and controls.
1380-1480 procedure to test keys and, if appropriate, move cat.
1490-1690 procedure to set up user-defined characters, arrays and ENVELOPES for SOUND commands.
1700-1730 function to read character at screen location.
1740-1760 procedure to WAIT for any required number of hundredths of a second.

```
  10  MODE 1:DIM X(4),Y(4),R(4),Q(4),C(4)
  20  Z$="ØØØØØØ":HS$="ØØØØØ":HS=Ø
  30  PROCsetup
  40  PROCinstructions
  50  REPEAT
  60    CC=18
  70    PROCscreen
  80    PROCscat
  90    MC=Ø:SC=Ø
 100    REPEAT
 110      PROCdelay(1ØØ)
 120      PROCsmouse:MD=Ø:F=Ø
 130      OMX%=MX%:OMY%=MY%
 140      MX%=MX%+X(D):MY%=MY%+Y(D)
 150      CE=Ø:ME=Ø
```

```
  160       MC=MC+1:FOR J=1 TO 10:PROCdelay(10)
  170         COLOUR 129
  180         SOUND 2,-10,100,1:PRINTTAB(OMX%,
OMY%)" "
  190         COLOUR 130
  200         PROCdelay(10):SOUND 2,-10,50,1:P
RINTTAB(OMX%,OMY%)" ":NEXT
  210       COLOUR 128:PRINTTAB(OMX%,OMY%)" ":
COLOUR 3
  220       REPEAT
  230         PROCmove
  240         UNTIL ME=1 OR CE=1
  250       IF ME=1 PROCeaten
  260       IF CE=1 PROCmunch:CC=CC-1
  270       UNTIL CC=0
  280     PROCend
  290     UNTIL FALSE
  300   DEF PROCmove
  310   OMX%=MX%:OMY%=MY%:MX%=MX%+X(D):MY%=MY%
+Y(D):MD=MD+1
  320   IF FNPEEK(MX%+1,MY%)=32 AND RND(1)>.98
 THEN MX%=MX%+1
  330   PROCcat
  340   IF FNPEEK(MX%+1,MY%)=CHEESE AND FNPEEK
(MX%,MY%)<>CHEESE THEN MX%=MX%+1
  350   IF FNPEEK(MX%-1,MY%)=CHEESE AND FNPEEK
(MX%,MY%)<>CHEESE THEN MX%=MX%-1
  360   PROCcat
  370   IF FNPEEK(MX%,MY%)=CAT THEN ME=1:ENDPR
OC
  380   IF FNPEEK(MX%,MY%)=CHEESE THEN CE=1:EN
DPROC
  390   PROCcat
  400   IF FNPEEK(MX%,MY%)=EDGE THEN MX%=OMX%:
MY%=OMY%:PROCrebound:ENDPROC
  410   PROCcat
  420   IF F=1 THEN PRINTTAB(OMX%,OMY%)" "
  430   F=1
  440   PRINTTAB(MX%,MY%)CHR$(226)
  450   PROCcat
  460   ENDPROC
  470   DEF PROCscreen
```

```
  480  CLS
  490  COLOUR 1
  500  FOR X%=0 TO 39
  510    PRINT TAB(X%,0);CHR$(227);TAB(X%,1);
CHR$(227);TAB(X%,23);CHR$(227);TAB(X%,24);CHR
$(227)
  520    NEXT
  530  FOR Y%=2 TO 22
  540    PRINT TAB(0,Y%);CHR$(227);TAB(1,Y%);
CHR$(227);TAB(38,Y%);CHR$(227);TAB(39,Y%);CHR
$(227)
  550    NEXT
  560  COLOUR 2
  570  FOR J%=11 TO 13
  580    PRINT TAB(17,J%)STRING$(6,CHR$(224))
  590    NEXT
  600  COLOUR 0
  610  COLOUR 131
  620  PRINT TAB(6,0)"SCORE: 00000      HIGH:
";HS$
  630  COLOUR 3
  640  COLOUR 128
  650  ENDPROC
  660  DEF PROCsmouse
  670  REPEAT
  680    MX%=RND(36)+1
  690    IF RND(1)>.5 MY%=RND(7)+1 ELSE MY%=R
ND(7)+15
  700    D=RND(4)
  710    UNTIL FNPEEK(MX%+X(D),MY%+Y(D))=32
  720  ENDPROC
  730  DEF PROCscat
  740  REPEAT
  750    CX%=RND(36)+1:CY%=RND(21)+1
  760    UNTIL FNPEEK(CX%,CY%)=32
  770  PRINTTAB(CX%,CY%)CHR$(225)
  780  ENDPROC
  790  DEF PROCeaten
  800  PRINTTAB(OMX%,OMY%)" "
  810  SOUND 1,1,200,10
  820  S1=INT((SC+450-MD*5)/10)*10
  830  IF S1>SC SC=S1
```

```
  840  S$=STR$(SC)
  850  L=LEN(S$):IF L<5 THEN S$=LEFT$(Z$,5-L)
+S$
  860  COLOUR 0:COLOUR 131:PRINT TAB(6,0)"SCO
RE: ";S$;"      HIGH: ";HS$
  870  COLOUR 3:COLOUR 128:ENDPROC
  880  DEF PROCmunch
  890  COLOUR 2
  900  PRINTTAB(OMX%,OMY%)" "
  910  FOR CH%=229 TO 232
  920    PRINTTAB(MX%,MY%)CHR$(CH%)
  930    A=INKEY(10)
  940    NEXT
  950  COLOUR 3
  960  PRINTTAB(MX%,MY%)" "
  970  SOUND 1,2,30,50
  980  ENDPROC
  990  DEF PROCrebound
 1000  SOUND 1,-10,0,2
 1010  IF(MX%=37 OR MX%=2)AND(MY%=22 OR MY%=2
) D=C(D):ENDPROC
 1020  IF MX%=37 OR MX%=2 D=Q(D):ENDPROC
 1030  D=R(D):ENDPROC
 1040  DEF PROCend
 1050  REM
 1060  PROCdelay(400)
 1070  CLS:PRINTTAB(12,0)"*** GAME OVER ***"
 1080  PRINTTAB(5,2)"ALL THE CHEESE HAS BEEN
EATEN!"
 1090  PRINTTAB(12,7)"YOUR SCORE:";SC
 1100  IF SC>HS THEN HS=SC
 1110  HS$=STR$(HS)
 1120  IF LEN(HS$)<5 THEN HS$=LEFT$(Z$,5-LEN(
HS$))+HS$
 1130  PRINTTAB(12,9)"HIGH SCORE:";HS
 1140  PRINTTAB(6,12)"PRESS ANY KEY TO PLAY A
GAIN."
 1150  *FX 15,1
 1160  KEY=GET
 1170  ENDPROC
 1180  DEF PROCinstructions
```

```
 1190  PRINTTAB(17,0)"CHEESE";TAB(17,1)"_____
_"
 1200  PRINT''"THE OBJECT OF THE GAME IS TO P
REVENT"
 1210  PRINT'"THE MICE ('"CHR$226"') REACHING
 THE CHEESE."
 1220  PRINT''"YOU CONTROL A CAT ('"CHR$(225)
"') AND MAY MOVE IT"
 1230  PRINT'"WITH THE KEYS :(UP), /(DOWN), Z
(LEFT)"
 1240  PRINT'"AND X(RIGHT). TO CATCH A MOUSE
YOU MUST"
 1250  PRINT'"INTERCEPT IT, IN WHICH CASE IT
WILL RUN"
 1260  PRINT'"INTO YOUR JAWS."
 1270  PRINT''"PRESS SPACE TO CONTINUE..."
 1280  *FX 15,1
 1290  REPEAT UNTIL GET=32
 1300  CLS:PRINT"THE GAME ENDS WHEN ALL THE C
HEESE HAS"
 1310  PRINT'"BEEN EATEN. YOU SCORE POINTS FO
R EACH"
 1320  PRINT'"MOUSE YOU EAT: THE QUICKER YOU
EAT A"
 1330  PRINT'"MOUSE, THE MORE YOU SCORE FOR I
T."
 1340  PRINT''"PRESS SPACE TO BEGIN..."
 1350  *FX 15,1
 1360  REPEAT UNTIL GET=32
 1370  ENDPROC
 1380  DEF PROCcat
 1390  OCX%=CX%:OCY%=CY%
 1400  IF INKEY(-98) CX%=CX%-1:GOTO1450
 1410  IF INKEY(-67) CX%=CX%+1:GOTO1450
 1420  IF INKEY(-73) CY%=CY%-1:GOTO1450
 1430  IF INKEY(-105) CY%=CY%+1
 1440  IF CX%=OCX% AND CY%=OCY% ENDPROC
 1450  IF FNPEEK(CX%,CY%)<>32 CX%=OCX%:CY%=OC
Y%:ENDPROC
 1460  PRINT TAB(OCX%,OCY%)" "
 1470  PRINT TAB(CX%,CY%)CHR$(225)
 1480  ENDPROC
```

```
 149Ø  DEF PROCsetup
 15ØØ  VDU 23,224,&FF,&9F,&9F,&FF,&FF,&F9,&F9
,&FF
 151Ø  VDU 23,225,&82,&BA,&7C,&54,&EE,&AA,&44
,&3C
 152Ø  VDU 23,226,Ø,&1Ø,&ØC,&Ø8,&98,&5C,&38,Ø
 153Ø  VDU 23,227,24Ø,24Ø,24Ø,24Ø,15,15,15,15
 154Ø  VDU 23,228,&18,&7E,&FE,&FC,&FE,&7C,&3C
,&18
 155Ø  VDU 23,229,Ø,&1C,&3C,&78,&78,&7C,&3C,Ø
 156Ø  VDU 23,23Ø,Ø,Ø,&1C,&3C,&3C,&3C,&1Ø,Ø
 157Ø  VDU 23,231,Ø,Ø,&18,&1C,&1C,&18,Ø,Ø
 158Ø  VDU 23,232,Ø,Ø,&CØ,&18,&18,Ø,Ø,Ø
 159Ø  VDU 12,28,Ø,29,39,3
 16ØØ  VDU 23;1Ø,32;Ø;Ø;Ø;
 161Ø  CHEESE=128:CAT=129:EDGE=131
 162Ø  FOR J%=1 TO 4:READ X(J%),Y(J%):NEXT
 163Ø  DATA 1,-1,1,1,-1,1,-1,-1
 164Ø  R(1)=2:R(2)=1:R(3)=4:R(4)=3
 165Ø  Q(1)=4:Q(2)=3:Q(3)=2:Q(4)=1
 166Ø  C(1)=3:C(2)=4:C(3)=1:C(4)=2
 167Ø  ENVELOPE1,2,-1,-1,-1,255,255,255,12Ø,Ø
,Ø,-12Ø,12Ø,12Ø
 168Ø  ENVELOPE2,3,-4,-1,2,6,6,28,81,-4,-5,-1
,126,63
 169Ø  ENDPROC
 17ØØ  DEF FNPEEK(XC%,YC%)
 171Ø  VDU 31,XC%,YC%
 172Ø  A%=135
 173Ø  =(USR(&FFF4) AND &FFFF)DIV &1ØØ
 174Ø  DEF PROCdelay(WAIT)
 175Ø  TIME=Ø:REPEAT UNTIL TIME>WAIT
 176Ø  ENDPROC
```

GO-MOKU

by JEFF AUGHTON
from his original program

GENERAL DESCRIPTION

This is a straight-forward yet intriguing game played on an 11 by 11 colour grid, in which the player pits his wits against the computer.

The object of the game is to place five counters in a straight row, either horizontally, vertically or diagonally, before your opponent (the computer) does so. This requires careful thinking ahead, taking into account blocking moves as well as potential winning moves, making both defensive and offensive games possible.

Users with a 32K BBC B model with disc unit will need to alter the memory position by entering: PAGE=&E00 (RETURN) before typing in the listing.

DETAILED DESCRIPTION

Lines 10-90 defines colour modes used during the game.

110-120 PROC INITIAL - describes the game and gives the rules.

230-440 PROC READTABLE - reads in values from the data before the game begins.

460-550 PROC GAME - gives order of procedures used during the game.

570-660 PROC START - sets up characters and variables used in game.

680-900 PROC BOARD - sets up board for game on screen.

920-1070 PROC YOU - asks for player's move.

1090-1160 PROC INPUT - deals with input key after computer has asked a question.

1180-1200 PROC DEL - deals with deletion of entered variable.

1220-1240 PROC ADD - processes entered number in input procedure.

1260-1430 PROC ABUSE - displays error message if an impossible move is entered.
1450-1510 PROC ME - deals with computer's moves.
1530-1600 PROC GUESS - chooses computer's first move if allowed to start.
1620-1720 PROC THINK - consideration of computer's moves.
1740-1880 PROC EVAL - evaluates positions on board to help computer decide moves.
1900-2000 PROC PLAY - displays moves on screen.
2020-2220 PROC UPDATE - checks to see if either player has won.
2240-2410 PROC COMMENTS - displays final comments on screen after the game is over.
2430-2540 PROC SHOWIN - shows the winning line on the screen.
2560-2720 PROC OVER - displays total number of wins by each player and offers choice of another go.
2740-2960 PROC RULES - displays rules.
2980-3070 FN SQUARE - function for deciding which square you wish to play in.

```
>LIST
   10 REM GO-MOKU BY J.AUGHTON
   20 MODE7
   30 PROCinitial
   40 REPEAT
   50   MODE5:PROCgame
   60   MODE7: PROCover
   70   UNTIL Done
   80 PRINT
   90 END
  110 DEF PROCinitial
  120 PROCrules
  130 READW1%,W2%,W3%,W4%,W5%,W6%
  140 DIM Table% 2541,W%(252),Cell%(121)
  150 PROCreadtable
  160 VDU23,224,16,56,124,124,124,124,56
,16
  170 @%=2:Att%=W1%
  180 Mwin%=0:Ywin%=0:Draw%=0
```

```
  190 PRINT'"  PLEASE PRESS THE SPACE BA
R TO START"
  200 REPEAT UNTIL GET$=" "
  210 ENDPROC
  230 DEF PROCreadtable
  240 FOR I%=0 TO 2541
  250    ?(Table%+I%)=0:NEXT
  260 Win%=1
  270 FOR I%=1 TO 4
  280   READ P1%,P2%,P3%,P4%,P5%
  290   FOR J%=1 TO P2%
  300     FOR K%=1 TO 7
  310       X%=P1%
  320       FOR L%=1 TO 5
  330         Y%=Table%+21*X%-22
  340         REPEAT Y%=Y%+1
  350           UNTIL ?Y%=0
  360         ?Y%=Win%
  370         X%=X%+P3%
  380         NEXT
  390       P1%=P1%+P4%
  400       Win%=Win%+1
  410       NEXT
  420     P1%=P1%+P5%
  430     NEXT J%,I%
  440   ENDPROC
  460   DEF PROCgame
  470   PROCstart
  480   PROCboard
  490   REPEAT
  500     M%=1-M%
  510     IF M% THEN PROCme ELSE PROCyou
  520     IF N%>0 THEN PROCplay
  530     UNTIL M%>1
  540   PROCcomments
  550   ENDPROC
  570   DEF PROCstart
  580   Win%=252:M3s=0:Y3s=0:Err%=0:Firs
t=1:Done=0
  590   Lb%=34:Ub%=88
  600   VDU 19,2,2,0,0,0
  610   VDU 28,0,31,19,26
```

```
  620   FOR I%=1 TO Win%
  630     W%(I%)=0:NEXT
  640   FOR I%=1 TO 121
  650     Cell%(I%)=0:NEXT
  660   ENDPROC
  680    DEF PROCboard
  690    GCOL0,3:GCOL0,130
  700    COLOUR0:COLOUR 131
  710    CLG:CLS:VDU 5
  720   FOR I%=192 TO 1072 STEP 80
  730     MOVE I%,288:DRAW I%,992
  740     NEXT
  750   FOR I%=228 TO 992 STEP 64
  760     MOVE 192,I%:DRAW 1072,I%
  770     NEXT
  780   GCOL0,0
  790   FOR I%=65 TO 75
  800     MOVE 80*I%-5E3,264:VDU I%
  810     NEXT
  820   FOR I%=1 TO 11
  830     MOVE 32,1024-64*I%:PRINT I%
  840     NEXT:VDU 4
  850   REPEAT
  860     PRINTTAB(0,1)"WANT TO START?
";
  870     VDU 8:PROCinput(1,77,90)
  880     UNTIL In$="Y" OR In$="N"
  890   IF In$="Y" THEN M%=1 ELSE M%=0
  900   ENDPROC
  920   DEF PROCyou
  930   REPEAT
  940     CLS
  950     PRINT'"LETTER? ";
  960     PROCinput(1,64,76)
  970     X%=ASC(In$)-64
  980     REPEAT
  990       PRINTTAB(0,3)"NUMBER?    ";
 1000       VDU 8,8
 1010       PROCinput(2,47,58)
 1020       Y%=VAL(In$)
 1030       UNTIL Y%>0 AND Y%<12
 1040     N%=FNsquare
```

```
 105Ø      IF N%=Ø THEN PROCabuse
 106Ø      UNTIL N%
 107Ø    ENDPROC
 109Ø     DEF PROCinput(L%,Lo%,Hi%)
 11ØØ    K%=Ø:In$=""
 111Ø    REPEAT
 112Ø      Z%=GET
 113Ø      IF Z%=127 AND K%>Ø THEN PROCde
l
 114Ø      IF Z%>Lo% AND K%<L% AND Z%<Hi%
 THEN PROCadd
 115Ø      UNTIL Z%=13 AND K%>Ø
 116Ø    ENDPROC
 118Ø    DEF PROCdel
 119Ø    K%=K%-1:In$=LEFT$(In$,K%):VDU Z%
 12ØØ    ENDPROC
 122Ø    DEF PROCadd
 123Ø    K%=K%+1:In$=In$+CHR$(Z%):VDU Z%
 124Ø    ENDPROC
 126Ø    DEF PROCabuse
 127Ø    CLS:Err%=Err%+1
 128Ø    ON Err% GOSUB 132Ø,134Ø,136Ø,138
Ø,14ØØ,142Ø
 129Ø    PROCwait(35Ø)
 13ØØ    ENDPROC
 132Ø    PRINT'"You can't play there"'"It
's already taken":RETURN
 134Ø    PRINT'"That square is taken"'"Pl
ease try another":RETURN
 136Ø    PRINT'"That's your third"''"mist
ake . Think!!!!":RETURN
 138Ø    PRINT'"I'm getting tired"''"of t
his.Concentrate!":RETURN
 14ØØ    PRINT'"That was your fifth"''"id
iotic mistake...":RETURN
 142Ø    PRINT'"RIGHT!,THAT'S IT!!"''"YOU
 LOSE A TURN !!"
 143Ø    N%=-1:Err%=Ø:RETURN
 145Ø    DEF PROCme
 146Ø    CLS:PRINT'"Let me think..."
 147Ø    IF First THEN PROCguess ELSE PRO
Cthink
```

```
 1480   CLS:PRINT'"I WILL PLAY ";
 1490   VDU 64+X%,32,7:PRINT Y%
 1500   PROCwait(200)
 1510   ENDPROC
 1530   DEF PROCguess
 1540   First=0
 1550   REPEAT
 1560     X%=3+RND(5):Y%=3+RND(5)
 1570     N%=FNsquare
 1580     UNTIL N%
 1590   PROCwait(280)
 1600    ENDPROC
 1620   DEF PROCthink
 1630   IFM3s<=Y3s AND Y3s>1 THEN Def%=W
2% ELSE Def%=W3%
 1640   IF RND(1)<.9 THEN P1%=W4% ELSE P
1%=W5%
 1650   V%=-9999
 1660   FOR I%=Lb% TO Ub%
 1670     IF Cell%(I%)=0 THEN PROCeval
 1680     NEXT
 1690   N%=K%:K%=K%+10
 1700   X%=1+(K% MOD 11)
 1710   Y%=K% DIV 11
 1720   ENDPROC
 1740   DEF PROCeval
 1750   T%=0:Twos%=0:L%=Table%+21*I%-21
 1760   REPEAT
 1770     Z%=W%(?L%)
 1780     IF Z%=9 THEN 1840
 1790     IF Z%=4 THEN T%=9999:GOTO 1840
 1800     IF Z%=-4 THEN T%=999
 1810     IF Z%=-2 THEN Twos%=Twos%+1
 1820     IF Z%<0 THEN Q%=Def% ELSE Q%=A
tt%
 1830     T%=T%+Q%*(Z%+1)-P1%*(Z%=3)
 1840     L%=L%+1
 1850     UNTIL ?L%=0
 1860   IF Twos%>4 THEN T%=T%+W6%
 1870    IF T%>V% THEN V%=T%:K%=I%
 1880   ENDPROC
 1900   DEF PROCplay
```

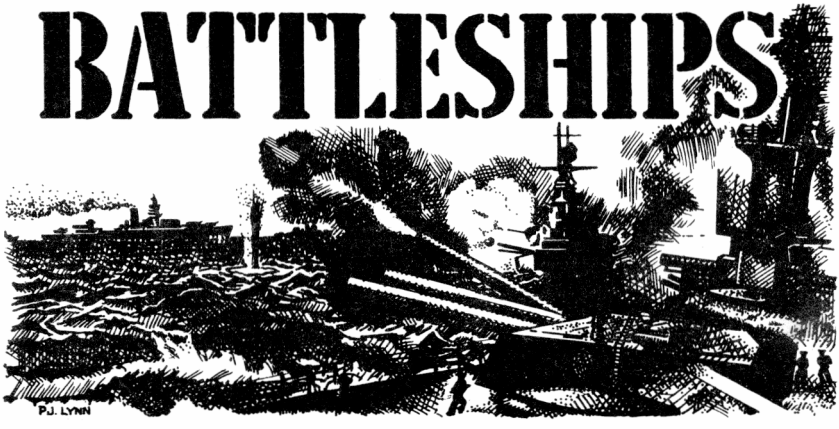
```
 1910    VDU 5
 1920    MOVE 128+80*X%,1036-64*Y%
 1930    FOR I%=1 TO 5
 1940      GCOL0,2:VDU 224,8
 1950      PROCwait(30)
 1960      GCOL0,M%:VDU 224,8
 1970      PROCwait(30)
 1980      NEXT:VDU 4
 1990    PROCupdate
 2000    ENDPROC
 2020    DEF PROCupdate
 2030    P%=2*M%-1:Cell%(N%)=P%
 2040     K%=Table%+21*N%-21
 2050    Q%=?K%
 2060    REPEAT
 2070      V%=W%(Q%)
 2080      IF V%=9 THEN 2150
 2090      IF V%*P%<0 THEN W%(Q%)=9:Win%=
Win%-1 ELSE W%(Q%)=V%+P%
 2100       IF V%=-3 THEN Y3s=Y3s-1
 2110      IF V%=+3 THEN M3s=M3s-1
 2120      IF V%=-2 AND P%<0 THEN Y3s=Y3s
+1
 2130      IF V%=+2 AND P%>0 THEN M3s=M3s
+1
 2140      IF ABS(W%(Q%))=5 THEN M%=(P%+5
)/2:Win%=Q%
 2150      K%=K%+1:Q%=?K%
 2160      UNTIL Q%=0
 2170    IF Win%=0 THEN M%=4
 2180    N%=N%-23
 2190    IF N%>0 AND N%<Lb% THEN Lb%=N%
 2200    N%=N%+46
 2210    IF N%<122 AND N%>Ub% THEN Ub%=N%
 2220    ENDPROC
 2240    DEF PROCcomments
 2250    IF M%<4 THEN PROCshowin
 2260    ON M%-1 GOSUB 2310,2350,2390
 2270    CLS:PRINTTAB(T%,1)A$
 2280     PROCwait(400)
 2290    ENDPROC
 2310    Ywin%=Ywin%+1:A$="You win!"
```

```
 2320    B$="Well played _ but you were l
ucky!!"
 2330    T%=5:RETURN
 2350    Mwin%=Mwin%+1:A$="I've won!!!!"
 2360    B$="WOW!! _ I really enjoyed tha
t!!!"
 2370    T%=4:RETURN
 2390     Draw%=Draw%+1:A$="It's a draw!!"
 2400    B$="No one could have won that!!"
 2410    T%=3:RETURN
 2430    DEF PROCshowin
 2440     X%=152+80*X%:Y%=1020-64*Y%
 2450    P1%=80:P2%=64
 2460    IF Win%<204 THEN P1%=-80
 2470    IF Win%<155 THEN P1%=0
 2480    IF Win%<78  THEN P1%=80:P2%=0
 2490    REPEAT
 2500      X%=X%+P1%:Y%=Y%+P2%
 2510      UNTIL POINT(X%,Y%)<>M%-2
 2520    MOVE X%-P1%,Y%-P2%
 2530    GCOL0,3:DRAW X%-5*P1%,Y%-5*P2%
 2540    ENDPROC
 2560    DEFPROCover
 2570     PRINT'B$
 2580    PRINT'"The score is now :"'
 2590    PRINT"     You  =";Ywin%
 2600    PRINT"     Me   =";Mwin%
 2610    PRINT"     Drawn=";Draw%
 2620    REPEAT
 2630      PRINTTAB(0,9)"Do you want to p
lay again?  ";
 2640      VDU 8:PROCinput(1,77,90)
 2650      UNTIL In$="Y" OR In$="N"
 2660    IF In$="N" THEN Done=1
 2670    ENDPROC
 2690    DEF PROCwait(W%)
 2700    T%=TIME
 2710    REPEAT UNTIL TIME>T%+W%
 2720    ENDPROC
 2740    DEF PROCrules
 2750    PRINTTAB(14)CHR$141;CHR$134"GO_M
OKU"
```

```
 2760   PRINTTAB(14)CHR$141;CHR$134"GU_H
URO"
 2770   PRINT'"  This game is similar to
 noughts_and-"
 2780   PRINT"crosses,the object being t
o create a"
 2790   PRINT"straight line of";CHR$131;
"FIVE OR MORE";CHR$135;"pieces"
 2800   PRINT"either horizontally,vertic
ally or"
 2810   PRINT"diagonally."
 2820   PRINT"  Notice that unlike some
versions of"
 2830   PRINT"this game,a winning line n
eed not"
 2840   PRINT"contain";CHR$129;"EXACTLY"
;CHR$135;"5 pieces and that I"
 2850   PRINT"treat any move into an uno
ccupied"
 2860   PRINT"square as legal_however ma
ny lines"
 2870   PRINT"of 3 are created."
 2880   PRINT"  My pieces are";CHR$130;"
RED";CHR$135;
 2890   PRINT"and yours are";CHR$132;"BL
ACK"
 2900   PRINT"(Oops,I seem to have got m
y colours a"
 2910   PRINT"bit mixed up !!!)"
 2920   PRINT"  You may choose who start
s but remember";
 2930   PRINT"that the first player has
an advantage"
 2940   PRINT"and so I expect to have fi
rst go at"
 2950   PRINT"least";CHR$133;"SOME";CHR$
135;"of the time !!!"
 2960   ENDPROC
 2980   DEF FNsquare
 2990   N%=11*Y%-11+X%
 3000   IF Cell%(N%) THEN N%=0
 3010   =N%
```

```
 3030    REM TPIDTPKO7BA,S.HTI_J.
 3050    DATA 10,-61,-20,220,55;135
 3070    DATA 1,11,1,1,4,1,11,11,11,-76,1
,7,12,1,4,5,7,10,1,4
```

BATTLESHIPS



JAMES BYRNE
Original program by John Edyvane

GENERAL DESCRIPTION

A version of the classic game of Battleships, in which your fleet is engaged in action with that of the computer. The shelling is accompanied by appropriate explode/splash sound effects!

After positioning its own fleet, the program asks you to set yours. This is done by specifying grid positions on the screen display. After this, you get to fire three rounds, followed by three for the computer, another three for you, and so on, until one or other's fleet has been consigned to Davy Jones' Locker! The display is continually updated to show the current state of events. Co-ordinates must be entered in form 'letternumber' with no spaces or commas.

DETAILED DESCRIPTION

Lines 10-100 main program, calling procedures to control the game.
110-420 procedure to set up various strings and constants.
430-580 procedure to produce screen display of battle-ocean.

590-880 procedure to position computer's fleet, and input positions for the user's fleet.
890-1210 procedure to control firing of shells and calculation of their effect.
1220-1340 procedure to input grid co-ordinates.
1350-1370 procedure to produce variable length delay.
1380-1640 procedures to fire computer's shells.
1650-1730 procedure to generate a "splash".
1740-1950 various small procedures.

```
LIST
   10  MODE 1
   20  PROCsetup
   30  PROCscreen
   40  PROCfleet
   50  REPEAT
   60    PROCfire
   70    UNTIL FNend=TRUE
   80  IF K%=2 THEN PRINT TAB(2,19)"YOU"
; ELSE PRINT TAB(2,19)"  I";
   90  PRINT" WIN."
  100  END
  110  DEF PROCsetup
  120  H%=0
  130  I%=0
  140  J%=0
  150  K%=2
  160  BX%=-1
  170  BY%=-1
  180  DIM A(2,10,10)
  190  DIM N(2,6)
  200  DIM N$(6)
  210  X=RND(-TIME)
  220  N$(1)="BATTLESHIP"
  230  N$(2)="CRUISER"
  240  N$(3)="DESTROYER 1"
  250  N$(4)="DESTROYER 2"
  260  N$(5)="SUBMARINE 1"
  270  N$(6)="SUBMARINE 2"
  280  FOR S%=1 TO 2
  290    N(S%,1)=5
  300    N(S%,2)=4
```

```
  310     N(S%,3)=3
  320     N(S%,4)=3
  330     N(S%,5)=2
  340     N(S%,6)=2
  350     NEXT
  360   B$=" ABCDEFGHIJ "
  370   ENVELOPE 1,2,-1,-1,-1,255,255,255
,120,0,0,-120,120,120
  380   ENVELOPE 2,3,0,0,0,1,1,1,120,-5,-
5,-10,120,90
  390   VDU 23,224,60,66,129,153,153,129,
66,60
  400   VDU 19,3,6;0;
  410   VDU 23;10,32;0;0;0;
  420   ENDPROC
  430   DEF PROCscreen
  440   VDU 28,5,27,35,5
  450   PRINT"   YOUR FLEET      ENEMY FLEET"
  460   COLOUR 0:COLOUR 130
  470   PRINT TAB(2,2)B$;TAB(18,2)B$
  480   C%=9
  490   FOR S%=3 TO 12
  500     PRINTTAB(2,S%);C%
  510     PRINTTAB(13,S%);C%
  520     PRINTTAB(18,S%);C%
  530     PRINTTAB(29,S%);C%
  540     C%=C%-1
  550     NEXT
  560   PRINT TAB(2,13)B$;TAB(18,13)B$
  570   COLOUR 2:COLOUR 128
  580   ENDPROC
  590   DEF PROCfleet
  600   PRINT TAB(2,15)"I AM POSITIONING
MY FLEET."
  610   FOR S%=1 TO 6
  620     L%=0
  630     PROCrnd
  640     REPEAT
  650       IF (P%+D%*L%<1 OR P%+D%*L%>10
) THEN L%=0:PROCrnd:UNTIL FALSE
  660     IF (Q%+U%*L%<1 OR Q%+U%*L%>10)
THEN L%=0:PROCrnd:UNTIL FALSE
```

```
  670     IF A(2,P%+D%*L%,Q%+U%*L%)<>0 TH
EN L%=0:PROCrnd:UNTIL FALSE
  680     L%=L%+1
  690     UNTIL L%=N(1,S%)
  700     FOR L%=0 TO N(1,S%)-1
  710       A(2,P%+D%*L%,Q%+U%*L%)=S%
  720       NEXT
  730     NEXT
  740   PRINT TAB(2,15)"ENTER YOUR SHIPS
NOW.      "
  750   FOR S%=1 TO 6
  760     PRINT TAB(2,17);N$(S%);"     "
  770     FOR L%=1 TO N(2,S%)
  780       PRINT TAB(2,19)"SQUARE ";L%
  790       PROCinput
  800       IF A(1,X%,Y%)<>0 PROCinput
  810       A(1,X%,Y%)=S%
  820       PRINT TAB(X%+2,13-Y%)LEFT$(N$
(S%),1)
  830       NEXT
  840     NEXT
  850   PRINT TAB(2,15)SPC(26)
  860   PRINT TAB(2,17)SPC(12)
  870   PRINT TAB(2,19)SPC(15)
  880   ENDPROC
  890   DEF PROCfire
  900   IF K%=1 THEN PRINT TAB(2,15);"MY
GO  "
  910   IF K%=2 THEN PRINT TAB(2,15);"YOU
R GO"
  920   FOR G%=1 TO 3
  930     PRINT TAB(2,17)SPC(24)
  940     PRINT TAB(2,16)"FIRE ";G%
  950     PROCdelay(100)
  960     REPEAT
  970       IF K%=1 PROCcomfire
  980       IF K%=2 PROCinput
  990       Z%=A(K%,X%,Y%)
 1000       UNTIL Z%<>-1
 1010     A(K%,X%,Y%)=-1
 1020     SOUND 1,1,200,50
 1030     PROCdelay(260)
```

```
 1040    IF Z%=0 PROCsplash:NEXT:PROCswa
p:ENDPROC
 1050  COLOUR 1
 1060  PRINT TAB(X%+(K%*16)-14,13-Y%);"*
"
 1070  PRINT TAB(2,17)"BANG!";SPC(15)
 1080  SOUND 0,2,6,10
 1090  PROCdelay(100)
 1100  COLOUR 2
 1110  IF K%=1 PROCa
 1120  N(K%,Z%)=N(K%,Z%)-1
 1130  IF N(K%,Z%)<>0 NEXT:PROCswap:ENDP
ROC
 1140  COLOUR 1
 1150  PRINT TAB(2,17)N$(Z%);" SUNK"
 1160  IF K%=1 H%=0:BX%=-1:BY%=-1
 1170  PROCdelay(200)
 1180  COLOUR 2
 1190  NEXT
 1200  PROCswap
 1210  ENDPROC
 1220  DEF PROCinput
 1230  REPEAT
 1240    REPEAT
 1250      *FX 15,0
 1260      PRINT TAB(2,21)SPC(28)
 1270      INPUT TAB(2,21)"WHICH CO-ORDS
",A$
 1280      PRINT TAB(2,21)SPC(28)
 1290      UNTIL LEN(A$)=2
 1300    X%=ASC(A$)-64
 1310    A$=RIGHT$(A$,1)
 1320    Y%=VAL(A$)+1
 1330    UNTIL ASC(A$)<58 AND Y%>0 AND Y
%<11 AND X%>0 AND X%<11
 1340  ENDPROC
 1350  DEF PROCdelay(T)
 1360  TIME=0:REPEAT UNTIL TIME>T
 1370  ENDPROC
 1380  DEF PROCcomfire
 1390  IF H%<>0 AND J%<>0 P%=H%:Q%=I%:PR
OCb:ENDPROC
```

```
 1400  IF H%<>0 PROCb:ENDPROC
 1410  REPEAT
 1420    X%=RND(10)
 1430    Y%=RND(10)
 1440    UNTIL A(1,X%,Y%)>=0
 1450  IF BX%<>-1 X%=BX%:Y%=BY%
 1460  ENDPROC
 1470  DEF PROCa
 1480  I%=Y%
 1490  IF K%=1 H%=X%:J%=1
 1500  ENDPROC
 1510  DEF PROCb
 1520  BX%=-1:BY%=-1
 1530  FOR D=-1 TO 1
 1540    FOR U=-1 TO 1
 1550      X%=U+P%
 1560      IF X%<1 OR X%>10 NEXT:NEXT:OH
%=H%:H%=0:PROCcomfire:H%=OH%:ENDPROC
 1570  Y%=D+Q%
 1580  IF Y%<1 OR Y%>10 NEXT:NEXT:OH%=H%
:H%=0:PROCcomfire:H%=OH%:ENDPROC
 1590  IF A(1,X%,Y%)<0 NEXT:NEXT:OH%=H%:
H%=0:PROCcomfire:H%=OH%:ENDPROC
 1600  BX%=X%:BY%=Y%
 1610  NEXT
 1620  NEXT
 1630  X%=BX%:Y%=BY%
 1640  ENDPROC
 1650  DEF PROCsplash
 1660  COLOUR 3
 1670  PRINT TAB(X%+(K%*16)-14,13-Y%);CH
R$(224)
 1680  PRINT TAB(2,17)"SPLASH";SPC(16)
 1690  SOUND 0,2,4,10
 1700  COLOUR 2
 1710  IF K%=1 J%=0
 1720  PROCdelay(100)
 1730  ENDPROC
 1740  DEF PROCswap
 1750  IF K%=1 K%=2:ENDPROC
 1760  IF K%=2 K%=1
 1770  ENDPROC
```

```
1780  DEF FNend
1790  FOR C%=1 TO 2
1800    PROCswap
1810    FL=TRUE
1820    FOR S%=1 TO 6
1830      IF N(K%,S%)>0 FL=FALSE
1840      NEXT
1850    IF FL=TRUE THEN =FL
1860    NEXT
1870  =FL
1880  DEF PROCrnd
1890  REPEAT
1900    D%=RND(3)-2
1910    U%=RND(3)-2
1920    UNTIL D%<>0 OR U%<>0
1930  P%=RND(10)
1940  Q%=RND(10)
1950  ENDPROC
```

DUCKSHOOT

BRIAN JONES and J.A. MARIS
Original program by B. Philips

GENERAL DESCRIPTION

This game allows you to play the part of an early morning duck hunter. When the sun comes up all the ducks will fly away, so you have to shoot the sun back down when it rises. You can move left and right among the rushes at the bottom of the screen using the G and J keys. H is used to fire. Full instructions can be obtained on running the game.

DETAILED DESCRIPTION

Lines 10-330 the main structure which sets up the screen, time values and sound, and moves the ducks.
340-420 PROC ENABLE - allows the keyboard to be read and moves gun if necessary.
430-550 PROC FIRE - moves bullet up screen and checks for collisions.
560-650 PROC SUN - sets the conditions for the sun, its position and duration, and allows you to shoot at it.
660-780 PROC DIE - moves dead ducks down to ground and adds to score.
790-900 PROC BEAST - prints final message, including choice of another go.
910-950 PROC SHOTSUN - registers hits on sun.
960-1160 PROC INSTRUCT - display instructions.
1170-1230 PROC OUT - out of time message.
1240-1390 PROC DEFINE - defines the characters used, for example CHR$(228) is man.

```
LIST
   1Ø REM **DUCK SHOOT**SOUTHERN SOFTWAR
E**198Ø         VØ.5
   2Ø REM **TRANSLATED FROM SHARP BASIC*
*M & T**1983
   3Ø MODE7:CLS:PRINTTAB(14,5)"DUCK SHOO
T"TAB(6,12)"DO YOU WANT INSTRUCTIONS ?"T
AB(14,15)"'Y' OR 'N'"
   4Ø A$=GET$:IFA$<>"Y"ANDA$<>"N"THEN4Ø
   5Ø IFA$="Y"THENPROCinstruct
   6Ø MODE1:CLS:COLOUR 1:PRINTTAB(14,2)"
DUCK SHOOT":COLOUR 3:PROCdefine:SUN=FALS
E
   7Ø TIME=Ø:VDU23;8202;Ø;Ø;Ø;:*FX11,Ø
   8Ø MX%=19:MY%=26:HI%=1:D%=Ø:VV%=Ø:V%=
Ø:SS%=Ø:C%=Ø:DX%=Ø:DY%=Ø
   9Ø FORI%=ØTO39:PRINTTAB(I%,26)CHR$(22
3+RND(2)):NEXT
  1ØØ FORI%=ØTO38STEP2:PRINTTAB(I%,27)CH
R$226TAB(I%+1,27)CHR$227:NEXT
  11Ø FORI%=ØTO38STEP2:PRINTTAB(I%,28)CH
R$238TAB(I%+1,28)CHR$239:NEXT
  12Ø PRINTTAB(MX%,MY%)CHR$228
  13Ø SD%=229:DM%=-1:DS%=RND(1Ø)+7
  14Ø IFTIME>18ØØØTHENPROCout:PROCbeast:
CLEAR:GOTO6Ø
  15Ø IFRND(1)>.8THENPROCsun
  16Ø IFRND(1)>.6GOTO24Ø
  17Ø FORI%=39TOØSTEP-1
  18Ø   IFRND(1)>.6THENSOUND&11,-15,5,1
  19Ø   PRINTTAB(I%,DS%)CHR$229:P%=I%
  2ØØ   TT%=TIME:REPEAT UNTIL TIME>TT%+5
  21Ø   PROCenable:IFC%=39THENPROCY:CLEA
R:GOTO6Ø ELSEPRINTTAB(I%,DS%)" "
  22Ø   IFV%=1THENV%=Ø:HI%=HI%+1:I%=Ø
  23Ø   NEXT:D%=D%+1:GOTO13Ø
  24Ø SD%=23Ø:DM%=1:DS%=RND(1Ø)+7
  25Ø IFRND(1)<.6GOTO13Ø
  26Ø FORI%=ØTO39
  27Ø   IFRND(1)<.4THENSOUND&11,-15,81,1
  28Ø   PRINTTAB(I%,DS%)CHR$23Ø:P%=I%
  29Ø   TT%=TIME:REPEAT UNTIL TIME>TT%+5
```

```
  300    PROCenable:IFC%=39THENPROCY:CLEA
R:GOTO60 ELSEPRINTTAB(I%,DS%)" "
  310    IFV%=1THENV%=0:HI%=HI%+1:I%=39
  320    NEXT:D%=D%+1:GOTO240
  330
  340 DEFPROCenable
  350 K$=INKEY$(0):ML%=MX%:*FX15,1
  360 IFMX%=39THENMX%=MX%-1:PRINTTAB(39,
26)CHR$(223+RND(2))
  370 IFMX%=0THENMX%=MX%+1:PRINTTAB(0,26
)CHR$(223+RND(2))
  380 IFK$="G"THENMX%=MX%-1:PRINTTAB(ML%
,26)CHR$(223+RND(2))TAB(MX%,26)CHR$(228)
  390 IFK$="J"THENMX%=MX%+1:PRINTTAB(ML%
,26)CHR$(223+RND(2))TAB(MX%,26)CHR$(228)
  400 IF K$="H" THEN PROCfire
  410 ENDPROC
  420
  430 DEFPROCfire
  440 J1%=25:FORJ%=24TO8STEP-1
  450    PRINTTAB(MX%,J1%)" "
  460    PRINTTAB(MX%,J%)CHR$235:J1%=J%
  470    IFSUN=TRUE THEN500
  480    IFMX%=I%ANDJ%=DS%THENV%=1:PRINTT
AB(MX%,J%)" ":DS%=0:J%=8:PRINTTAB(MX%,25
)" "
  490    GOTO510
  500    IFMX%=DX%ANDJ%=DY%THENPRINTTAB(D
X%,DY%)CHR$236:PRINTTAB(MX%,J%)" ":SOUND
&11,-15,101,3:SOUND1,-15,89,1:SOUND1,-15
,185,1:VV%=1:SS%=SS%+1:PRINTTAB(MX%,25)"
 "
  510    NEXT:IFV%=1THENPROCdie
  520 IFVV%=1THENPROCshotsun
  530 PRINTTAB(MX%,J1%)" "TAB(MX%,J%)" "
:SOUND&11,-15,5,20:PRINTTAB(MX%,25)" "
  540 ENDPROC
  550
  560 DEFPROCsun:*FX12,0
  570 SUN=TRUE : COLOUR 2
  580 DX%=RND(38):DY%=RND(10)+7:PRINTTAB
(DX%,DY%)CHR$237
```

```
  590 COLOUR 3 :  I1%=RND(2500):FORI%=1TO
I1%
  600    PROCenable
  610    IFVV%=1THENVV%=0:I%=I1%
  620    NEXT:*FX11,0
  630 PRINTTAB(DX%,DY%)"  ":DX%=0:DY%=0
  640 SUN=FALSE:ENDPROC
  650
  660 DEFPROCdie:ML%=MX%:COLOUR 1
  670 FORY%=J1%-1 TO25:E=RND(1):ML%=ML%+ DM%
  680    IFE>.7THENML%=ML%+1:SD%=240
  690    IFE<.5THENML%=ML%-1:SD%=241
  700    IFE<.3THENML%=ML%+4:SD%=240
  710    IFML%<0THENML%=0
  720    IFML%>39THENML%=39
  730    PRINTTAB(ML%,Y%)CHR$(SD%):FORI%=
1TO200:NEXT:SOUND&11,-15,129,1:PRINTTAB(
ML%,Y%)"  ":NEXT
  740 C%=C%+1
  750 FORX%=0TO39-C%:PRINTTAB(X%,28)CHR$
229:PRINTTAB(X%,28)"  ":NEXT
  760 PRINTTAB(X%,28)CHR$229
  770 COLOUR 3:ENDPROC
  780
  790 DEFPROCbeast:CLS
  800 PRINTTAB(10,1)"*    * ***** *    *"T
AB(10,2)"*    * *    * *    *"TAB(10,3)"*
 * *    * *    *"TAB(10,4)" * *  *    * *
 *"TAB(10,5)"  *    *    * *    *"TAB(10,6)
"  *    ***** *****"
  810 PRINTTAB(4,9)"****  *****    *    **
*** *****"TAB(4,10)"*    * *       * *  *
       *"TAB(4,11)"*    * *      *    * *
     *"TAB(4,12)"***** ****  *****  ***
  *"TAB(4,13)"*    * *      *    *      *    *"
  820 PRINTTAB(4,14)"*    * *      *    *
   *    *"TAB(4,15)"****  ***** *    * ***
**    *"
  830 PRINTTAB(4,18)"YOU KILLED ";HI%-1;
" POOR LITTLE DUCKS !"
  840 B%=(HI%-1)*10:B1%=SS%*25:B2%=B%+B1
%+((HI%-1)/5)*15
```

```
  850 PRINTTAB(4,20)"BUT FORTUNATELY ";D
%-HI%;" GOT AWAY !"
  860 PRINTTAB(4,22)"YOU PUT OUT THE SUN
 ";SS%;" TIMES "
  870 PRINTTAB(4,24)"YOUR SCORE IS ";B2%
  880 PRINTTAB(4,26)"ANY KEY FOR ANOTHER
 GO.":*FX15,0
  890 IFGET$<>""THENENDPROC ELSE890
  900
  910 DEFPROCshotsun
  920 COLOUR 2 : FORX%=0TO39-SS%:PRINTTA
B(X%,27)"*":PRINTTAB(X%,27)" ":NEXT
  930 PRINTTAB(X%,27)"*" : COLOUR 3
  940 ENDPROC
  950
  960 DEFPROCinstruct:CLS:PRINT"**DUCK S
HOOT .........Instructions**":PRINT
  970 PRINT"This is a wicked game of ski
ll at"
  980 PRINT"Duck Shooting."
  990 PRINT"You are stomping through the
 marsh-"
 1000 PRINT"land waiting for the poor un
suspecting"
 1010 PRINT"duck to fly over."
 1020 PRINT"Every duck you shoot is laid
 on the"
 1030 PRINT"ground beneath you, in order
 that you"
 1040 PRINT"may count your spoils."
 1050 PRINT"When the sun comes out, the
ducks do"
 1060 PRINT"not fly. You may then take a
 potshot at"
 1070 PRINT"the sun to gain extra points
."
 1080 PRINT"Every 5 ducks earns extra 15
points."
 1090 PRINT"Every sun earns 25 points.":
PRINT
 1100 PRINT"Keys 'G' and 'J' move you ar
ound."
```

```
 1110 PRINT"Key 'H' fires your gun."
 1120 PRINT"YOUR TIME IS LIMITED.So shoo
t carefully."
 1130 PRINT"              Any key to start
."
 1140 REPEAT UNTIL GET$<>""
 1150 ENDPROC
 1160
 1170 DEFPROCout:CLS
 1180 PRINTTAB(11,5)" **   *   *  *****"
TAB(11,6)"*  *  *   *    *"TAB(11,7)"*
*  *   *    *"TAB(11,8)" **    ***     *
"
 1190 PRINTTAB(11,11)"OF TIME !!!!!!!!!"
 1200 PRINTTAB(11,14)"THE SUN HAS RISEN"
TAB(11,16)"AND THE DUCKS ARE"TAB(11,18)"
AWAY ON THE WATER."
 1210 FORI=1TO10000:NEXT
 1220 ENDPROC
 1230
 1240 DEFPROCdefine
 1250 VDU23,224,128,64,32,16,8,4,2,1
 1260  VDU23,225,1,2,4,8,16,32,64,128
 1270 VDU23,226,73,146,36,73,146,36,73,1
46
 1280 VDU23,227,146,73,36,146,73,36,146,
73
 1290 VDU23,228,56,56,16,254,16,40,68,13
0
 1300 VDU23,229,3,6,12,255,12,6,3,0
 1310 VDU23,230,192,96,48,255,48,96,192,
0
 1320 VDU23,235,0,0,24,60,60,24,0,0
 1330 VDU23,236,0,24,60,90,126,36,24,0
 1340 VDU23,237,24,36,66,165,129,90,36,24
 1350 VDU23,238,255,128,128,128,128,128,
128,128
 1360 VDU23,239,255,1,1,1,1,1,1,1
 1370 VDU23,240,24,24,255,126,24,24,24,24
 1380 VDU23,241,24,24,24,24,126,255,24,24
 1390 ENDPROC
```

By SIMON WILLIAMS
from his original program

GENERAL DESCRIPTION

This is a very long program (over 15K), but don't worry, most of it is text and spelling mistakes are easy to debug.

The program uses a pack of Tarot cards to reveal your fortune and guide you to greater success in your endeavours. Ten cards are dealt, and interpreted according to their meaning and position.

DETAILED DESCRIPTION

Lines 10-80 dimension an array. Put up a title page and wait 5 seconds (the printed characters must be copied exactly as they will be converted to Teletext graphics on the screen).

90-260 print the instructions.

270-400 input and error check your sex and age and work out your type.

410-490 use your type and your hair colour to work out your 'Significator'.

500-570 print your 'Significator' and finish the instructions.

580-710 lay out the cards, number them and seed the random number generator until you are ready.
720-750 do the reading using PROCreadcard.
760-840 ask if you want another reading, if yes go round again, if no then end.
1000-1040 put Teletext control codes on the screen to control colour, graphics etc.
2000-2030 wait for T% hundredths of a second.
3000-3030 put a 'TAROT' graphic on the screen.
4000-4290 interpret the cards.
4010-4060 set the meaning of the card's position.
4070-4140 select a card and check that it has not been used already.
4150-4290 print the name of the card and interpret it, and go back to the main loop.
5000-5880 are DATA statements for interpreting the cards.
The rest of the program may be re-numbered but because of the way lines 4010 and 4150 work, this section must be numbered exactly as shown. Some of the text is enclosed in double quotes ("), this is because it contains a comma which the BASIC would otherwise interpret as separating two DATA statements.

```
    1 REM            TAROT
    2 REM          (c)  1983
    3 REM        S. Williams
    4 REM
   1Ø MODE 7: VDU23;11,Ø;Ø;Ø;Ø
   2Ø DIM CARD%(1Ø)
   3Ø PROCcodes(Ø,1,6,19,147,32,32)
   4Ø PRINT TAB(31,6)"j5" TAB(4,7)"*/////?/
/////" TAB(31,7)"j5" TAB(1Ø,8)"5" TAB(29,8)
"_pzuppØ" TAB(8,9)"_>5" TAB(31,9)"j5"
   5Ø PRINT TAB(7,1Ø)"x7 5 _pp  _" TAB(27,1
Ø)"pØ  j5" TAB(5,11)"'  5 '  i  k48,Ø 6 "
"4 j5" TAB(4,12)"h^?   5   _z5 j?  'j   j j
5"
   6Ø PRINT TAB(4,13)"%   5 x'!j5 j5
  j5j5" TAB(3,14)    5j5  j5 j5   o   j
%j5" TAB(3,15)Ø   5 +pp:eØj5    epp% ""
-pp&"
```

```
   70 PRINT TAB(4,16)"5   5" TAB(5,17)"+)
Ø  5   Ø" TAB(6,18)"""/){=,''";CHR$134;"(
c) 1983 S.Williams" TAB(11,19)5"
   80 PROCpause(500)
   90 PROCheading(13)
  100 PROCcodes(Ø,1,4,21,134,32,32)
  110 VDU28,1,24,38,4,30
  120 PRINT"The Tarot is an ancient method of
     fortune telling which makes use of a  pa
ck of 78 cards. These are divided   into two
sections, the major and minor'Arcanas'."
  130 PRINT'"The major arcana is a set of 22
pict- orial cards depicting people and      o
bjects. The minor arcana is similar"
  140 PRINT"to an ordinary pack of playing ca
rds  with an extra card, the 'page', in    ea
ch suit. The four suits are named   Wands, Cu
ps, Swords and Pentacles."
  150 PRINT'"The major and minor arcanas are
      shuffled together and dealt out, face d
own, in one of several arrangements  known as
 'lays'."
  160 PRINT''CHR$133;"    Press space bar to
continue";
  170 *FX15,1
  180 REPEAT UNTIL GET=32
  190 REPEAT
  200   VDU26
  210   PROCheading(13)
  220   PROCcodes(Ø,1,4,14,134,32,32)
  230   VDU28,1,24,38,4,30
  240   PRINT"The cards are revealed one at a
 time, and read according to their individ-
ual meanings and their positions in"
  250   PRINT"relation to the other cards. I
shall  read from the ten card or 'Celtic
Cross' lay."
  260   PRINT'"Before we start, I need to kno
w some  things about you in order to assign
 you one of the court cards as your    'Signi
ficator'."
  270   PRINT'"Are you male or female (M/F)?"
```

```
  280   *FX15,1
  290   REPEAT: SEX$=GET$: UNTIL SEX$="M" OR
SEX$="F"
  300   PRINT TAB(31,12)SEX$
  310   PRINT'"Are you over 40 (Y/N)?- be hon
est!"
  320   *FX15,1
  330   REPEAT: AGE$=GET$: UNTIL AGE$="Y" OR
AGE$="N"
  340   PRINT TAB(36,14)AGE$
  350   PRINT'"What colour is your hair (1-5)
?"''"1-Very fair   2-Gray  3-Light brown   4-
Dark brown  5-Black"
  360   *FX15,1
  370   REPEAT: HAIR$=GET$: UNTIL VAL(HAIR$)>
0 AND VAL(HAIR$)<6
  380   PRINT TAB(33,16)HAIR$
  390   HAIR%=VAL(HAIR$)
  400   TYPE$=SEX$+AGE$
  410   IF TYPE$="MY" CARD$="King of ": CARD%
=14
  420   IF TYPE$="FY" CARD$="Queen of ": CARD
%=13
  430   IF TYPE$="MN" CARD$="Knight of ": CAR
D%=12
  440   IF TYPE$="FN" CARD$="Page of ": CARD%
=11
  450   IF HAIR%<3 SUIT$="Wands": CARD%=CARD%
+64
  460   IF HAIR%=3 SUIT$="Cups": CARD%=CARD%+
50
  470   IF HAIR%=4 SUIT$="Swords": CARD%=CARD
%+36
  480   IF HAIR%=5 SUIT$="Pentacles": CARD%=C
ARD%+22
  490   SIG$=CARD$+SUIT$: CARD%(0)=CARD%
  500   VDU26
  510   PROCcodes(0,1,20,23,134,32,32)
  520   VDU28,1,24,38,16,12
  530   PRINT"Thank you. Your significator is
:"''"The ";SIG$
  540   PRINT'"This card is placed face up on
```

```
 the    table and the first two cards of the
 lay are placed on top of it as they   are de
alt."
  550   PRINT CHR$133;"    Press space bar to
 continue";
  560   *FX15,1
  570   REPEAT UNTIL GET=32
  580   VDU26,12
  590   PROCheading(8)
  600   PROCcodes(28,30,2,5,135,157,130)
  610   PROCcodes(9,20,7,10,157,130,156)
  620   PROCcodes(28,30,7,10,135,157,130)
  630   PROCcodes(0,2,12,15,157,130,32)
  640   PROCcodes(9,20,17,20,157,130,156)
  650   PROCcodes(28,30,17,20,135,157,130)
  660   PROCcodes(0,1,22,24,133,32,32)
  670   PRINT TAB(34,3)"10"; TAB(14,8)"3"; TA
B(34,8)"9"; TAB(5,13)"5"; TAB(14,13)"1"; TAB(
24,13)"6"; TAB(34,13)"8"; TAB(14,18)"4"; TAB(
34,18)"7";
  680   VDU28,1,24,39,22,30
  690   PRINT"  Please concentrate on any mat
ter of   concern before pressing the space ba
r           to begin the reading";
  700   *FX15,1
  710   REPEAT: D%=RND(-TIME): UNTIL GET=32
  720   VDU28,0,24,39,21,12,26
  730   FOR M%=1 TO 10
  740     PROCreadcard
  750     A=GET: NEXT
  760   PROCcodes(0,1,21,24,133,32,32)
  770   VDU28,1,24,39,21,12
  780   PRINT'"       Press A for another read
ing                    or Q to quit"
  790   *FX15,1
  800   REPEAT E%=GET: UNTIL E%=65 OR E%=81
  810   FOR N%=0 TO 10: CARD%(N%)=0: NEXT
  820   UNTIL E%=81
  830 MODE 7
  840 END
  999
```

```
 1000 DEF PROCcodes(X1%,X2%,Y1%,Y2%,C1%,C2%,C
3%)
 1010 FOR Y%=Y1% TO Y2%
 1020   PRINT TAB(X1%,Y%)CHR$(C1%);CHR$(C2%);
 TAB(X2%,Y%)CHR$(C3%);
 1030   NEXT
 1040 ENDPROC
 1999

 2000 DEF PROCpause(T%)
 2010 TIME=0
 2020 REPEAT UNTIL TIME>T%
 2030 ENDPROC
 2999

 3000 DEF PROCheading(X%)
 3010 CLS
 3020 PRINT TAB(X%,0)CHR$147;"ppp0      h" TAB
(X%,1)CHR$147;"hk ,0hd_dk!" TAB(X%,2)CHR$147;
"*z$wej *:*8"
 3030 ENDPROC
 3999

 4000 DEF PROCreadcard
 4010 RESTORE M%*10+5000
 4020 READ X%,Y%,IND$
 4030 VDU28,0,5,27,4,12,28,0,24,39,21,12,26
 4040 IF M%=2 VDU28,X%,Y%+4,X%+8,Y%,12,26: PR
INT TAB(14,13)"2";: PROCpause(150)
 4050 PRINT TAB(X%+1,Y%+1)CHR$136; TAB(X%+6,Y
%+1)CHR$137;
 4060 PROCpause(300)
 4070 REPEAT
 4080   CARD%=RND(156): FL%=0
 4090   IF CARD%>78 R%=TRUE: CARD%=CARD%-78 E
LSE R%=FALSE
 4100   FOR N%=0 TO 10
 4110     IF CARD%=CARD%(N%) FL%=1
 4120     NEXT
 4130   UNTIL FL%=0
 4140 CARD%(M%)=CARD%
 4150 RESTORE CARD%*10+5100
```

```
 4160 READ L1$,L2$,L3$,L4$,MEAN$
 4170 IF R% READ MEAN$
 4180 PRINT TAB(0,4)"This card indicates:"; T
AB(0,5)IND$;SPC(27-LEN(IND$));
 4190 PROCpause(200)
 4200 VDU28,X%,Y%+4,X%+8,Y%,12,26
 4210 PRINT TAB(X%,Y%)L1$;
 4220 PRINT TAB(X%,Y%+1)L2$;
 4230 PRINT TAB(X%,Y%+2)L3$;
 4240 PRINT TAB(X%,Y%+3)L4$;
 4250 PROCpause(150)
 4260 IF R% PRINT TAB(0,21)"This";CHR$136;"re
versed";CHR$137;"card implies:" ELSE PRINT TA
B(0,21)"This card implies:           "
 4270 VDU28,0,24,39,22,12,26
 4280 PRINT TAB(0,22)MEAN$; TAB(0,24)CHR$133"
     Press space bar to continue";
 4290 ENDPROC
 4999

 5000 REM        *** ALL DATA ***
 5010 DATA 11,12,Your general surroundings
 5020 DATA 11,12,Current problems
 5030 DATA 11,7,Your aims and ideals
 5040 DATA 11,17,The background situation
 5050 DATA 2,12,What is behind you
 5060 DATA 20,12,What is ahead of you
 5070 DATA 31,17,Your attitude to the matter
 5080 DATA 31,12,Your current environment
 5090 DATA 31,7,Your hopes and fears
 5100 DATA 31,2,The final outcome
 5110 DATA "",THE,MAGICIAN,"","Either the nee
d for, or the skills of,  diplomacy.",Disquie
t and a feeling that all is not  right.
 5120 DATA "",THE,HIGH,PRIESTESS,"Mystery and
 the future, or a woman with psychic powers."
,The physical side of love - passion. A  very
 strong card.
 5130 DATA "",THE,EMPRESS,"",Fruitfullness in
 all its associations - initiative.,Light and
 truth in matters. Complex      problems revea
led.
```

```
 514Ø DATA "",THE,EMPEROR,"",Stability in lif
e. The aid of a          powerful person.,Bene
volence of others towards you.
 515Ø DATA "",THE,HIERO-,PHANT,Associations w
ith others and             particularly marria
ge.,An increase in your understanding of    s
ociety.
 516Ø DATA "",THE,LOVERS,"",All forms of love
 and attraction to     beautiful things.,Cert
ain projects may be foolishly        thought o
ut.
 517Ø DATA "",THE,CHARIOT,"",Triumph only thr
ough perseverence and   doggedness.,A dispute
. Possible legal complications.
 518Ø DATA "",STRENGTH,"","",The just outcome
 of problems.,A tendency to bias. Watch again
st        bigotry.
 519Ø DATA "",THE,HERMIT,"",Corruption. Other
s may try to betray     you.,An incautious att
itude. Think things out- then act.
 52ØØ DATA THE,WHEEL,OF,FORTUNE,"The forces o
f luck, and therefore          success.",Abundan
ce of the good things in life.   Enjoy them.
 521Ø DATA "",JUSTICE,"","",A time for taking
 courage and acting     energetically.,An abus
e of power through weakness.
 522Ø DATA "",THE,HANGED,MAN,Intuition - your
 own or advice from one who has.,Obstruction
through selfishness. Take    heed.
 523Ø DATA "",DEATH,"","",The end of a cycle
of events. Change andrenewal.,Lethargy. Inact
ion could cause problems.
 524Ø DATA "",TEMP-,ERANCE,"",A need for econ
omy and good management.,Connections with the
 Church or religion look well.
 525Ø DATA "",THE,DEVIL,"",The effect of extr
a effort - possibly   violence.,Blindness to
what is right. Pettiness.
 526Ø DATA "",THE,TOWER,"",Adversity and dist
ress. Misery caused byconflict.,Problems caus
ed by argument or             disagreement.
 527Ø DATA "",THE,STAR,"",Loss by theft of ma
```

```
terial or spiritual things.,Loss by theft of
material or spiritual  things.
 528Ø DATA "",THE,MOON,"",A warning of hidden
 enemies who will    deceive.,Instability and
 inconstancy of those    about you.
 529Ø DATA "",THE,SUN,"",Material happiness.
Contentment through good marriage.,Contentmen
t on material and spiritual   planes.
 53ØØ DATA "",JUDGEMENT,"","",Change of posit
ion. Renewal and rebirth.,Weakness through an
 over-simplification of events.
 531Ø DATA "",THE,WORLD,"",Assured success vi
a a voyage or change  of place.,Inertia or fi
xity may cause stagnation.
 532Ø DATA "",THE,FOOL,"",A time to consider
your career -         consolidation.,Folly thr
ough extravagence or pleasure  -seeking.
 533Ø DATA THE,ACE,OF,PENTACLES,Perfect conte
ntment - also financial    reward.,Bad intell
igence. The evil side of       wealth.
 534Ø DATA THE,TWO,OF,PENTACLES,A time for ga
iety and recreation.        Perhaps good news.
,A facade of enjoyment to keep others    happ
y.
 535Ø DATA THE,THREE,OF,PENTACLES,Craftsmansh
ip and the need to use        practical gifts.
,Mediocrity in work. A tendency to        pett
iness.
 536Ø DATA THE,FOUR,OF,PENTACLES,Gift or lega
cy. The aquisition of new   possessions.,Dela
y to some project through others'   oppositio
n.
 537Ø DATA THE,FIVE,OF,PENTACLES,"Material di
fficulties, but love is well looked on.",Diso
rder or chaos. It may spell ruin forsome proj
ect.
 538Ø DATA THE,SIX,OF,PENTACLES,Gratification
 through gifts or other    pleasures.,"Envy a
nd jealousy, which lead to         illusory de
sires."
 539Ø DATA THE,SEVEN,OF,PENTACLES,Ingenuity i
n business and money matters.,Quarrels over m
```

```
oney or possessions.
 5400 DATA THE,EIGHT,OF,PENTACLES,A creative
period in craft or business.,A tendency to va
nity. This may stand     against ambition.
 5410 DATA THE,NINE,OF,PENTACLES,Prudence. Sa
fety and accomplishment fromthis.,Deception.
Beware ofothers keeeping bad faith.
 5420 DATA THE,TEN,OF,PENTACLES,Gain in monet
ary or family matters.,Monetary loss through
gambling or         robbery. Take no chances.
 5430 DATA THE,PAGE,OF,PENTACLES,A time for s
cholarship or reflection.   Look to your affa
irs.,Unfavourable news causing worry.
 5440 DATA THE,KNIGHT,OF,PENTACLES,The effect
 of someone with your affairs at heart.,Idlen
ess and lack of energy. Stagnation.
 5450 DATA THE,QUEEN,OF,PENTACLES,"The effect
s of an opulent, though         generous woman.
",Suspicion and mistrust may affect your  rea
soning.
 5460 DATA THE,KING,OF,PENTACLES,"Intellect,
particularly mathematics.     Look to business
.",The temptation of vice. Watch for         we
akness.
 5470 DATA THE,ACE,OF,SWORDS,Great force in l
ove or hate. Triumph     through this.,Great f
orce in love or hate. Not to the good.
 5480 DATA THE,TWO,OF,SWORDS,"Conformity, and
 the stability this may  bring.",Lies and dis
loyalty. Beware of untruths.
 5490 DATA THE,THREE,OF,SWORDS,A feeling of l
oss of someone or           something.,Confusio
n and a feeling of intellectual isolation.
 5500 DATA THE,FOUR,OF,SWORDS,The need for so
litude. A time for         contemplation.,Preca
utions. Be careful what you do.
 5510 DATA THE,FIVE,OF,SWORDS,Loss or destruc
tion of something close  to you.,Loss or dest
ruction of something close  to you.
 5520 DATA THE,SIX,OF,SWORDS,"A journey, whic
h may include travel by  or over water.","A p
roposal of love, or other              declarat
```

```
ion. Publicity."
 553Ø DATA THE,SEVEN,OF,SWORDS,Annoyance at t
he failure of a plan       through disagreemen
t.,Good advice from a wise person. Take    he
ed of it.
 554Ø DATA THE,EIGHT,OF,SWORDS,Bad news of an
 illness or other crisis.,Difficulties and op
position. Treachery.
 555Ø DATA THE,NINE,OF,SWORDS,"Deep disappoin
tment, possibly through   death.",Suspicion c
ast on you or a feeling of    shame.
 556Ø DATA THE,TEN,OF,SWORDS,Pain and sadness
. All is not well.,Transitory profit or advan
tage. Make    hay....
 557Ø DATA THE,PAGE,OF,SWORDS,Examination. Th
e ability to study       details normally mis
sed.,The use of observation to bad ends.
Spying.
 558Ø DATA THE,KNIGHT,OF,SWORDS,Skill and bra
very. The time to use such virtues.,Foolishne
ss. The inability to think      soundly.
 559Ø DATA THE,QUEEN,OF,SWORDS,Sadness or emb
arrassment for you (if    female) or a woman
close to you.,A closed mind. Watch against bi
gotry or deceit.
 56ØØ DATA THE,KING,OF,SWORDS,The effects of
one with authority to    make decisions.,A te
ndency to cruelty or perversity.    Watch thi
s.
 561Ø DATA THE,ACE,OF,CUPS,Contentment with a
ll relationships.,Discontentment. The desire
to put right your problems.
 562Ø DATA THE,TWO,OF,CUPS,Love and passion.
The inter-relation of the sexes.,Love and pas
sion. The inter-relation of the sexes.
 563Ø DATA THE,THREE,OF,CUPS,A happy conclusi
on. Merriment and       filfilment.,A new ven
ture or holiday. Achievement   from this.
 564Ø DATA THE,FOUR,OF,CUPS,Tiredness and ima
ginary worries caused  by a strict routine.,N
ew relationships and a delight in      novelt
y.
```

```
 5650 DATA THE,FIVE,OF,CUPS,Receipt of a gift
. This results in       another loss,The chanc
e of new alliences. Business   looks well.
 5660 DATA THE,SIX,OF,CUPS,Reflections on the
 past and childhood.,The chance of fresh exci
tements.          Renewal.
 5670 DATA THE,SEVEN,OF,CUPS,"Imagination. A
good time for creative   work, though transit
ory.",A desire to succeed. Force of will.
 5680 DATA THE,EIGHT,OF,CUPS,Dejection over t
hings as they are. This will pass.,Great joy
and happiness. A celebration.
 5690 DATA THE,NINE,OF,CUPS,A victory in some
thing you desire.,Loyalty of others to you. H
onesty and   good advice.
 5700 DATA THE,TEN,OF,CUPS,Contentment with y
our present            surroundings and things
 as they are.,Indignation at an affront in lo
ve. A    violent response.
 5710 DATA THE,PAGE,OF,CUPS,The effects of a
fair and studious youngman. He may help you,A
 taste for risk could lead to          decept
ion.
 5720 DATA THE,KNIGHT,OF,CUPS,A proposition o
r invitation.,A trick or fraud. Be on your gu
ard.
 5730 DATA THE,QUEEN,OF,CUPS,"The effects of
a fair woman. She is      dreamy, but also vis
ionary.",The influence of a distinguished wom
an. Don't trust her.
 5740 DATA THE,KING,OF,CUPS,"The effects of a
 creative man in        business, Law or the
Church.",The effect of a rogue or scandal-mon
ger.He is dishonest.
 5750 DATA THE,ACE,OF,WANDS,The beginning of
new things - creation. Possibly a birth.,A fa
ll in fortune. Ruin of something   establish
ed.
 5760 DATA THE,TWO,OF,WANDS,Physical sufferin
g or sadness.          Dissatisfaction with p
ossessions.,Surprise and wonder leading to di
squiet.
```

```
 5770 DATA THE,THREE,OF,WANDS,Strength in tra
de and dealings with      people.,The end of t
roubles. Adversity is over.
 5780 DATA THE,FOUR,OF,WANDS,A happy life. Pe
ace and prosperity.,Prosperity and increase.
An appreciationof beauty.
 5790 DATA THE,FIVE,OF,WANDS,Imitation or sha
m. A superficial view.,Contradiction and the
tendency of othersto trick you.
 5800 DATA THE,SIX,OF,WANDS,Success in undert
akings or great news.,Apprehension and the fe
ar of defeat.
 5810 DATA THE,SEVEN,OF,WANDS,A need to work
by discussion and         negotiation. Final s
uccess.,Perplexity. Embarrassments and worrie
s.
 5820 DATA THE,EIGHT,OF,WANDS,Haste towards a
 favourite outcome.       Change.,Quarrels ari
sing from jealousy or envy.
 5830 DATA THE,NINE,OF,WANDS,Delay caused by
opposition to your        ideas.,Obstacles to y
our ideas. Possible       calamity.
 5840 DATA THE,TEN,OF,WANDS,Financial gain bu
t opposition in law.,Difficulties and the int
rigue of others against you.
 5850 DATA THE,PAGE,OF,WANDS,The influence of
 a dark young man. He     may bring news.,Bad n
ews or announcements. Indecision.
 5860 DATA THE,KNIGHT,OF,WANDS,Departure or a
bsence. A move or even    emigration.,Discord
. Interruption of the present    status quo.
 5870 DATA THE,QUEEN,OF,WANDS,The influence o
f a country woman who is loving and honourabl
e.,Problems of jealousy - deceipt or        in
fidelity.
 5880 DATA THE,KING,OF,WANDS,The influence of
 a friendly countryman  who is honest and con
scientious.,"The influence of a good but seve
re man. He is austere, yet tolerant."
```

BALLOON

BRIAN JONES
Original program by A. Roe

GENERAL DESCRIPTION

In this exciting game you must steer a balloon through a rocky ravine to safety. You control your balloon with the keys shown in the introduction sequence and you must avoid having the rocks burst your balloon. Easy, except that your balloon sways from side to side in the wind and can easily swing back at just the wrong moment. It takes a lot of patience and skill to win at this game! Bon voyage, and I hope it doesn't take you eighty days to get through this one.

DETAILED DESCRIPTION

Lines 10-60 data listings.
70-90 rems.
100-240 order of running procedures.
250-370 game over routine.
380-520 PROC MOVE - moves balloon through maze.
530-580 error message.
590-730 PROC CHARS - defines characters used.
740-830 PROC BALLOON - checks position of balloon to see if it has burst, otherwise prints new position of balloon.
840-960 PROC COURSE - draws course on screen.
970-1030 PROC LIFE - procedure dealing with bonuses and number of lives left.
1040-1140 PROC STRINGS - joins characters to form graphics of balloons used in opening titles, also defines sound during this display.
1150-1200 PROC INIT - sets initial values of variables, and position of balloon.
1210-1260 PROC BURST - deals with balloon hitting side of

maze and bursting.
1270-1290 PROC DELAY - initiates short delay in running.
1300-1530 PROC TITLE - displays opening titles.
1540-1620 PROC CIRCLE - draws circle for opening titles.
1630-1670 PROC SCORE - works out score and bonuses.
1680-1780 PROC GOAL - procedure used once balloon has been successfully manoeuvred through maze.
1790-1820 PROC GETSPACE - gets space to continue with program.
1830-1860 PROC MUSIC - runs music.

```
   9Ø REM BALLOON - BBC VERSION
  1ØØ DATAP,5,C,21,W,33,.,53
  11Ø DATA27,33,1Ø,53,1Ø,53,4,61,4,69,1Ø,53,1
Ø,81,2,81,2,81,2,81,2,81,2,81,5,73,5,69,15,69
,5,73,5,81,5,69,5,73,6,61,2,61,2,61,2,61,2,61
,2,61,5,53,5,53,18
  12Ø DATA1,8,32,88,2,17,16,4,3,88,2,19,13,6,
2,88,2,22,7,1Ø,1,88,2,24,1,14,1,88,1,39,1,88,
1,39,1,88,1,39,1,88,2,Ø,8,31,1,88,3,Ø,7,22,1,
9,1,88,4,Ø,3,8,4,7,8,9,1,88,4,Ø,2,8,6,6,9,8,1
,88,4,Ø,1,8,8,6,9,7,1,88,4,Ø,2,6,1Ø,6,1Ø,5,1,
88
  13Ø DATA4,Ø,2,5,12,8,6,6,1,88,3,Ø,1,5,14,19
,1,88,2,Ø,1,38,1,88,2,Ø,1,38,1,88,2,Ø,1,38,1,
88,2,Ø,1,38,1,88,3,Ø,1,22,2,14,1,88,3,Ø,1,19,
9,7,4,88,2,Ø,1,5,34,88,2,Ø,2,6,32,88
  14Ø DATA2,Ø,2,28,1Ø,88,2,Ø,2,38,88,2,Ø,2,38
,88,2,Ø,2,38,88,2,Ø,2,38,88,2,Ø,5,35,88,2,Ø,2
Ø,2Ø,88,1,Ø,4Ø
  15Ø DATA14,53,2Ø,73,6,73,4,81,6,1Ø1,1Ø,85,4
,89,1Ø,89,1Ø,89,1Ø,89,1Ø,93,6,81,13,73,6,81,1
Ø
  16Ø REM*******************************
  17Ø REM****BALLOON******************
  23Ø REM*******************************
  24Ø ON ERROR GOTO67Ø
  25Ø MODE2
  26Ø VDU23,1,Ø;Ø;Ø;Ø;
  27Ø PROCstrings
  28Ø PROCtitle
```

```
  290 MODE1
  300 VDU 5
  310 PROCinit
  320 PROCchars
  330 PROCcourse
  340 PROClife
  350 REPEAT
  360   PROCmove:PROCscore
  370   UNTIL LIFE%=4ORGOALF%=1
  380 IFLIFE%<4 RESTORE120:GOALF%=0:GCOL0,3:C
LS:PROCinit:GOTO330
  390 PRINTTAB(15,18)"GAME OVER":TIME=0:REPEA
T UNTIL TIME>=300
  400 MODE2
  410 COLOUR3
  420 PRINTTAB(0,3)"YOU SCORED ";SCORE%
  430 PRINTTAB(0,6)"ANOTHER GAME ?"
  440 *FX15,0
  450 RESTORE110:PROCmusic:PROCdelay
  460 IFGET$="Y" RESTORE120:MODE1:VDU5:PROCin
it:SCORE%=0:LEVEL%=1:DIFF%=45:LIFE%=1:GOALF%=
0:FREEF%=1:GOTO330
  470 PROCdelay
  480 *FX11,50
  490 *FX12,7
  500 *FX15,0
  510 VDU4:MODE1:END
  520 DEFPROCmove
  530 GCOL0,0
  540 PN%=1
  550 IFINKEY(-72)ANDINKEY(-25) SOUND1,-10,20
0,1:PROCballoon:XPOS%=XPOS%-32:YPOS%=YPOS%+32
:GOTO640
  560 IFINKEY(-72)ANDINKEY(-73) SOUND1,-10,20
0,1:PROCballoon:XPOS%=XPOS%-32:YPOS%=YPOS%-32
:GOTO640
  570 IFINKEY(-57)ANDINKEY(-73) SOUND1,-10,20
0,1:PROCballoon:XPOS%=XPOS%+32:YPOS%=YPOS%-32
:GOTO640
  580 IFINKEY(-57)ANDINKEY(-25) SOUND1,-10,20
0,1:PROCballoon:XPOS%=XPOS%+32:YPOS%=YPOS%+32
:GOTO640
```

```
  590 IFINKEY(-57)ANDXPOS%>=960ANDYPOS%<=224
GOTO640
  600 IFINKEY(-25) SOUND1,-10,200,1:PROCballo
on:YPOS%=YPOS%+32:GOTO640
  610 IFINKEY(-73) SOUND1,-10,200,1:PROCballo
on:YPOS%=YPOS%-32:GOTO640
  620 IFINKEY(-57) SOUND1,-10,200,1:PROCballo
on:XPOS%=XPOS%+32:GOTO640
  630 IFINKEY(-72) SOUND1,-10,200,1:PROCballo
on:XPOS%=XPOS%-32
  640 GCOL0,2
  650 PROCballoon:PN%=0:IFTIME>=DIFF% GCOL0,0
:PROCballoon:GCOL0,2:SW%=SW%+1:NSWINGS%=NSWIN
GS%+1:TIME=0:IFSW%=5 SW%=1
  660 ENDPROC
  670 MODE1:REPORT:PRINT " IN LINE ";ERL
  680 VDU4
  690 *FX11,50
  700 *FX12,7
  710 *FX15,0
  720 END
  730 DEFPROCchars
  740 VDU23,225,7,24,32,64,64,128,128,128
  750 VDU23,226,224,24,4,2,2,1,1,1
  760 VDU23,227,128,128,128,64,64,32,24,7
  770 VDU23,228,1,1,1,2,2,4,24,224
  780 VDU23,229,7,7,7,7,0,0,0,0
  790 VDU23,230,224,224,224,224,0,0,0,0
  800 VDU23,231,1,1,1,1,1,1,1,1
  810 VDU23,232,0,0,0,0,0,0,0,0
  820 VDU23,240,128,64,32,16,8,4,2,1
  830 VDU23,241,1,2,4,8,16,32,64,128
  840 VDU23,242,255,255,255,255,255,255,255,2
55
  850 VDU23,245,12,62,126,127,252,248,224,64
  860 VDU23,246,2,4,8,16,32,112,112,0
  870 ENDPROC
  880 DEFPROCballoon
  890 IFPN%=0 GOTO931
  900 IF(SW%=1ORSW%=3)AND(POINT(XPOS%+16,YPOS
%-16)=3ORPOINT(XPOS%+48,YPOS%-16)=3ORPOINT(XP
OS%+16,YPOS%-48)=3ORPOINT(XPOS%+48,YPOS%-48)=
```

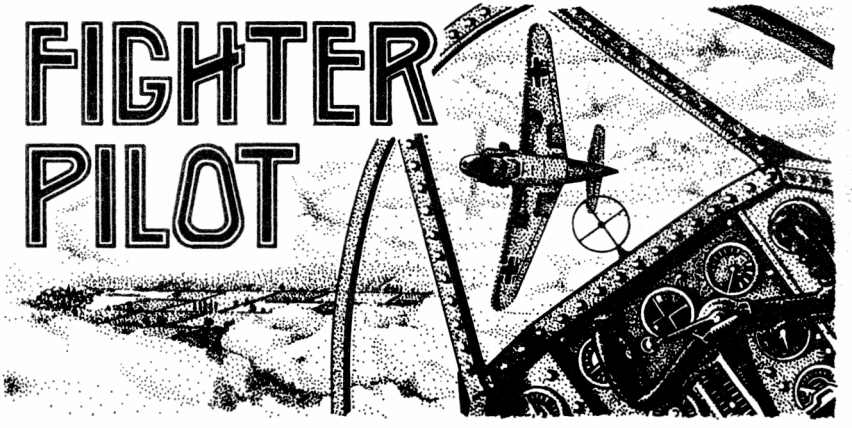
```
3) PROCburst:ENDPROC
  910  IFSW%=2AND(POINT(XPOS%+112,YPOS%-16)=3
ORPOINT(XPOS%+80,YPOS%-16)=3ORPOINT(XPOS%+80,
YPOS%-48)=3ORPOINT(XPOS%+80,YPOS%-16)=3ORPOIN
T(XPOS%+80,YPOS%-80)=3ORPOINT(XPOS%+56,YPOS%-
116)=3) PROCburst:ENDPROC
  920 IFSW%=4AND(POINT(XPOS%-56,YPOS%-16)=3OR
POINT(XPOS%-16,YPOS%-16)=3ORPOINT(XPOS%-56,YP
OS%-48)=3ORPOINT(XPOS%-16,YPOS%-80)=3ORPOINT(
XPOS%+16,YPOS%-116)=3) PROCburst:ENDPROC
  930 IFPOINT(XPOS%+48,YPOS%-148)=3ORPOINT(XP
OS%+16,YPOS%-148)=3 PROCburst:ENDPROC
  931 VDU 5
  940 MOVEXPOS%,YPOS%:PRINTBALLOON$(SW%)
  941 VDU 4
  950 ENDPROC
  960 DEFPROCcourse
  961 MOVE 0,1023
  970 FORI%=0TO31
  980   READ BLOCKS%:IF BLOCKS%=88 GOTO980
 1000   FORJ%=1TOBLOCKS%
 1010     READSPACES%:IFSPACES%=0 GOTO1030
 1020     PRINTSPC(SPACES%);
 1030     READCROSS%:IFCROSS%=88 GOTO1050
 1040     PRINTSTRING$(CROSS%,"*");
 1050     NEXT:NEXT
 1060 IFLIFE%<3 COLOUR 1:VDU 4:FORI%=1TO4-LIF
E%:PRINTTAB(34+I%,26);MINI$;:NEXT
 1061 VDU 4
 1070 ENDPROC
 1080 DEFPROClife
 1090 GCOL0,1:MOVE16,1000:DRAW16,1000:DRAW232
,1000:MOVE232,768:DRAW16,768:DRAW16,1000
 1100 MOVE1000,32:DRAW1000,32:DRAW1264,32:DRA
W1264,224:DRAW1000,224
 1110 COLOUR 2:PRINTTAB(1,2)"BONUS":PRINTTAB(
1,4)STRING$(4,CHR$242):PRINTTAB(1,4);BONUS%
 1120 COLOUR 1:PRINTTAB(34,29)"START"
 1130 COLOUR 0:PRINTTAB(34+LIFE%,26);MINI$:CO
LOUR 2
 1140 ENDPROC
 1150 DEFPROCstrings
```

```
 1160 DIMBALLOON$(4)
 1170 BALLOON$(1)=CHR$225+CHR$226+CHR$8+CHR$8
+CHR$10+CHR$227+CHR$228+CHR$8+CHR$8+CHR$10+CH
R$231+CHR$8+CHR$10+CHR$231+CHR$8+CHR$10+CHR$2
29+CHR$230
 1180 BALLOON$(2)=CHR$9+CHR$9+CHR$225+CHR$226
+CHR$8+CHR$8+CHR$10+CHR$227+CHR$228+CHR$8+CHR
$8+CHR$10+CHR$241+CHR$8+CHR$8+CHR$10+CHR$241+
CHR$8+CHR$8+CHR$10+CHR$229+CHR$230
 1190 BALLOON$(3)=CHR$225+CHR$226+CHR$8+CHR$8
+CHR$10+CHR$227+CHR$228+CHR$8+CHR$8+CHR$10+CH
R$231+CHR$8+CHR$10+CHR$231+CHR$8+CHR$10+CHR$2
29+CHR$230
 1200 BALLOON$(4)=CHR$8+CHR$8+CHR$225+CHR$226
+CHR$8+CHR$8+CHR$10+CHR$227+CHR$228+CHR$8+CHR
$10+CHR$240+CHR$10+CHR$240+CHR$8+CHR$10+CHR$2
29+CHR$230
 1210 MINI$=CHR$245+CHR$8+CHR$8+CHR$10+CHR$24
6
 1220 ENVELOPE3,1,0,0,0,0,0,0,127,-1,-1,-1,12
0,60
 1230 ENVELOPE1,1,4,-10,4,5,20,5,127,0,0,-5,1
26,126
 1240 ENVELOPE2,7,10,-10,50,5,20,5,127,0,0,-1
,126,126
 1250 ENDPROC
 1260 DEFPROCinit
 1270 XPOS%=960:YPOS%=192:SW%=1:TIME=0:PN%=0:
NSWINGS%=0:HIGHT%=192:A=30:BONUS%=1000
 1280 *FX15,0
 1290 *FX11,1
 1300 *FX12,1
 1310 ENDPROC
 1320 DEFPROCburst
 1330 SOUND0,-15,4,2:VDU19,0,7,0,0,0:VDU19,3,
0,0,0,0:PROCdelay:VDU20
 1340 LIFE%=LIFE%+1
 1350 COLOUR 0:PRINTTAB(35,26);MINI$:COLOUR 2
 1360 PROCinit:IFLIFE%=4 ENDPROC ELSE PROClif
e
 1370 ENDPROC
 1380 DEFPROCdelay
```

```
 1390 TIME=0:REPEAT UNTIL TIME>=40
 1400 ENDPROC
 1410 DEFPROCtitle
 1420 FORI=1TO4:READ NAME$,PIT%:COLOUR1:PRINT
TAB(I,1);NAME$:SOUND1,2,PIT%,2:PROCdelay:NEXT
:COLOUR3:PRINTTAB(6,1)"PRESENTS..."
 1430 VDU19,6,0,0,0,0:VDU19,5,0,0,0,0:VDU19,7
,0,0,0,0:VDU19,11,0,0,0,0:VDU19,9,0,0,0,0:VDU
19,139,0,0,0,0
 1440 GCOL0,11:COLOUR9:COLOUR139
 1450 PROCcircle(610,640,300)
 1460 PRINTTAB(6,10)"BALLOON":PRINTTAB(8,12)"
FUN"
 1470 GCOL0,6:MOVE800,406:DRAW640,128:MOVE416
,406:DRAW560,128
 1480 GCOL0,5:PROCcircle(600,186,20):GCOL0,7:
PROCcircle(600,140,25)
 1490 GCOL0,6:MOVE560,32:MOVE560,128:PLOT85,6
40,32
 1500 MOVE560,128:PLOT85,640,128
 1510 VDU20
 1520 PROCmusic
 1530 PRINTTAB(2,25)"HIT"TAB(1,26)"SPACE"TAB(
14,25)"FOR"TAB(13,26)"MORE!"
 1540 PROCgetspace
 1550 COLOUR128:PRINTTAB(1,1)SPC16:COLOUR139:
PRINTTAB(6,10)SPC7TAB(8,12)SPC3
 1560 VDU19,139,4,0,0,0
 1570 FORI=7TO4STEP-3:COLOURI:PRINTTAB(7,5)"G
UIDE"TAB(6,8)"BALLOON"TAB(6,11)"THROUGH"TAB(7
,14)"MAZES"TAB(7,17)"USING:":IFI=7 PROCgetspa
ce
 1580    NEXT
 1590 FORI=7TO4STEP-3:COLOURI:PRINTTAB(9,7)"N
"TAB(9,9)"^"TAB(5,11)"W @    [ E"TAB(9,13)":"T
AB(9,15)"S":IFI=7 PROCgetspace
 1600    NEXT
 1610 FORI=7TO4STEP-3:COLOURI:PRINTTAB(7,5)"B
ONUS"TAB(6,8)"BALLOON"TAB(8,11)"FOR"TAB(7,14)
"10,000"TAB(7,17)"POINTS":IFI=7 PROCgetspace
 1620    NEXT
 1630 COLOUR12:PRINTTAB(7,11)"GOOD"TAB(6,13)"
```

```
FLYING.."
 1640 PROCgetspace:CLG:VDU20:DIFF%=45:LEVEL%=
1:SCORE%=0:FREEF%=1:GOALF%=0:LIFE%=1:ENDPROC
 1650 REM**********************
 1660 DEFPROCcircle(X%,Y%,R%)
 1670 LOCALA%
 1680 FORA%=0TO360STEP20
 1690     MOVEX%,Y%
 1700    MOVESIN(RAD(A%))*R%+X%,COS(RAD(A%))*R
%+Y%
 1710    PLOT85,SIN(RAD(A%+20))*R%+X%,COS(RAD(
A%+20))*R%+Y%
 1720    NEXT
 1730 ENDPROC
 1740 DEFPROCscore
 1750 IFYPOS%>HIGHT% HIGHT%=YPOS%:SCORE%=SCOR
E%+(50*LEVEL%):IFFREEF%=1ANDSCORE%>=10000 LIF
E%=LIFE%-1:SOUND0,1,0,75:PRINTTAB(34+LIFE%,26
);MINI$:FREEF%=0
 1760 IFNSWINGS%>A COLOUR 2:BONUS%=BONUS%-20:
PRINTTAB(1,4);BONUS%:A=10:NSWINGS%=0
 1770 IFYPOS%>900ANDXPOS%<232 PROCgoal
 1780 ENDPROC
 1790  DEFPROCgoal:VDU 5
 1800 GCOL0,0:MOVEXPOS%,YPOS%:PRINTBALLOON$(S
W%):COLOUR 2:VDU 4:PRINTTAB(2,2);BALLOON$(1)
 1810 VDU19,2,13,0,0,0:VDU19,3,10,0,0,0:VDU19
,0,4,0,0,0
 1820 PROCmusic
 1830 LEVEL%=LEVEL%+1
 1840 SCORE%=SCORE%+BONUS%
 1850 GOALF%=1
 1860 DIFF%=DIFF%-10
 1870 TIME=0:REPEAT UNTIL TIME>=200
 1880 VDU 20
 1890 ENDPROC
 1900 DEFPROCgetspace:*FX15,1
 1910 REPEAT UNTIL GET$=" "
 1920 SOUND1,2,20,2
 1930 ENDPROC
 1940 DEFPROCmusic
 1950 READNUM%
```

```
 1960 FORJ=1TONUM%:READPIT%,DUR%:FORT=1TO3:SO
UNDT,3,PIT%,DUR%:NEXTT,J
 1970    ENDPROC
```



HARCHARAM WASSI
Original program by E.G. Kemplen

GENERAL DESCRIPTION

You are seated in the cockpit of your Spitfire, hot on the tail of an enemy aircraft. He, not unnaturally, is doing his level best to get out of your gun sights. The object of the game is to score 50 points by hitting the target as often as you can, before your fuel runs out.

You can control your aircraft so as to climb, dive, swoop to the left or right; the screen display always shows the position of the enemy in your line of fire. Displays of your current score, number of shots and fuel level are also provided. A good eye, and considerable dexterity are needed to play this game well; eleven fingers are also an advantage!

DETAILED DESCRIPTION

Lines 10-70 initialise sound and screen.
70-100 give instructions and start game.
110-160 set up display for gun.
170-210 set up enemy plane and put it on the screen.
220-240 update score and status display on screen.
250-320 firing routine check for hits.

330-380 move enemy plane about then...
390 go back round game loop.
400-510 routine to convert keyboard commands to aim direction.
520-740 PROC HIT - makes explosion with sound effects and adds to score when you hit a plane.
750-840 routine for end of game. Display statistics from game and allow another game if desired.
850-1250 routine giving comprehensive instructions for playing the game.
1260 PROC PRINT - places strings on screen with centring.
1270 re-enable cursor at end of game.

```
   10 REM***FIGHTER PILOT****
   20 REM***SET UP SCREEN****
   30 ON ERROR : MODE 7 : END
   40 *TV255
   50 MODE7
   60 VDU23,1,0;0;0;0;:REM*CURSOR OFF*
   70 ENVELOPE 1,2,7,-1,0,15,45,0,5,-1,0,-127
,127,50
   80 PROCinstruct
   90 CLS
  100 T=TIME:SC=0:S=0:AIM=0:II=0
  110 ST%=0 : SF%=175
  120 REM***SET UP GUNSIGHT****
  130  FOR I=1 TO 25:?(31764+40*I)=43:NEXT
  140  FOR I=1 TO 40:?(32263+I)=43:?(31744+I)
=32:NEXT
  150 C=32284
  160  ?C=79
  170 A=31844+(RND(5)*40):REM*POSITION OF ENE
MY*
  180 B1=?(A-2):B2=?(A-1):B3=?(A):B4=?(A+1):B
5=?(A+2):B6=?(A-40)
  190 REM***PUT IN ENEMY PLANE****
  200 ?(A-2)=96:?(A-1)=96:?(A)=48:?(A+1)=96:?
(A+2)=96:?(A-40)=33
  210 REM***SCORE LINE****
  220 FOR N=1 TO 10:NEXT
  230 PRINT CHR$30;CHR$130;"FUEL ";30-INT((TI
ME-T)/100)/10;TAB(11)"GALS";" ";TAB(17);CHR$1
```

```
34;"SCORE ";SC;TAB(30);CHR$133;"SHOTS ";S
  235 *FX 15,1
  240 REM***CHECK AIM****
  250 X$=INKEY$(2):IF X$<>"" THEN 410
  260 REM***CHECK FOR FIRE****
  270 F%=INKEY(-1):IF F%=-1 THEN B7=?(C):?C=4
2:FOR I=1 TO 100:NEXT:?C=B7:S=S+1:SOUND &10,1
,40,10 ELSE IF ST%=0 SOUND &11,-3,SF%,-1 : SO
UND &10,-11,7,-1
  280 IF ST%<>0 ST%=ST%-1
  290 F%=INKEY(-1):IF F%=-1 AND ?(C)<>79 THEN
 PROChit
  300 IF S>249 THEN PRINT "OUT OF AMMUNITION
RUN LIKE HELL FOR HOME":GOTO 800
  310 IF (TIME-T)/100>300 THEN PRINT "YOU'RE
OUT OF FUEL, BREAK OFF":GOTO 800
  320 REM***ENEMY AVOIDING ACTION****
  330 N=-41+(RND(3)-1)+(RND(3)-1)*40
  340 REM***CALCULATE NEW ENEMY POSITION***
  350 ?(A-2)=B1:?(A-1)=B2:?(A)=B3:?(A+1)=B4:?
(A+2)=B5:?(A-40)=B6
  360 A=A+N+AIM
  370 IF A<31827 THEN A=A+80
  380 IF A>32700 THEN A=A-80
  390 GOTO 180
  400 REM***AIM CORRECTION****
  410 IF X$="B" AIM=-39 : PROC_sd
  420 IF X$="N" AIM=-40 : PROC_sd
  430 IF X$="M" AIM=-41 : PROC_sd
  440 IF X$="G" AIM=1
  450 IF X$="H" THEN AIM=0
  460 IF X$="J" THEN AIM=-1
  470 IF X$="T" AIM=41 : PROC_su
  480 IF X$="Y" AIM=40 : PROC_su
  490 IF X$="U" AIM=39 : PROC_su
  500 IF ST%=0 SOUND &11,0,SF%,-1 : SOUND &10
,-9,7,-1
  510 GOTO 270
  520 DEF PROC_su
  530 IF ST%<>0 ST%=ST%-1 : ENDPROC
  540 IF SF%<255 SF%=SF%+1
  550 ENDPROC
```

```
  560
  570 DEF PROC_sd
  580 IF ST%<>0 ST%=ST%-1 : ENDPROC
  590 IF SF%>93 SF%=SF%-1
  600 ENDPROC
  610
  620 REM***HIT SUBROUTINE****
  630 DEFPROChit
  640 IF ?(C)=96 OR ?(C)=33 THEN II=2 : SOUND
 &10,-15,4,20 : ST%=8
  650 IF ?(C)=48 THEN II=5 : SOUND 2,1,120,24
  660 SC=SC+II
  670 FOR I=1 TO II
  680   ?(C-39)=42:?(C-41)=42:?(C+41)=42:?(C+
39)=42
  690   IF SC>49 THEN 760
  700   FOR J=1 TO 25:NEXT
  710   ?(C-39)=32:?(C-41)=32:?(C+41)=32:?(C+
39)=32
  720   FOR J=1 TO 10:NEXT
  730   NEXT
  740 ENDPROC
  750 REM***FINAL DISPLAY****
  760 T1=INT((TIME-T)/100)
  770 PRINT TAB(17)"SCORE ";SC
  780 PRINT "ENEMY PLANE DESTROYED IN ";T1;"
SECONDS"
  790 PRINT "USING ";S;" ROUNDS OF AMMUNITION
"
  800 PRINT "TO PLAY AGAIN PRESS SPACE "
  810 PRINT "TO STOP PRESS 0 "
  820 IF GET$="0" THEN 1270
  830 IF GET$=" " THEN 90
  840 GOTO 820
  850 REM***INSTRUCTIONS****
  860 DEFPROCinstruct
  870 CLS:PRINT '';TAB(4);CHR$(146)"j7'!k7h7'
k4j5 j5'k7'j7''j7'k4"
  880 PRINT TAB(4);CHR$(146)"j=, j5j5 p0j=,n5
 j5 j=, j=1.!"
  890 PRINT TAB(4);CHR$(146)"j5  zu*upz%j5 j5
 j5 juppj5 )0"
```

```
  900 PRINT:PRINT
  910 PRINT TAB(9);CHR$(146)"j7'k4k7j5  h7'k4
'k7'"
  920 PRINT TAB(9);CHR$(146)"j=,.!j5j5  j5 j5
 j5 "
  930 PRINT TAB(9);CHR$(146)"j5   zujupp*upz%
 j5 "
  940
  950 PRINT:PRINT
  960 XX$="YOU ARE THE PILOT OF A SPITFIRE":P
ROCprint
  970 XX$="AHEAD OF YOU IS AN M.E.109":PROCpr
int
  980 XX$="HE IS TAKING AVOIDING ACTION, AND
YOU":PROCprint
  990 XX$="MUST TRY TO SHOOT HIM DOWN":PROCpr
int
 1000 XX$="YOUR GUNS ARE FIRED BY PRESSING TH
E":PROCprint
 1010  XX$=CHR$(133)+"SHIFT"+CHR$(135)+" KEY"
:PROCprint
 1020 PRINT:XX$="YOU CAN CHANGE YOUR AIM USIN
G THE":PROCprint
 1030 XX$=CHR$(131)+"T Y U G H J B N M"+CHR$(
135)+"KEYS AS FOLLOWS ":PROCprint
 1040 PRINT:PRINT:XX$=" PRESS"+CHR$(134)+"SPA
CE"+CHR$(135)+"TO CONTINUE":PROCprint
 1050 IF GET$=" " THEN 1070
 1060 GOTO 1050
 1070 CLS:PRINT'':XX$="THE"+CHR$+(131)+"Y"+CH
R$(135)+"KEY MAKES YOUR PLANE CLIMB ":PROCpri
nt
 1080 XX$="THE"+CHR$+(131)+"N"+CHR$(135)+"KEY
 MAKES YOUR PLANE DIVE ":PROCprint
 1090 XX$="THE"+CHR$+(131)+"G"+CHR$(135)+"KEY
 TURNS YOU TO THE LEFT":PROCprint
 1100 XX$="THE"+CHR$+(131)+"J"+CHR$(135)+"KEY
 TO THE RIGHT":PROCprint
 1110 XX$="THE"+CHR$(131)+"T U B M"+CHR$(135)
+"KEYS MAKE YOU TURN AS":PROCprint
 1120 XX$="WELL AS CLIMBING OR DIVING":PROCpr
int
```

```
 1130 XX$="THE"+CHR$(133)+"H"+CHR$(135)+"KEY
CENTRALISES YOUR CONTROLS":PROCprint
 1140 PRINT:XX$="T Y U  ":PROCprint
 1150 PRINT:XX$="G H J ":PROCprint
 1160 PRINT:XX$=" B N M":PROCprint
 1170 PRINT:XX$="YOU CANNOT ALTER COURSE WHIL
ST FIRING":PROCprint
 1180 XX$="YOUR GUNS":PROCprint
 1190 PRINT:XX$="THE OBJECT OF THE GAME IS TO
 SCORE":PROCprint
 1200 XX$="50 POINTS":PROCprint
 1210 PRINT:XX$="A HIT ON THE WING OR TAIL SC
ORES 2":PROCprint
 1220 XX$="A FUSELAGE HIT SCORES 5":PROCprint
 1230 PRINT:XX$=" PRESS"+CHR$(134)+"SPACE"+CH
R$(135)+"TO START THE GAME":PROCprint
 1240 IF GET$=" " THEN ENDPROC
 1250 GOTO 1240
 1260 DEFPROCprint:PRINT TAB((40-LEN(XX$))/2)
;XX$:ENDPROC
 1270 VDU23,1,1;0;0;0;
```

BOOK INDEX

PAUL BORRETT
Original program by Ian Andrews

GENERAL DESCRIPTION

This is a utility, in colour, for indexing information. Material is stored in the form of records within a file and up to 200 such records may be created. The program allows for the records to be displayed on the screen or printed out, and additionally search and sorting options may be used.

The search option will search for a string in a given section of the file, and show all occurences of this string using the forward and backward cursor keys. The sort option sorts the records in the chosen section putting graphics first, then numbers, and finally letters in alphabetical order.

The files created may be stored on tape or disc and reloaded from tape or disc. Note that only one file may exist at any one time, and records cannot be created until a file has been created. Also, if the display of records does not all fit on the screen, that is if there are eight sections within a record, press SHIFT to see the lower display.

DETAILED DESCRIPTION

Lines 10-280 initial set-up and choice of procedures.

290-420 PROC CFILE - creates a file.

430-530 PROC DISPLAY - displays records on file.

540-640 PROC DISPLAY - deals with after-display options, eg. printing out of file.

650-740 PROC CRECORD - creates records, (maximum 200).

750-780 PROC SURE - determines that user is sure of choice of option.

790-1000 PROC SORT - sorts sections of records into

alphabetical order, with graphics first followed by numbers and letters.
1010-1040 PROC DELETE - deletes records.
1050-1390 PROC SEARCH - searches for an entry using given search string.
1400-1740 PROC TAPE - saves a program on tape or disc, and loads a program from tape or disc.
1750-1860 PROC PRINT - prints out a record.
1870-1920 PROC LIMIT - limits amount of information displayed on screen at any time.

```
LIST
   10 *TV255,1
   20 @%=0:*FX4,1
   30 MODE7
   40 DIMR$(7),A$(7,200):P%=0:R%=0
   50 VDU26:CLS:FORI=1TO24:PRINTTAB(0,I-
1);CHR$(129+(I MOD6));:NEXT:VDU28,1,24,3
9,0
   60 REPEAT:CLS:PRINTTAB(7)CHR$141"Book
 Index"'TAB(6)CHR$130CHR$141"Book Index"
'TAB(28);FNM;" bytes"SPC31"free"
   70   VDU31,32,5:IFP%=0 PRINT"No" ELSE
IFP%=1 PRINT"1" ELSEPRINTP%
   80   VDU31,30,6:PRINT"Records":IFP%=1
 VDU31,36,6,32
   90   VDU31,30,7:PRINT"Created"
  100   PRINTTAB(9,3);"Commands"
  110   PRINTTAB(4,6);"1. Clear File"
  120   PRINTTAB(4,8);"2. Search For A R
ecord"
  130   PRINTTAB(4,10);"3. Create A Reco
rd"
  140   PRINTTAB(4,12);"4. Create A File
"
  150   PRINTTAB(4,14);"5. Sort Records"
  160   PRINTTAB(4,16);"6. Display File"
  170   PRINTTAB(4,18);"7. Use Tape Reco
rder"
  180   PRINTTAB(9,21);"Which One ?";
  190   REPEAT:G%=GET:G%=G%-48:UNTILG%>0
```

```
ANDG%<8
  200    IFG%=1PROCsure:IFC%=ØRUN
  210    IFG%=2PROCsearch
  220    IFG%=3PROCcrecord
  230    IFG%=4PROCcfile
  240    IFG%=5PROCsort
  250    IFG%=6PROCdisplay
  260    IFG%=7PROCtape
  270    *FX15,Ø
  280    UNTILFALSE
  290 DEFPROCcfile
  300 IFR%>Ø THENENDPROC
  310 PROCsure:IFC%=1 ENDPROC
  320  REPEAT:CLS:PRINTTAB(4,2);"How Man
y Sections Per"SPC(22)"Record  (1-8) ";
  330    INPUTR%:UNTILR%>ØANDR%<12
  340 I%=Ø
  350 REPEAT:PRINTTAB(4,7)SPC(511)TAB(4,
7)"Name Of Index For Section ";I%+1
  360    INPUTLINER$(I%):IFR$(I%)=""UNTIL
FALSE
  370 I%=I%+1:UNTILI%=R%
  380 CLS:PRINT''"Would You Like A Title
 On The"''"Paper Printouts (Y/N)?";
  390 H%=GET:IFH%<>89ANDH%<>121 TI$="":E
NDPROC
  400 CLS:PRINT'''"Please Enter Title"''
(Up To 79 Characters)"''
  410 INPUTTI$:IFTI$=""ORLENTI$>79THEN4Ø
Ø
  420 ENDPROC
  430 DEFPROCdisplay1
  440 CLS:PRINT"Record No.";Z%;" Out Of
";P%
  450 X%=3
  460 Y%=Ø:VDU28,1,22,39,1,3Ø,14:FORI%=Ø
TOR%-1
  470    PRINTR$(I%)
  480    Z$=A$(I%,Z%-1)
  490    PROClimit
  500    NEXT:VDU15,28,1,24,39,Ø
  510 PRINTTAB(Ø,23);"'P' To Print,'D' T
```

```
o Delete,'E' To End"
  520 PRINTTAB(Ø,24)"'[' Last Record,']'
 Next Record";TAB(Ø,Ø)
  530 ENDPROC
  540 DEFPROCdisplay
  550 Z%=1
  560 IFP%=Ø ENDPROC
  570 PROCdisplay1
  580 H%=GET:IFH%=69 ENDPROC
  590 IFH%=8Ø PROCprint
  600 IFH%=68 PROCdelete:GOTO56Ø
  610 IFH%<>136 THEN63Ø
  620 Z%=Z%-1:IFZ%<1 Z%=1:GOTO57Ø ELSE57
Ø
  630 IFH%<>137 THEN58Ø
  640 Z%=Z%+1:IFZ%>P% Z%=P%:GOTO57Ø ELSE
57Ø
  650 DEFPROCcrecord
  660 IFR%<1 ENDPROC
  670 IFFNM<1ØØ THENCLS:PRINT"No More Ro
om"''"Press A Key":H%=GET:ENDPROC
  680 P%=P%+1:CLS:PRINTTAB(Ø,1);"Record
No.";P%
  690 FORI%=ØTOR%-1:CLS
  700   PRINTTAB(Ø,6);"Section ";I%+1;"
Out Of ";R%'"Called ";R$(I%)'
  710   INPUTLINEZ$:IFZ$=""THEN7ØØ
  720   A$(I%,P%-1)=Z$
  730   NEXT
  740 ENDPROC
  750 DEFPROCsure
  760 CLS
  770 C%=Ø:PRINT''CHR$129"Are You Sure ?
";:IFGET$<>"Y"THENC%=1
  780 ENDPROC
  790 DEFPROCsort
  800 IFP%<1 ENDPROC
  810 PROCsure
  820 IFR%=1 THENS%=Ø:GOTO87Ø
  830 PRINTTAB(Ø,5);"By Which Do You Wan
t To Sort"
  840 FORI%=ØTOR%-1
```

```
  850    PRINT'I%+1;". ";R$(I%):NEXT
  860 REPEAT:S%=GET:S%=S%-48:UNTILS%>ØAN
DS%<R%+1:S%=S%-1
  870 CLS:PRINT"OKAY!!"
  880 N%=P%-1
  890 REPEAT
  900    F%=-1
  910    FORI%=ØTON%-1
  920       IFA$(S%,I%)<=A$(S%,I%+1)THEN99
Ø
  930       F%=Ø
  940       V%=Ø:FORL%=ØTOR%-1
  950          A$=A$(L%,I%)
  960          A$(L%,I%)=A$(L%,I%+1)
  970          A$(L%,I%+1)=A$
  980          NEXT
  990       NEXT:UNTILF%
 1000 ENDPROC
 1010 DEFPROCdelete
 1020 FORI%=ØTOR%-1:A$(I%,Z%-1)="":NEXT:
FORJ%=Z%TOP%-1:FORI%=ØTOR%-1:A$(I%,J%-1)
=A$(I%,J%):NEXT,
 1030    P%=P%-1:IFZ%>P% Z%=P%
 1040    ENDPROC
 1050    DEFPROCsearch
 1060    IFP%<1ENDPROC
 1070    CLS
 1080    IFR%=1 S%=Ø:GOTO1100
 1090    PRINTTAB(Ø,7);"By Which Do You W
ant To Sort":FORI%=ØTOR%-1:PRINT'I%+1;".
 ";R$(I%):NEXT:REPEAT:S%=GET:S%=S%-49:UN
TILS%>-1ANDS%<R%
 1100    CLS:PRINT'''"Please Enter The Se
arch String"'':INPUTLINE"] "S$
 1110    IFS$=""ORLENS$>38 VDU7:GOTO1100
 1120    IFINSTR(S$,"*")>Ø S$=LEFT$(S$,IN
STR(S$,"*"))
 1130    Z%=Ø:Q%=1:B%=1
 1140    Z%=Z%+Q%:IFZ%<1 Q%=1:Z%=Ø
 1150    V%=Ø:IFS%>1 FORK%=1TOS%-1:V%=V%+
R%(K%):NEXT
 1160    T$=A$(S%,Z%-1)
```

```
 1170    IFLEFT$(S$,1)<>"&"THEN1210
 1180    A$=RIGHT$(S$,LENS$-1):IFA$=""THE
N1210
 1190    IFLENA$>LENT$ THEN1210
 1200    IFINSTR(T$,A$)>0 THEN1270
 1210    FORJ%=1TOLENS$
 1220      IFMID$(S$,J%,1)="?"THEN1250
 1230      IFMID$(S$,J%,1)="*"J%=LENS$:NE
XT:GOTO1270
 1240    IFMID$(S$,J%,1)<>MID$(T$,J%,1)J%
=LENS$:NEXT:GOTO1360
 1250 NEXT
 1260 IFLENT$<>LENS$ THEN1370
 1270 IFP%=0 ENDPROC
 1280 B%=0:PROCdisplay1
 1290 H%=GET:IFH%=136 Q%=-1
 1300 IFH%=80 PROCprint
 1310 IFH%<>68 THEN1340
 1320 PROCdelete:IFP%=0 ENDPROC
 1330 GOTO1280
 1340 IFH%=137 Q%=1
 1350 IFH%=69 ENDPROC
 1360 IFZ%+Q%<1 THEN1290
 1370 IFZ%=P% ANDB%=1 CLS:PRINT"There Ar
e No Records That Match"''"Press A Key":H
%=GET:ENDPROC
 1380 IFZ%+Q%>P% THEN1290
 1390 GOTO1140
 1400 DEFPROCtape
 1410 PROCsure:IFC%=1 ENDPROC
 1420 CLS:PRINT"Do You Want To Save Or L
oad A File"''"Press 'S' Or 'L'";
 1430 H%=GET:IFH%<>83 ANDH%<>76 THEN1430
 1440 IFH%=83 ANDP%<1 THENCLS:PRINT"But
There Are No Records To Save"''"Press A K
ey":H%=GET:ENDPROC
 1450 CLS:PRINT"Please Enter The File Na
me"
 1460 INPUTS$:IFS$=""ANDH%=83 PRINT"You
Need A File Name":GOTO1460 ELSEIFLENS$>
10PRINT"Too Long":GOTO1460
 1470 CLS
```

```
 148Ø IFH%=83 THEN162Ø
 149Ø F%=OPENUP(S$)
 15ØØ INPUT#F%,P%
 151Ø PRINTTAB(Ø,5);"There Are ";P%;" Re
cords"'"In This File"
 152Ø INPUT#F%,TI$
 153Ø INPUT#F%,R%
 154Ø FORI%=ØTOR%-1:INPUT#F%,R$(I%):NEXT
 155Ø FORJ%=ØTOP%-1:FORI%=ØTOR%-1
 156Ø      INPUT#F%,Z$:A$(I%,J%)=Z$
 157Ø      NEXT
 158Ø    PRINTTAB(Ø,9);"Record No.";J%+1
 159Ø    NEXT
 16ØØ CLOSE#F%
 161Ø ENDPROC
 162Ø F%=OPENOUT(S$)
 163Ø PRINT#F%,P%
 164Ø PRINTTAB(Ø,5);"There Will Be ";P%;
" Records"'"In This File"
 165Ø PRINT#F%,TI$
 166Ø PRINT#F%,R%
 167Ø FORI%=ØTOR%-1:PRINT#F%,R$(I%):NEXT
 168Ø FORJ%=ØTOP%-1:FORI%=ØTOR%-1
 169Ø      Z$=A$(I%,J%):PRINT#F%,Z$
 17ØØ      NEXT
 171Ø    PRINTTAB(Ø,9);"Record No.";J%+1
 172Ø    NEXT
 173Ø CLOSE#F%
 174Ø ENDPROC
 175Ø DEFPROCprint
 176Ø VDU28,39,Ø,39,Ø,2
 177Ø VDU2,1,18:PRINT"SØ"'"A"
 178Ø IFTI$=""THEN18ØØ
 179Ø PRINTTI$'
 18ØØ FORJ%=ØTOR%-1
 181Ø    PRINT'R$(J%)
 182Ø    Z$=A$(J%,Z%-1)
 183Ø    PROClimit
 184Ø    NEXT
 185Ø PRINT'':VDU3,28,1,24,39,Ø
 186Ø ENDPROC
 187Ø DEFPROClimit
```

```
 1880 IFLENZ$<39 PRINTZ$':ENDPROC
 1890 J%=INSTR(MID$(Z$,28,12),","):IFJ%>
0 PRINTLEFT$(Z$,26+J%):Z$=RIGHT$(Z$,LENZ
$-27-J%):GOTO1880
 1900 J%=INSTR(MID$(Z$,28,12)," "):IFJ%>
0 PRINTLEFT$(Z$,26+J%):Z$=RIGHT$(Z$,LENZ
$-27-J%):GOTO1880
 1910 PRINTLEFT$(Z$,39);:Z$=RIGHT$(Z$,LE
NZ$-39):GOTO1880
 1920 DEFFNM=31740-(?2+?3*256)
```

DANIEL LUCAS
Original program by P. Barker

GENERAL DESCRIPTION

In this game you are trapped in a graveyard at midnight, and all the Zombies are crawling out of their graves to try and catch you! Your task is to run around the graveyard trying to make all the Zombies fall back into their graves. If you fail and you get caught by a Zombie you will be 'Zombified', which is a fate worse than death...

DETAILED DESCRIPTION

Lines 10-290 defines characters, prints the instructions, sets up variables and plays the game.
300-330 loops back if you want to play again.
340-390 inputs the number of ZOMBIES and checks that it is sensible.
400-550 prints the border, the graves and you, chooses a position for each ZOMBIE, checks to see if it is occupied and chooses another if it is or prints a ZOMBIE if it isn't.
560-970 is the main body (sorry) of the game.
580-620 inputs what direction you wish to go.

640-650 checks if it is clear to move or if you have run into a ZOMBIE.
670-750 picks a ZOMBIE at random, works out in what direction it ought to go, checks to see if it has found you or a grave, moves if it hasn't and loops back.
760-800 removes a ZOMBIE (by actually removing the one with the highest number and moving the one that should be removed to the position of the one that has been removed) and loops back if there are any ZOMBIES left.
800-820 adds a few more ZOMBIES for the next turn, sets up the board and goes round again.
830-970 end of game. Plays a tune, prints your score, checks it against the highest and goes round again.
980-1020 defines characters.
1030-1070 checks character on screen at X, Y. Puts the cursor over the character you wish to check, calls OSBYTE 135 and puts the ASCI code of the character (returned in X) into the variable CHAR. As the character definitions are 'imploded', defining character 255, for example, will also define 159, 191, and 223 as the same character and the routine will return with the first one it finds. This explains why lines 650, 720, and 730 appear to be looking for the wrong character.

```
    5REM NOT SUITABLE FOR Ø.1 OP  SYSTEM
   1Ø  HSMIN=Ø:HSSEC=Ø
   2Ø  MODE 1
   3Ø  VDU23;8202;Ø;Ø;Ø;
   4Ø  PROCchar_define
   5Ø  COLOUR1
   6Ø  CLS
   7Ø  PRINT''TAB(15)"ZOMBIES"
   8Ø  PRINT''"          YOU ARE IN A GRA
VEYARD, THE"
   9Ø  PRINT"CHURCH CLOCK HAS JUST STRUC
K MIDNIGHT !"
  1ØØ  PRINT"YOU MUST MUST EVADE THE ZOM
BIES FOR AS"
  11Ø  PRINT"LONG AS POSSIBLE BY LURING
THEM INTO "
```

```
  120  PRINT"OPEN GRAVES ("CHR$255;CHR$2
55CHR$255").BUT IF YOU ARE CAUGHT"
  130  PRINT"YOU ARE ZOMBIFIED"
  140  PRINT''"  KEYBOARD CONTROLS: '
  150  PRINT"  LEFT      Z"
  160  PRINT"  RIGHT     X"
  170  PRINT"  UP        *"
  180  PRINT"  DOWN      ?"
  190  PRINT''"PRESS SPACE TO PLAY ZOMBI
ES"
  200  REPEAT UNTIL GET=32
  210  CLS
  220  COLOUR2
  230  PRINTTAB(2,14)"ZOMBIES"
  240  SW=40:SL=25
  250  DIM Z%(50,1)
  260  VDU19,1,2,0,0,0
  270  PROCinitialise
  280  PROCset_up_board
  290  PROCplay_game
  300  COLOUR1:*FX15,1
  310  PRINT'''"           ANOTHER GAME (
Y/N)"
  320  IF GET$="N" THEN END
  330  GOTO 270
  340  DEFPROCinitialise
  350  INPUT''"How many ZOMBIES do you w
ant 10-50",Z$
  360  Z=VAL(Z$)
  370  IF Z>50 THEN PRINT''"TOO MANY":GO
TO350
  380  IF Z<10 THEN PRINT''"TOO FEW":GOT
O350
  390  ENDPROC
  400  DEFPROCset_up_board
  410  COLOUR2:GCOL0,2
  420  CLS:PRINT'''
  430  B$="   "+CHR$255+CHR$255+CHR$255:B
$=" "+STRING$(7,B$)
  440  FORI=1TO7:PRINTB$'':NEXT
  450  MOVE16,160:DRAW16,940:DRAW1240,94
0:DRAW1240,160:DRAW16,160
```

```
  460  O0=20:O1=13:ZZ=Z
  470  COLOUR1
  480  PRINTTAB(O0,O1)CHR$245:FORI=1TOZ
  490    COLOUR3
  500    P1=RND(SW-3)+1:P2=RND(SL-2)+1
  510    PROCchar_check(P1,P2)
  520    IF CHAR<>32 THEN 500
  530    PRINT TAB(P1,P2)CHR$246:Z%(I,0)
=P1:Z%(I,1)=P2
  540    NEXT
  550  ENDPROC
  560  DEFPROCplay_game
  570  NOW=TIME
  580  D0=0:D1=0
  590  IFINKEY(-98) THEN D0=-1
  600  IFINKEY(-67) THEN D0=1
  610  IFINKEY(-73) THEN D1=-1
  620  IFINKEY(-105) THEN D1=1
  630  COLOUR1
  640  PROCchar_check(O0+D0,O1+D1):IF CH
AR=32 THEN PRINTTAB(O0,O1)" ":O0=O0+D0:O
1=O1+D1:PRINTTAB(O0,O1)CHR$245
  650  IF CHAR=150 THEN PRINTTAB(O0,O1)"
 ":GOTO 840
  660  COLOUR3
  670  J=RND(Z-1)+1
  680  PRINTTAB(Z%(J,0),Z%(J,1));" "
  690  Z%(J,0)=Z%(J,0)-SGN(Z%(J,0)-O0)
  700  Z%(J,1)=Z%(J,1)-SGN(Z%(J,1)-O1)
  710  PROCchar_check(Z%(J,0),Z%(J,1))
  720  IF CHAR=149 THEN 830
  730  IF CHAR=159 OR CHAR=128 THEN 760
  740  PRINTTAB(Z%(J,0),Z%(J,1))CHR$246
  750  GOTO580
  760  SOUND1,-15,20,2
  770  Z%(J,0)=Z%(Z,0)
  780  Z%(J,1)=Z%(Z,1)
  790  Z=Z-1
  800  IF Z>0 THEN GOTO580 ELSE Z=ZZ+RND
(5):IFZ>50THENZ=50
  810  PROCset_up_board
  820  GOTO580
```

```
  830  PRINTTAB(Z%(J,0),Z%(J,1))CHR$246
  840  FORM=20 TO 60 STEP 5
  850    SOUND 1,-15,M,3:NEXT
  860  CLS
  870  MIN=((TIME-NOW) DIV 6000)MOD 60
  880  SEC=((TIME-NOW) DIV 100)MOD 60
  890  COLOUR2
  900  PRINT TAB(5,10)"YOU LASTED ";MIN;
" MINUTES AND ";SEC;" SECONDS"
  910  IF (MIN*60+SEC)<(HSMIN*60+HSSEC)
THEN 960
  920  COLOUR1
  930  PRINT''"      YOU HAVE SURVIVED TH
E LONGEST"
  940  HSMIN=MIN:HSSEC=SEC
  950  COLOUR2
  960  PRINT''" HIGH SCORE IS ";HSMIN;"
MINUTES AND ";HSSEC;" SECONDS"
  970  ENDPROC
  980  DEFPROCchar_define
  990  VDU23,245,28,8,62,93,28,28,20,20
 1000  VDU23,246,56,16,56,56,56,56,40,40
 1010  VDU23,255,255,255,255,255,255,255
,255,255
 1020  ENDPROC
 1030  DEFPROCchar_check(X,Y)
 1040  PRINTTAB(X,Y);
 1050  A%=135
 1060  CHAR=(USR(&FFF4)AND &FF00)/&100
 1070  ENDPROC
```

AIRCRAFT LANDER

JAMES BYRNE
Original program by P. Bailey

GENERAL DESCRIPTION

This program puts you in the position of a pilot faced with landing his aircraft safely on the airport runway. By using different control keys the plane has to be guided down at the right speed, and at the right time, to ensure safe arrival of all aboard.

A continuous display provides a pilot's-eye view of the approaching runway. In addition, a number of instrument readings give essential details of such things as altitude, speed, direction, pitch, roll, and distance from runway. Keys may be used to manoeuvre the aeroplane in terms of its speed, roll and pitch, the pilot's efforts being reflected in the instrument readings. Realistic sound effects accompany the simulation, which will soon discover whether you can say with honesty "I'll take good care of you!"

DETAILED DESCRIPTION

Lines 10-250 main program which calls a number of procedures to read user inputs, perform calculations, and change the screen display. Landings, crash or otherwise, produce different endings to the simulation.
260-390 procedures to help with screen displays.
400-870 initialisation procedure. Asks user for wind speed and direction, sets a variety of constants to their starting values, outputs display heading.
880-920 procedure to calculate aircraft velocity in X, Y directions.
930-1030 procedure to interpret in-flight control keys.
1040-1100 procedure to interpret control keys during landing.
1110-1390 procedures to calculate relative position of

aircraft and to effect its movement.
1400-1670 procedure to update screen display - produces new picture of runway, and new values for instrument readings.
1680-1920 procedures to check for, and report on, crash landings!
1930-1960 procedure to report a safe landing - most unlikely to be obeyed!
1970-2070 procedures to display information on final position of aircraft, and to produce variable length delay.
2080-2470 procedure to display operating instructions.
2480-2590 procedure, and data, for safe landing fanfare!

```
   10  REM AIRCRAFT LANDER
   20  MODE 7
   30  PROCscreen
   40  PRINTTAB(0,10)"Do you require instruct
ions?";
   50  REPEAT IN$=GET$:UNTIL IN$="Y" OR IN$="
N"
   60  IF IN$="Y" PROCinstructions
   70  MODE 4
   80  PROCsetup
   90  REPEAT
  100    PROCkeys
  110    PROCupdate_pos
  120    PROCupdate_screen
  130    PROCkeys
  140    UNTIL PZ<=0
  150  PROCcrash_check
  160  IF LF=1 THEN PROCcrash:PROCend:END
  170  PRINTTAB(16,4)"TOUCHDOWN";
  180  PROCland_keys
  190  REPEAT
  200    PROCmove
  210    PROCupdate_screen
  220    UNTIL LF=2 OR LF=3
  230  IF LF=2 PROCoff_runway:PROCend
  240  IF LF=3 PROCgood_land
  250  END
  260  DEF PROCnext_page
```

```
  270  PRINT TAB(0,20)"Press any key to conti
nue";
  280  *FX 15,0
  290  KEY=GET
  300  CLS
  310  ENDPROC
  320  DEF PROCscreen
  330  PRINT TAB(3,0)CHR$132;CHR$157;TAB(36,0
);CHR$156
  340  FOR Y%=1 TO 2
  350    PRINTTAB(3,Y%)CHR$132;CHR$157;CHR$13
4;CHR$141;"AIRCRAFT LANDING SIMULATOR   ";CHR
$156
  360    NEXT
  370  PRINT TAB(3,3)CHR$132;CHR$157;TAB(36,3
);CHR$156
  380  VDU 28,0,24,39,4
  390  ENDPROC
  400  DEF PROCsetup
  410  DIM Q(11,3),R(11,2),TS(9)
  420  C1=PI/180:C2=180/PI
  430  C5=.5
  440  LF=0:REM LANDING FLAG
  450  PZ=3000:REM ALTITUDE
  460  PY=-2E4:REM DISTANCE FROM RUNWAY
  470  PX=0:REM LATERAL DEVIATION
  480  RL=0:REM ROLL
  490  PT=0:REM PITCH
  500  AS=150:REM AIRSPEED
  510  AD=0:REM AIR DIRECTION
  520  PROCcalcvel
  530  CLS
  540  PRINT"Please select wind speed and dir
ection. A direction of 0 means that the wind
is blowing straight at you and this gives  th
e easiest landing since you do not    need to
 alter roll and bearing. The"
  550  PRINT"direction should be between -90
(from   the left) and +90 (from the right). A
   wind speed of 5 is a light breeze, a    sp
eed of 30 a gale."
  560  REPEAT
```

```
  570    PRINT TAB(0,10)STRING$(39," ")
  580    INPUTTAB(0,10)"WIND SPEED (M/S)",X0
  590    UNTIL X0>0 AND X0<35
  600  REPEAT
  610    PRINT TAB(0,10)STRING$(39," ")
  620    INPUTTAB(0,10)"WIND DIRECTION (DEG)"
,X1
  630    UNTIL X0>=-90 AND X0<=90
  640  WY=-X0*COS(X1*C1)
  650  WX=-X0*SIN(X1*C1)
  660  VZ=0
  670  GZ=VZ:GY=VY+WY:GX=VX+WX
  680  TC=5:REM THROTTLE CONTROL
  690  BC=0:REM BANKING CONTROL
  700  NC=0:REM NOSE UP/DOWN CONTROL
  710  RC=0:REM RUDDER CONTROL
  720  FD=0:REM BRAKES
  730  FOR I=0 TO 9:READ TS(I):NEXT
  740  DATA 0,25,50,75,100,150,200,250,275,30
0
  750  YT=20:REM YAW TOLERANCE
  760  RT=3:REM ROLL TOLERANCE
  770  TP=5:REM PITCH TOLEARANCE
  780  LR=2000:REM RUNWAY LENGTH
  790  WR=50:REM RUNWAY WIDTH
  800  HD=3E4:REM DISTANCE TO HORIZON
  810  CLS
  820  VDU 23,224,255,255,255,255,255,255,255
,255
  830  VDU 23;10,32;0;0;0;
  840  PRINT"SPEED BEAR PITCH ROLL ALT   DIST
   DEV"
  850  PRINTTAB(0,4)STRING$(39,CHR$224)
  860  PRINTTAB(0,31)STRING$(39,CHR$224);
  870  ENDPROC
  880  DEF PROCcalcvel
  890  RA=AD*C1
  900  VX=AS*SIN(RA)
  910  VY=AS*COS(RA)
  920  ENDPROC
  930  DEF PROCkeys
  940  X$=INKEY$(0)
```

```
  950  IF ASC(X$)>47 AND ASC(X$)<58 THEN TC=V
AL(X$)
  960  IF X$="F" BC=-1
  970  IF X$="G" BC=Ø
  980  IF X$="H" BC=1
  990  IF X$="M" NC=1
 1000  IF X$="K" NC=Ø
 1010  IF X$="O" NC=-1
 1020  *FX 15,1
 1030  ENDPROC
 1040  DEF PROCland_keys
 1050  X$=INKEY$(Ø)
 1060  IF X$="R" RC=1
 1070  IF X$="E" RC=-1
 1080  IF X$="C" RC=Ø
 1090  IF X$="B" FD=2
 1100  ENDPROC
 1110  DEF PROCupdate_pos
 1120  PZ=PZ+GZ
 1130  PY=PY+GY
 1140  PX=PX+GX
 1150  PT=PT+NC
 1160  RL=RL+BC
 1170  AS=AS+2*SGN(TS(TC)-AS)
 1180  AD=AD+RL
 1190  VZ=1*(TC-5)+AS*SIN(PT*C1)
 1200  PROCcalcvel
 1210  GZ=VZ
 1220  GY=VY+WY
 1230  GX=VX+WX
 1240  GD=-ATN(VX/VY)*C2
 1250  SOUND Ø,-10,7,15
 1260  IF AS<155 SOUND 1,Ø,AS+100,15 ELSE  SO
UND 1,Ø,255,15
 1270  ENDPROC
 1280  DEF PROCmove
 1290  PT=Ø:RL=Ø:PZ=Ø
 1300  PROCland_keys
 1310  PX=PX+VX
 1320  PY=PY+VY
 1330  IF ABS(PX)>WR OR PY>Ø LF=2
 1340  AD=AD+AD*(SGN(AD)*SGN(RC))
```

```
 135Ø  AS=AS-FD:IF AS<=Ø THEN LF=3
 136Ø  SOUND Ø,-1Ø,7,15
 137Ø  IF AS<155 SOUND 1,Ø,AS+1ØØ,15 ELSE  SO
UND 1,Ø,255,15
 138Ø  PROCcalcvel
 139Ø  ENDPROC
 14ØØ  DEF PROCupdate_screen
 141Ø  HT=-PZ-5:LD=-PX-WR:RD=-PX+WR
 142Ø  HD=HT*7:IF HD<-PY THEN HD=-PY
 143Ø  IF HD<3ØØØ THEN HD=3ØØØ
 144Ø  FOR I=Ø TO 4
 145Ø    YD=-PY-I*5ØØ
 146Ø    Q(I,1)=LD:Q(I+5,1)=RD
 147Ø    Q(I,2)=YD:Q(I+5,2)=YD
 148Ø    Q(I,3)=HT:Q(I+5,3)=HT
 149Ø    NEXT
 15ØØ  Q(1Ø,1)=-PX:Q(1Ø,2)=-PY:Q(1Ø,3)=HT
 151Ø  Q(11,1)=-PX:Q(11,2)=-PY-LR:Q(11,3)=HT
 152Ø  A1=64/WR:A2=64.5:A3=15.5:A4=31
 153Ø  FOR I=Ø TO 11
 154Ø    AØ=(HD-Q(I,2))/HD
 155Ø    R(I,1)=INT(Q(I,1)*AØ*A1+A2)
 156Ø    R(I,2)=INT(A3+AØ*A4)-3
 157Ø    NEXT
 158Ø  VDU 28,Ø,3Ø,39,5,12,26
 159Ø  MOVE R(4,1)*1Ø,1Ø24-(R(4,2)*21.7)
 16ØØ  DRAW R(Ø,1)*1Ø,1Ø24-(R(Ø,2)*21.7)
 161Ø  DRAW R(5,1)*1Ø,1Ø24-(R(5,2)*21.7)
 162Ø  DRAW R(9,1)*1Ø,1Ø24-(R(9,2)*21.7)
 163Ø  DRAW R(4,1)*1Ø,1Ø24-(R(4,2)*21.7)
 164Ø  MOVE R(1Ø,1)*1Ø,1Ø24-(R(1Ø,2)*21.7)
 165Ø  PLOT 21,R(11,1)*1Ø,1Ø24-(R(11,2)*21.7)
 166Ø  PRINT TAB(Ø,Ø);TAB(Ø);INT(AS+C5);TAB(6
);INT(AD+C5);TAB(11);INT(PT+C5);TAB(17);INT(R
L+C5);TAB(22);INT(PZ+C5);TAB(28);INT(ABS(PY)+
C5);TAB(35);INT(PX+C5);" ";
 167Ø  ENDPROC
 168Ø  DEF PROCcrash_check
 169Ø  IF ABS(RL)>RT OR PT>TP OR PT<Ø OR ABS(
AD)>YT OR AS<45 OR AS>55 THEN LF=1:PRINTTAB(1
2,4)"CRASH ON LANDING";
 17ØØ  IF ABS(PX)>WR OR PY>Ø OR PY<-LR THEN L
```

```
F=1:PRINTTAB(15,5)"OFF RUNWAY";
 1710  IF LF=1 PROCdelay(500)
 1720  ENDPROC
 1730  DEF PROCcrash
 1740  CLS
 1750  VDU 19,1,8;0;
 1760  PRINTTAB(15,15)"FATAL CRASH"
 1770  *FX 15,1
 1780  WAIT=GET
 1790  CLS
 1800  VDU 19,1,7;0;
 1810  PRINT"THE FLIGHT RECORDER HAS BEEN REC
OVERED"
 1820  PRINT"THE FINAL FLIGHT DETAILS WERE AS
 FOLLOWS"
 1830  ENDPROC
 1840  DEF PROCoff_runway
 1850  CLS
 1860  IF PY>0 THEN PRINT"YOU HAVE GONE OFF T
HE END OF THE RUNWAY.TRY USING YOUR BRAKES NE
XT TIME."
 1870  IF ABS(PX)>WR THEN PRINT"YOU ARE OFF T
HE SIDE OF THE RUNWAY.      ISN'T IT WIDE ENOU
GH FOR YOU?"
 1880  IF AS<=10 THEN PRINT"LUCKILY THERE IS
ONLY MINOR DAMAGE AT   THIS SPEED."
 1890  IF AS>10 AND AS<=25 THEN PRINT"SERIOUS
 DAMAGE TO YOUR AIRCRAFT, BUT YOUSURVIVE TO S
CARE YOUR PASSENGERS TO      DEATH AGAIN."
 1900  IF AS>25 THEN PRINT'"DISASTER - FATAL
CRASH"
 1910  PRINT'"DETAILS OF THE FLIGHT WHEN YOU
LEFT THE RUNWAY:-"'
 1920  ENDPROC
 1930  DEF PROCgood_land
 1940  PRINT TAB(0,4)"CONGRATULATIONS CAPTAIN
 - A SAFE LANDING"
 1950  PROCtune
 1960  ENDPROC
 1970  DEF PROCend
 1980  PRINT"SPEED = ";AS;" M/S"
 1990  PRINT"DIRECTION = ";AD;" DEG"
```

```
 2000  PRINT"DISTANCE FROM END OF RUNWAY = ";
ABS(PY);" M"
 2010  PRINT"DISTANCE FROM CENTRE OF RUNWAY =
 ";ABS(PX);" M"
 2020  PRINT"PITCH = ";PT;" DEG"
 2030  PRINT"ROLL = ";RL;" DEG"
 2040  ENDPROC
 2050  DEF PROCdelay(T)
 2060  TIME=0:REPEAT UNTIL TIME=T
 2070  ENDPROC
 2080  DEF PROCinstructions
 2090  CLS
 2100  PRINT'"It is dark and raining. You are
          approaching the airport with 300
    passengers on board and your co-pilot   h
as just become unconscious, so you mustland t
he aircraft alone. To be honest"
 2110  PRINT"he was not doing a great job bef
ore he  passed out."
 2120  PRINT'"You are in level flight at an a
ltitude  of 3000 M but only 20 KM from the fa
r   end of the runway. Your aircraft is     p
ointing in the correct direction but"
 2130  PRINT"any crosswind will cause a devia
tion    from this path."
 2140  PROCnext_page
 2150  PRINT"You must land with an airspeed o
f about 50 M/S with no appreciable roll or pi
tchRoll must be less than 3 degrees and    pi
tch between 0 and 5 degrees. Since therunway
is 2000 M long the distance"
 2160  PRINT"indicator must show less than th
is      figure on touchdown. Similarly the
   runway is 100 M wide so that if the     de
viation is greater than 50 M on       touchdo
wn you will miss the runway."
 2170  PRINT'"After touchdown you must use th
e rudder to correct the direction of travel a
nd  the brakes to stop before you reach the e
nd of the runway (this is indicated by"
 2180  PRINT"a distance reading of 0). If you
 fail tostop or correct your direction (which
```

```
   probably will not be Ø when landing in  a
crosswind) you will crash."
 219Ø  PROCnext_page
 22ØØ  PRINT"You can see the runway lights ah
ead and in the distance the lights of the cit
y  you hope to avoid."
 221Ø  PRINT'"Your instruments give the follo
wing:"
 222Ø  PRINT"SPEED airspeed in M/S"
 223Ø  PRINT"BEAR  bearing - direction in whi
ch             aircraft is pointing. Ø is
          straight ahead positive values to
    the right."
 224Ø  PRINT"PITCH positive values when aircr
aft's          nose is up."
 225Ø  PRINT"ROLL  positive values clockwise.
 Roll          causes banking which changes th
e         aircraft's bearing."
 226Ø  PRINT"ALT   height above ground in M."
 227Ø  PRINT"DIST  distance from the far end
of the         runway in M."
 228Ø  PRINT"DEV   sideways deviation from th
e centre       of the runway."
 229Ø  PROCnext_page
 23ØØ  PRINT "In flight you have the followin
g         controls:"
 231Ø  PRINT'"F  roll to left"
 232Ø  PRINT"G  maintain roll at this level"
 233Ø  PRINT"H  roll to right"
 234Ø  PRINT"M  nose up (increase pitch)"
 235Ø  PRINT"K  maintain pitch at this level"
 236Ø  PRINT"O  nose down (decrease pitch)"
 237Ø  PRINT'"Ø-9 throttle control"
 238Ø  PRINT"    Each setting has a terminal
value of    speed which is reached eventually
.       E.G. 5 - 15Ø M/S, 2 - 5Ø M/S.
  Landing is usually achieved on          set
ting 2."
 239Ø  PROCnext_page
 24ØØ  PRINT'"After touchdown only the follow
ing have effect:"
 241Ø  PRINT'"E  rudder left"
```

```
 2420  PRINT"R  rudder right"
 2430  PRINT"B  apply brakes"
 2440  PRINT'"N.B. All controls have a delaye
d effect on a large aircraft. In general they
    have effect on the second display after p
ressing the key."
 2450  PRINT'"GOOD LUCK"
 2460  PROCnext_page
 2470  ENDPROC
 2480  DEF PROCtune
 2490  RESTORE 2560
 2500  FOR notes%=1 TO 44
 2510    READ P%,D%
 2520    SOUND 1,-15,P%,D%*1.5
 2530    SOUND 1,0,0,1
 2540    NEXT
 2550  ENDPROC
 2560  DATA 33,2,41,2,49,2,53,4,33,5,53,2,49,
2,53,2,61,4,41,5
 2570  DATA 41,2,49,2,53,2,69,2,61,2,61,2,53,
2,53,2,49,2,41,2,49,2,33,4
 2580  DATA 33,2,41,2,49,2,53,4,33,5,53,2,49,
2,53,2,61,4,41,5
 2590  DATA 41,2,49,2,53,2,69,2,61,2,61,2,53,
2,53,2,49,2,41,2,49,2,53,4
```

DANIEL LUCAS
Original program by Alan Green

GENERAL DESCRIPTION

This is a simple but addictive game that uses sound and colour to good effect.

Menacing Aliens are trying to colonise the earth and to save us all you must shoot them down before they land. Inexorably, the nearer they get the higher they score, and you have got to act very fast - only five have to land to take over the planet!

DETAILED DESCRIPTION

Lines 10-200 set up defined characters, variables, print instructions if required and set up the screen.
210-250 choose where the Alien will start, how it moves and what type it is.
260-280 disable the cursor, print the laser base and the Alien, (with spaces to wipe out the previous position), and set a value for the laser base position.
300-370 input a move for the laser base, fire the laser if required, work out the new position of the Alien and print the new position of the laser base.

380-420 input a move for the laser base, jump out if the Alien has landed, or loop back if it hasn't.
430-460 chalk up another success for the Aliens and loop back if less than five have landed.
470-520 detect if a laser bolt hit an Alien and, if so, destroy it and increase the score.
530-590 end of game: print your score, check it against the highest, play a tune, wait for you to press SPACE and go round again.
600-910 DATA for defined characters and instructions.

```
LOADLIST
   1Ø  ENVELOPE1,3,Ø,Ø,Ø,Ø,Ø,Ø,121,-1Ø,-
5,-2,12Ø,12Ø
   2Ø  MODE1
   3Ø  PROCintro
   4Ø  REM *** Alien Blaster ***
   5Ø  REM ***   Converted   ***
   6Ø  REM ***  By D. Lucas  ***
   7Ø  REM ***      1983      ***
   8Ø  FOR M=251 TO 254
   9Ø    READ A,B,C,D,E,F,G,H
  1ØØ    VDU 23,M,A,B,C,D,E,F,G,H
  11Ø    NEXT M
  12Ø  SS=Ø
  13Ø  A=2Ø:S=Ø:Q=1:Z=31:Y=12Ø
  14Ø  PRINT'''"          Do you want inst
ructions"
  15Ø  A$=GET$:IF A$="Y"THENPROCinstruct
ions ELSE IF A$<>"N" THEN 15Ø
  16Ø  VDU19,2,3,Ø,Ø,Ø:COLOUR2:CLS
  17Ø  *FX11,1
  18Ø  *FX12,1Ø
  19Ø  PRINTTAB(Ø,2)"--------------------
----------------------";TAB(Ø,31)"-------
----------------------------------";
  2ØØ  PRINTTAB(2,Ø)"Score:";S:PRINTTAB(
2Ø,Ø);"High:";SS
  21Ø  C=RND(38):B=3
  22Ø  IF C<=7 THEN W=1
  23Ø  IF C>7 AND C<32 THEN W=Ø
  24Ø  IF C>=32 THEN W=-1
```

```
  250  CH=RND(2)
  260  VDU23;8202;0;0;
  270  COLOUR3:PRINTTAB(A,29)" "CHR$(252
)" "
  280  PRINTTAB(C-1,B-1);"      "TAB(C,B)
CHR$(252+CH)
  290  X=(A*32)+48
  300  A$=INKEY$(0)
  310  IF A$="1" AND A>0 THEN A=A-1
  320  IF A$="2" AND A<37 THEN A=A+1
  330  MOVEX,100
  340  IF A$="0" THEN DRAW X,950:SOUND1,
1,20,5:GCOL0,0:DRAWX,100:GCOL0,2:GOTO460

  350  C=C+W
  360  IF W=0 THEN B=B+.5 ELSE B=B+1
  370  COLOUR3:PRINTTAB(A,29)" "CHR$(252
)" "
  380  A$=INKEY$(0)
  390  IF B=30 THEN 420
  400  IF A$="2" AND A<37 THEN A=A+1
  410  GOTO 270
  420  VDU7
  430  PRINTTAB(C-1,29);"    ";TAB(Q*2,31
)CHR$(252+CH);
  440  IF Q=5 THEN GOTO 520
  450  Q=Q+1:GOTO210
  460  IF X=(C*32)+16 THEN 480
  470  GOTO 350
  480  FORDE=1TO10:NEXT
  490  PRINTTAB(C,B);CHR$(251):SOUND 0,1
,4,2
  500  FORDE=1TO300:NEXT DE
  510  S=S+INT(B+.5+(CH*2)):PRINT TAB(8,
0);S;TAB(C,B);" ":GOTO210
  520  PRINTTAB(15,15)"GAME OVER"
  530  IF S>SS THEN SS=S
  540  FORM=1TO2:SOUND1,-15,40,10:SOUND1
,-15,100,10:NEXT
  550  PRINTTAB(7,18)"Press SPACE to pla
y again"
  560  REPEAT UNTIL GET=32
```

```
  570  VDU7:GOTO130
  580  END
  590  DATA28,20,100,150,137,85,38,24
  600  DATA0,24,24,153,153,255,129,129
  610  DATA60,126,219,126,90,129,66,40
  620  DATA60,36,60,126,255,36,66,129
  630  DEFPROCinstructions
  640  *FX12,50
  650  *FX11,50
  660  CLS
  670  PRINT"ALIEN BLASTER"''"===== =====
==="
  680  PRINT''"  The Aliens are invading
your system,  your only hope is to blast
 them before  they land and colonise the
 earth !"
  690  VDU19,2,15,0,0,0:COLOUR2
  700  PRINT''TAB(15)"WARNING..."
  710  COLOUR7
  720  PRINT''"  Only five aliens have t
o land to take over !!!"
  730  PRINT'"    Press any key for next
 page"
  740  *FX15,0
  750  VDU7:REPEAT UNTIL GET<>0
  760  CLS
  770  PRINT''"  To move base right pres
s '2'"
  780  PRINT'"  To move base left press
 '1'"
  790  PRINT'"  To fire press
 '0'"
  800  PRINT''"  There are two different
 aliens one    scores more than the othe
r"
  810  PRINT"  The aliens are worth more
 the nearer  they are to the planet."
  820  PRINT'';CHR$253;"...Raider"'';CH
R$254;"...Droid"
  830  PRINT''"   Press any key to start
."
  840  REPEAT UNTIL GET<>0
```

```
850  ENDPROC
860  DEFPROCintro
870  PRINT TAB(13,9)"Alien Blaster"
880  FOR M=30 TO 100 STEP 5
890    SOUND 1,-15,M,2
900    NEXT M:ENDPROC
```

STOCKMARKET

BRIAN JONES
Original program by Bob Chappell

GENERAL DESCRIPTION

This simulation gives you a chance to make money with your micro, but unfortunately you lose it all when you switch the power off!

You are dealing in four commodities: GOLD, TIN, ZINC and LEAD. If you buy and sell at the right time, you can make your fortune, but watch out for takeovers or you could lose your shirt. On each turn your portfolio, profit and loss account, and the market trends are displayed in colour in MODE 7, and you can choose to buy or sell shares. After a few games you may feel ready for the real thing!

DETAILED DESCRIPTION

Lines 30 call a procedure (1450-2005) which sets up arrays and prints instructions.
40-140 define procedures for incorrect data entry, continue to next page, wait 1.5 seconds, and print heading for newsflash.
150-182 set starting bank balance.
190-410 print menu and buy or sell shares.
420-670 print market news.
680-990 print newsflash.
1000-1190 print profit and loss account.
1200-1350 print portfolio and go round again.
1360-1430 market collapse and end of game.
2099-2104 useful procedure for printing a string in double height, anywhere on the screen in any colour.
2106-2113 defined functions for selecting a random number, and print formatting.

```
LIST
   10 REM**STOCK MARKET**BOB CHAPPELL**1
4/2/82**
   20 REM**ACKNOWLEDGEMENTS TO A FLEET**
   25
   26 MODE 7
   30 PROCinstructions : GOTO 150
   35
   40 DEF PROCfraud
   50 PRINT'SPC(8); : VDU 136,157,132 :
PRINT "FRAUD!!   "; : VDU 156
   60 PROCpause : ENDPROC
   65
   69 DEF PROCspace
   70 PRINT TAB(0,24); : VDU 157,132 : P
RINT "PRESS SPACE TO CONTINUE  ";
   80 REPEAT : A$=GET$ : UNTIL A$=" "
   81 CLS
   90 ENDPROC
   95
  110 DEF PROCpause
  111 T%=TIME : REPEAT UNTIL TIME>T%+150
 : ENDPROC
  112
  120 DEF PROCnewsflash
  130 PRINT'SPC(8); : VDU 136,157,132 :
PRINT "NEWSFLASH  ";CHR$(156)
  140 PROCpause : ENDPROC
  145
  150 CLS : PROCdouble(10,0,"STOCK MARKE
T",1)
  160 PRINT ''"HOW MUCH DO YOU WANT IN Y
OUR BANK"
  170 REPEAT : PRINT SPC(40);CHR$(11);
  180   INPUT "ACCOUNT ",B$ : B=INT(VAL(
B$))
  181   UNTIL B>0 AND B<100000
  182 XZ=1 : GOTO 430
  183
  190 PRINT 'CHR$(130);"BUYING AND SELLI
NG"
  191 FOR J=1 TO 6 : PRINT 'J;". ",M$(J)
```

```
 : NEXT
  200 INPUT "SELECTION",C%
  201 IF C%<1 OR C%>6 THEN PROCfraud : G
OTO 430
  220 IF C%=5 THEN GOTO 440
  230 IF C%=6 THEN TT=1 : GOTO 1010
  240 IF F(C%)=1 THEN PRINT"DEALING SUSP
ENDED" : GOTO 430
  250 PRINT : INPUT"HOW MANY SHARES",S%
  260 IF S%=0 THEN PROCfraud : GOTO 430
  270 PRINT : INPUT"BUY OR SELL (B/S)",A
$
  280 IF A$<>"B" AND A$<>"S" THEN PROCfr
aud : GOTO 430
  290 R=FNA(R) : IF R=0 THEN PRINT"MARKE
T SUSPENDED"
  300 P=V(C%)*(5^(4-C%)) : T=S%*P : IF A
$="B" THEN 370
  310 IF S%>C(C%) THEN PROCfraud : GOTO
430
  320 B=B+T : C(C%)=C(C%)-S%
  330 J=S%*P(C%) : IF J=T THEN PRINT 'CH
R$(131);"NO LOSS OR PROFIT ",:GOTO 360
  340 IF J<T THEN PRINT'CHR$(132);"A PRO
FIT OF"T-J; : PL(C%)=PL(C%)+(T-J) : GOTO
 360
  350 PRINT'CHR$(129);"A LOSS OF"J-T; :
PL(C%)=PL(C%)-(J-T)
  360 PRINT : GOTO 390
  370 IF B<T THEN PRINT 'CHR$(133);"INAD
EQUATE FUNDS" : GOTO 430
  380 B=B-T : TC=P(C%)*C(C%) : C(C%)=C(C
%)+S% : P(C%)=(TC+T) DIV C(C%)
  390 R=FNA(R) : IF R>0 THEN 430
  400 R=FNA(R) : IF R<5 THEN 400
  410 PROCspace : GOTO 810
  420 REM ** MARKET NEWS **
  430 PROCspace
  440 L1=0 : FOR J= 1 TO 5 : R=FNA(R) :
IF R>4 THEN F(J)=0
  450   NEXT : CLS : PROCdouble(4,0,"MAR
KET NEWS",4)
```

```
  451 PRINT TAB(18,Ø);CHR$(14Ø);CHR$(134
);"* = RECOMMENDED"
  452 FOR J=1 TO 4
  46Ø    PRINT'J;". ";M$(J);" "; : IF F(J
)=1 THEN PRINT "SUSPENDED": GOTO56Ø
  47Ø    R=FNA(R) : IF R=Ø THEN 68Ø
  48Ø    X=R-INT((V(J)*.4)+.9) : IF X=Ø T
HEN PRINT "HOLD"
  49Ø    OP=V(J)*(5^(4-J)) : V(J)=V(J)+X
  5ØØ    CP=V(J)*(5^(4-J)) : Q=22  : IF X
<=Ø THEN 52Ø
  51Ø    PRINT "UP ";SPC(FNB(INT(CP-OP)))
;INT(CP-OP); : GOTO 53Ø
  52Ø    IF X<Ø THEN PRINT "DOWN ";SPC(FN
B(INT(CP-OP)));INT(CP-OP);
  53Ø    Q=37 : PRINT TAB(25);CHR$(129);"
PRICE";CHR$(135);
  531    PRINT SPC(FNB(INT(CP)));INT(CP);
  54Ø    IF CP<A(J) THEN PRINT "*": GOTO
56Ø
  55Ø    PRINT
  56Ø    NEXT : IF L1<Ø THEN PRINT 'CHR$(
129);"TOTAL LOSS OF"ABS(L1)
  57Ø IF L1>Ø THEN PRINT 'CHR$(132);"TOT
AL PROFIT OF"L1
  58Ø L1=Ø : IF XZ=1 THEN XZ=Ø : PROCspa
ce : CLS : GOTO 19Ø
  59Ø PRINT''CHR$(13Ø);"    BANK ";:R=FNA
(R): IF R=Ø THEN PRINT:PROCnewsflash:GOT
O 63Ø
  6ØØ X=INT(R-(.2*I)) : I=I+X
  61Ø IF I>2Ø THEN I=2Ø
  62Ø PRINT "RATE ";I;"%" : GOTO 67Ø
  63Ø R=FNA(R) : IF R<>Ø THEN 65Ø
  64Ø PRINT SPC(5);"FAILS ": L1=L1-B : P
L(5)=PL(5)-B : B=Ø : GOTO 66Ø
  65Ø PRINT SPC(5);"SUSPENDED" : F(5)=1
  66Ø IF L1<Ø THEN PRINT'CHR$(129)"LOSS
OF ";ABS(L1)
  67Ø PROCspace : GOTO8ØØ
  68Ø PROCnewsflash : F(J)=1 : R=FNA(R)
: IF R<7 THEN 74Ø
```

```
  690 PRINT SPC(5);"TAKEOVER ";: PROCpau
se : R=FNA(R)
  700 IF R=0 THEN PRINT "SUSPENDED" : GO
TO 560
  710 PRINT CHR$(131);"SELL AT"; : P=21*
(5^(4-J)) : D=(20+(R/10))/100 : PRINTINT
(P*D)
  720 T=INT(P*D*C(J)): B=B+T : T1=T-C(J)
*P(J) : L1=L1+T1 : PL(J)=PL(J)+T1
  730 C(J)=0 : V(J)=1 : P(J)=0 : GOTO 56
0
  740 IF R>2 THEN PRINT SPC(5);"SUSPENDE
D" : F(J)=1 : GOTO 560
  750 PRINT SPC(10);"BANKRUPT "
  760 L1=L1-(C(J)*P(J))
  770 PL(J)=PL(J)-(C(J)*P(J))
  780 C(J)=0 : P(J)=0 : V(J)=1 : GOTO 56
0
  790 REM ** FLASH **
  800 L2=0 : R=FNA(R) : IF R<5 THEN 1010
  810 L2=0 : CLS : PRINT''''' : PROCnew
sflash : IF R>7 THEN 890
  820 R=FNA(R) : IF R>4 OR R=0 THEN 870
  830 J=R : PRINT CHR$(130);M$(J);" BONU
S "; : R=FNA(R) : PROCpause
  840 IF R=0 THEN PRINT "SUSPENDED" : GO
TO 970
  850 R=10*R : PRINTCHR$(135);R;"%" : CP
=V(J)*(5^(J-4)) : L2=INT(CP/100*R)*C(J)
  860 B=B+L2 : PL(J)=PL(J)+L2 : GOTO970
  870 PRINTSPC(6);CHR$(134);"TAX BONUS "
;:PROCpause : R=FNA(R) : IF R=0 OR B<1 T
HEN R=0 : GOTO 840
  880 R=10*R : PRINTCHR$(135);R;"%" : L2
=INT(B/100*R) : B=B+L2 : PL(6)=PL(6)+L2
: GOTO 970
  890 R=FNA(R) : IF R=0 THEN 1360
  900 IF R<5 THEN 940
  910 PRINT'SPC(6);CHR$(134);"SUPER TAX
";:PROCpause : R=FNA(R) : IF R=0 THEN 84
0
  920 R=10*R : PRINTCHR$(135);R;"%" : T=
```

```
INT(B/100*R) : PL(6)=PL(6)-ABS(T) : B=B-
ABS(T)
  930 L2=L2-ABS(T) : GOTO 970
  940 PRINT'SPC(6);CHR$(134);"BONUS ISSU
E";:T=P(R)*C(R)
  950 C(R)=C(R)+(INT(C(R)/2))
  960 IF C(R)<>0 THEN P(R)=T DIV C(R)
  970 IF L2<0 THEN PRINT''SPC(5);CHR$(1
29);"A LOSS OF "ABS(L2)
  980 IF L2>0 THEN PRINT''SPC(5);CHR$(1
32);"A PROFIT  OF "L2
  990 PROCspace
 1000 REM ** P/L ACCOUNT **
 1010 GOSUB 1020 : GOTO 1210
 1020 CLS : IF TT=1 THEN A$="CLOSING " E
LSE A$=""
 1030 PROCdouble(4,0,A$+"PROFIT AND LOSS
 ACCOUNT ",2) : T=0 : Q=29 : FOR J=1 TO
4
 1040    PRINT'TAB(9);M$(J);
 1050    PRINT SPC(FNC(PL(J)));:IF PL(J)<
0 THEN PRINT CHR$(129);
 1060    PRINT ABS(PL(J))
 1070    T=T+PL(J) : NEXT : L2=0 : IF TT=
1 THEN 1090
 1080 IF F(5)<>1 THEN L2=B DIV (100*I) :
 B=B+L2
 1090 F(5)=0 : PL(5)=PL(5)+L2 : PRINT'TA
B(9)"INTEREST";SPC(FNC(PL(5)));
 1100 IF PL(5)<0 THEN PRINT CHR$(129);
 1110 PRINT ABS(PL(5))
 1120 PRINT'TAB(9)"TAX "SPC(FNC(PL(6)));
 1130 IF PL(6)<0 THEN PRINTCHR$(129);
 1140 PRINT ABS(PL(6))
 1150 T=T+PL(5)+PL(6)
 1160 PRINT''TAB(9)"TOTAL"SPC(FNC(T));:
IF T<0 THEN PRINT CHR$(129);
 1170 PRINT ABS(T)
 1180 PRINT''TAB(23);CHR$(129);"RED";CHR
$(135);"= LOSS"
 1190 PROCspace : RETURN
 1200 REM ** PORTFOLIO **
```

```
 1210 CLS : IF TT=1 THEN A$="CLOSING " E
LSE A$=""
 1220 PROCdouble(4,0,A$+"PORTFOLIO ",5)
 1221 PRINT LL$:PRINT TAB(19);"ORIGINAL"
;TAB(31);"CURRENT"
 1230 PRINT"  METAL     SHARES     COST
    PRICE" : PRINT LL$'
 1240 FOR J=1 TO 4 : Q=14 : PRINT"  "M$(
J);SPC(FNC(C(J)));
 1250    PRINT C(J);" "; : IF C(J)=0 THEN
 1270
 1260    Q=25 : PRINT SPC(FNC(P(J)));P(J)
;
 1270    Q=36 : CP=V(J)*(5^(4-J)):PRINT S
PC(FNC(CP));CP':NEXT
 1280 PRINT''"      BANK"SPC(FNC(B));B
 1290 IF B<0 THEN PRINT CHR$(129);"
OVERDRAWN"
 1300 IF B<-999 AND C(1)=0 AND C(2)=0 AN
D C(3)=0 AND C(4)=0 THEN1330
 1310 PRINT:PROCspace: IF C%=6 THEN 1430
 1320 GOTO 190
 1330 PRINT''CHR$(129);" YOU ARE BANKRUP
T AND YOUR NEW"
 1340 PRINTCHR$(129);" ADDRESS IS QUEER
STREET!!" : END
 1350 REM** END **
 1360 CLS : PROCdouble(10,0,"MARKET COLL
APSE",1) : PROCpause
 1361 PRINT''CHR$(129);" BANK TAKEOVER "
 : PROCpause
 1370 Q=23 : PRINT' : FOR J=1 TO 4 : PRI
NT'J;". "M$(J)" SOLD AT ";
 1380    R=FNA(R) : P=21*(5^(4-J)) : T=(1
00-(10/R))/100:R=INT(P*T) : L2=C(J)*R
 1390    B=B+L2 : T=C(J)*P(J)
 1400    PL(J)=PL(J)+(L2-T)
 1410    PRINTSPC(FNB(R));R : NEXT
 1420 PRINT''CHR$(132)"CLOSING BANK BALA
NCE "B : PROCspace : TT=1 : GOSUB 1020
 1430 CLS : PROCdouble(8,11,"END OF SPEC
ULATIONS!!",5) : VDU 26 : END
```

```
 1440 REM ** INITIALISE **
 1450 DEF PROCinstructions
 1460 DIM M$(6),F(5),P(4),PL(6),V(4),A(4
),C(4)
 1470 R=RND(10) : FOR I=1 TO 6 : READ M$
(I) : NEXT
 1471 DATAGOLD,TIN,ZINC,LEAD,PASS,QUIT
 1472 FORI=1TO4 : READ A(I) : NEXT
 1473 DATA1500,300,60,12
 1480 LL$="_____________________________
___________"
 1490 TT=0
 1620 PROCdouble(10,0,"STOCK MARKET",1)
 1630 PRINT''' : INPUT "DO YOU WANT INST
RUCTIONS (Y/N) ",A$
 1640 IF LEFT$(A$,1)<>"Y" THEN ENDPROC
 1650 CLS : PROCdouble(10,0,"STOCK MARKE
T",2)
 1660 PRINT'"YOU ARE FIRST ASKED HOW MUC
H CAPITAL"
 1670 PRINT'"YOU WANT TO START WITH."
 1680 PRINT'"DEALING IS IN 4 METALS :-"'
'"GOLD, TIN, SILVER AND LEAD."
 1690 PRINT''"VARIOUS REPORTS WILL HELP
YOU CONTROL"
 1700 PRINT'"YOUR SPECULATIONS."
 1710 PRINT''"THE GAME ENDS WHEN YOU DEC
IDE TO QUIT,"
 1720 PRINT'"OR IF YOU GET TOO DEEPLY IN
 DEBT,"
 1730 PRINT'"OR IF THE MARKET COLLAPSES.
"
 1740 PROCspace : CLS : PROCdouble(4,0,"
MARKET NEWS EXPLANATIONS",3)
 1750 PRINT 'CHR$(129);"UP";CHR$(135);"P
RICE HAS RISEN"
 1760 PRINT 'CHR$(129);"DOWN";CHR$(135);
"PRICE HAS FALLEN"
 1770 PRINT 'CHR$(129);"HOLD";CHR$(135);
"UNCHANGED"
 1780 PRINT 'CHR$(129);"SUSPENDED";CHR$(
135);"NO DEALING ALLOWED"
```

```
 1790 PRINT 'CHR$(129);"TAKEOVER";CHR$(1
35);"SHARES AUTOMATICALLY SOLD AT"
 1800 PRINT SPC(10);"PRICE SHOWN"
 1810 PRINT 'CHR$(129);"TAKEOVER SUSPEND
ED";CHR$(135);"NARROW ESCAPE!"
 1820 PRINT 'CHR$(129);"BANKRUPT";CHR$(1
35);"SHARES FORFEITED"
 1830 PROCspace
 1840 CLS : PROCdouble(10,0,"BANK RULES"
,2)
 1850 PRINT ''CHR$(129);"RATE X%";CHR$(1
35);"BANK ACCOUNT INCREASED BY X%"
 1860 PRINT SPC(9);"(IF OVERDRAWN INTERE
ST PAYABLE"'SPC(9);"ON OVERDRAFT)"
 1870 PRINT''CHR$(129);"SUSPENDED";CHR$(
135);"NO INTEREST PAID"
 1880 PRINT''CHR$(129);"FAILS";CHR$(135)
;"ALL MONEY LOST (NEW BANK ARISES"
 1890 PRINT SPC(7);"NEXT TURN)"
 1900 PROCspace : CLS : PROCdouble(4,2,"
PROFIT AND LOSS ACCOUNT",1)
 1910 PRINT'"SHOWS NET GAINS AND LOSSES"
 1920 PROCdouble(10,7,"PORTFOLIO",1)
 1930 PRINT'"SHOWS SHARES HELD AND BANK
ACCOUNT"
 1940 PROCspace : CLS : PROCdouble(10,0,
"NEWSFLASHES",3)
 1941 PRINT'CHR$(129);"TAX BONUS X%";CHR
$(135);"BANK BALANCE CREDITED"
 1942 PRINT'CHR$(129);"SUPER TAX X%";CHR
$(135);"BANK BALANCE DEBITED"
 1950 PRINT'CHR$(129);"METAL BONUS X%";C
HR$(135);"BANK BALANCE CREDITED"
 1951 PRINT SPC(16);"WITH INCREASE IN VA
LUE"'SPC(16);"OF HOLDINGS"
 1952 PRINT'CHR$(129);"METAL BONUS ISSUE
";CHR$(135);"HOLDINGS INCREASED"
 1953 PRINT SPC(19);"BY 1 SHARE FOR EVER
Y"'SPC(19)"2 HELD"
 1960 PRINT 'CHR$(130);"ANY OF THE ABOVE
 MAY BE SUSPENDED"
 1961 PRINT 'CHR$(130);"AT THE LAST MOME
```

```
NT"
 1962 PRINT'CHR$(129);"MARKET FAILS";CHR
$(135);"END OF GAME - ALL"
 1963 PRINT SPC(14);"HOLDINGS SOLD AT CU
RRENT":PRINT SPC(14);"MARKET PRICE"
 2000 PROCspace : CLS : PRINT''''''
 2001 PRINT CHR$(129);"  YOU ARE NOW ABO
UT TO BECOME"
 2002 PRINT CHR$(129);"  VERY RICH OR VE
RY POOR"
 2003 PRINT ''"    BUT REMEMBER IT'S ONLY
 A GAME!!"
 2005 PROCspace : ENDPROC
 2098
 2099 DEF PROCdouble(X,Y,A$,C)
 2100 LOCAL I%
 2101 FOR I%=0 TO 1
 2102   PRINT TAB(X-2,Y+I%);CHR$(141);CH
R$(128+C);A$
 2103   NEXT I%
 2104 ENDPROC
 2105
 2106 DEF FNA(R) : =RND(10)-1
 2107
 2108 DEF FNB(Z)
 2109 B%=Q-COUNT-LEN(STR$(INT(Z)))-(ABS(
Z)<1)
 2110 IF B%>0 THEN =B% ELSE=0
 2111 DEF FNC(X)
 2112 B%=Q-COUNT-LEN(STR$(X))
 2113 IF B%>0 THEN =B% ELSE=0
```

ROCKET ATTACK

PHILIP LINDSAY
Original program by John Gooderson

GENERAL DESCRIPTION

A game for up to four people, acting as commanders of their own missile launching sites. Their task - to defend a nuclear power station against a rocket attack: all they have to do is to shoot down the rockets!

The Commander can control his missile by using the numeric keys 1-5, and the space bar; the former guide the missile's left/right bearing, and the latter gives it emergency boost as it nears its target. Failure to hit the rocket means that the power station under threat is likely to have a lump knocked out of it! Each Commander can volunteer for up to three postings on active service, assuming that the strain is not too great. End-of-posting reports are produced which give success/failure ratings, together with recommendations as to the Commander's future career development!

DETAILED DESCRIPTION

Lines 10-460 initialisation of graphics characters, output of rules.
Inputs number of Commanders and how many postings.
470-540 main program, calling procedures to control the game.
550-790 procedures to draw power station, launch rocket.
800-1000 graphics procedures plotting paths of rocket and missile.
1010-1400 procedure for end of rocket! Controls the effect of the rocket if it has hit the power station.
1410-1680 procedures to produce Commander's report, and start new campaign.
1690-1970 various small procedures.

Note Those with disc systems should set Page=&E00 before typing in this listing.

```
   10 REM ************************************
***************
   20 REM ROCKET ATTACK BY JOHN GOODERSON. -
COPYRIGHT 1981.
   30 REM ************************************
***************
   40 MODE 1
   50 ON ERROR MODE 7 : END
   60 VDU23,240,24,24,24,24,24,60,60,60
   70 VDU23,241,60,60,255,255,255,255,255,255
   80 VDU23,242,255,255,255,255,255,255,255,2
55
   90 VDU23,243,60,60,60,60,60,60,60,60
  100 VDU23,244,1,1,1,1,1,127,127,127
  110 VDU23,245,199,199,199,199,199,248,248,2
48
  120 VDU23,246,127,127,127,1,1,1,1,1
  130 VDU23,247,248,248,248,199,199,199,199,1
99
  140 VDU23,248,248,248,248,7,7,7,255,255
  150 VDU23,249,31,31,31,224,224,224,255,255
  160 VDU23,250,255,7,7,7,7,7,7,0
  170 VDU23,251,255,224,224,224,224,224,224,0
  180 VDU23,252,&08,&08,&1C,&1C,&1C,&3E,&7F,&
6B
  190 CLS: PRINTTAB(13,2)"ROCKET ATTACK"
  200 PRINT: PRINT
  210 PRINT"  A NUMBER OF NUCLEAR POWER STATI
ONS ARE";
  220 PRINT"LOCATED IN GREAT BRITAIN AND THEY
 ARE"
  230 PRINT "ALL UNDER IMMINENT ROCKET ATTACK
 FROM A"
  240 PRINT "FOREIGN POWER. EACH POWER STATIO
N IS"
  250 PRINT"DEFENDED BY A COMPUTER CONTROLLED
"
  260 PRINT "MISSILE LAUNCHING SITE."
  270 PRINT
  280 PRINT "  MISSILES ARE GUIDED BY THE MIS
SILE"
  290 PRINT"SITE COMMANDER WHO USES KEYS 1-5
```

```
FOR"
  300 PRINT "NORMAL GUIDANCE AND THE SPACE BA
R FOR"
  310 PRINT"EMERGENCY BOOST. ANY KEY CAN BE U
SED"
  320 PRINT
  330 PRINT "  EACH MISSILE SITE COMMANDER IS
 RATED"
  340 PRINT"FOR ABILITY TO DEFEND THE POWER"
  350 PRINT "STATIONS. AS MIGHT BE EXPECTED T
HE"
  360 PRINT"RATING IS BASED ON A COMBINATION
OF"
  370 PRINT"SKILL AND LUCK."
  380 PRINT:INPUT "HOW MANY SITE COMMANDERS A
RE THERE? "CN%
  390 IF CN%<1 THEN 380
  400 IF CN%>4 THEN PRINT"THERE ARE ONLY 4 TR
AINED COMMANDERS": PRINT"AVAILABLE IN GREAT B
RITAIN": GOTO 380
  410 DIM E%(CN%): DIM EE%(3,CN%)
  420 PRINT:INPUT"HOW MANY POSTINGS TO ACTIVE
 SERVICE CAN EACH COMMANDER UNDERTAKE? "PN%
  430 IF PN%<1 THEN PRINT"ALL COWARDS WILL BE
 SHOT AT DAWN.":GOTO420
  440 IF PN%>3 THEN PRINT"ALL COMMANDERS GO I
NSANE AFTER MORE THAN 3 POSTINGS.":GOTO 420
  450 VDU 23;8202;0;0;0;
  460 U%=675: V%=180: CC%=0: PC%=1: CLS:PRINT
  470 FOR COM%=1 TO CN%
  480   CC%=CC%+1: PC%=1
  490   FOR POST%=1 TO PN%
  500     PROCGAME
  510     NEXTPOST%
  520   NEXTCOM%
  530 PROCRATING: PROCEND
  540 RUN
  550 DEF PROCGAME
  560 IF Z%=25 THEN Z%=Z%-1: PROCSTATOK
  570 PRINTTAB(2,20)"COMMANDER NUMBER. ";CC%
  580 PRINT "  PRESS 'P' FOR POSTING NO. ";PC
%
```

```
  590 A$=GET$: IF A$<>"P" THEN 590
  600 REM DRAW STATION AND LAUNCHING
  610 CLS: PRINT TAB(2,3)"AN ATTACK IS IMMINE
NT"
  620 FOR T=0 TO 1000: NEXT: PRINT"  STAND BY
": FOR T=0 TO 1000: NEXT
  630 CLS
  640 PRINTTAB(21,27)CHR$(240)
  650 PRINTTAB(21,28)CHR$(241)
  660 FOR X%=0 TO 79: GCOL 0,RND(3): PLOT 69,
RND(1280),RND(1024): NEXT
  670 COLOUR 1
  680 FOR D%=29 TO 30: FOR C%=20 TO 22: PRINT
TAB(C%,D%)CHR$(242): NEXT C%: NEXT D%
  690 FOR D%=26 TO 30: FOR C%=5 TO 9: PRINTTA
B(C%,D%)CHR$(242): NEXT C%: NEXT D%
  700 FOR D%=25 TO 26: FOR C%=5 TO 9: PRINTTA
B(C%,D%)CHR$(243): NEXT C%: NEXT D%
  710 COLOUR 130
  720 PRINT TAB(0,31)STRING$(39," ");
  730 COLOUR 128
  740 Y%=1: W%=0: X%=0: Z%=0: STAT=0
  750 PROCGRAPH1
  760 PROCGRAPH2
  770 IF K%<35 THEN PROCENDROC ELSE 760
  780 IF STAT=0 THEN 750
  790 ENDPROC
  800 DEF PROCGRAPH1
  810 IF Z%<>0 THEN H%=X%*100/Z%: PRINTTAB(24
,25);"AIMING ";H%"%"
  820 J%=1200: K%=900: Z%=Z%+1: R%=40+3*RND(5
)
  830 S%=(100-(W%*100/11))*(11-W%)/(26-Z%): I
F S%>99 THEN S%=99
  840 PRINTTAB(12,16)"SURVIVAL CHANCE ";S%"%"
  850 PRINTTAB(24,26);"ROC. LAUNCHED ";Z%
  860 *FX 15,1
  870 PROCMOVEMIS
  880 J%=J%-R%
  890 VDU5: GCOL 4,1: PROCMIS: PROCHROC: PROC
MIS: PROCHROC: GCOL 0,1
  900 IF U%>J%-30 AND U%<J%+40 AND V%>K%-30 A
```

```
ND V%<K%+4Ø THEN PROCHITROC: GOTO 77Ø
  91Ø VDU4: IF V%>1Ø24 THEN V%=18Ø: U%=675: Y
%=Y%+1
  92Ø IF J%>RND(35Ø) THEN 87Ø
  93Ø ENDPROC
  94Ø DEF PROCGRAPH2
  95Ø PROCMOVEMIS
  96Ø K%=K%-R%
  97Ø VDU5: GCOL 4,1: PROCMIS: PROCVROC: PROC
MIS: PROCVROC: GCOL Ø,1
  98Ø IF U%>J%-3Ø AND U%<J%+4Ø AND V%>K%-3Ø A
ND V%<K%+4Ø THEN PROCHITROC: GOTO 77Ø
  99Ø VDU4: IF V%>1Ø24 THEN V%=18Ø: U%=675: Y
%=Y%+1
 1ØØØ ENDPROC
 1Ø1Ø DEF PROCENDROC
 1Ø2Ø IF J%=12ØØ THEN ENDPROC
 1Ø3Ø IF K%=-1 THEN ENDPROC
 1Ø4Ø D%=31-RND(6): W%=W%+1
 1Ø5Ø IF J%>168 AND J%<25Ø THEN PROCHITSTAT:
ENDPROC
 1Ø6Ø PRINTTAB(J%/31,D%);"WOOMF"
 1Ø7Ø FOR T%=Ø TO 1ØØØ: NEXT
 1Ø8Ø PRINTTAB(J%/31,D%);"      "
 1Ø9Ø ENDPROC
 11ØØ DEF PROCHITSTAT
 111Ø COLOUR 3
 112Ø PRINTTAB(2,28)"*  VAROOM  *"
 113Ø FOR NN%=4TO5
 114Ø   SOUNDØ,-15,NN%,1Ø
 115Ø   SOUNDØ,-15,NN%+1,1Ø
 116Ø   NEXT NN%
 117Ø FOR T%=1 TO 1ØØØØ: NEXT
 118Ø CLS: STAT=-1
 119Ø PRINTTAB(1,4);"SURVIVAL CHANCE Ø%"
 12ØØ PRINTTAB(1,5);"AIMING ABILITY "H%;"%"
 121Ø PRINTTAB(1,6);"ROCKETS LAUNCHED ";Z%
 122Ø PRINTTAB(1,7);"MISSILES FIRED ";Y%
 123Ø PRINTTAB(1,8);"ROCKETS DESTROYED "X%
 124Ø PRINTTAB(1,9);"HITS ON STATION "W%
 125Ø FOR T%=Ø TO 1ØØ: NEXT
 126Ø PRINTTAB(1,11);"NUCLEAR POWER STATION D
```

```
ESTROYED"
 1270 PRINTTAB(1,12);"ALL BUT YOU DEAD WITHIN
 50 MILE RADIUS"
 1280 FOR T%=0 TO 2000: NEXT
 1290 E%=H%*S%/100: EE%(PC%,CC%)=E%: PC%=PC%+
1: Y%=1: J%=1200
 1300 PRINTTAB(1,14);"YOUR RATING ON THIS POS
TING WAS ";E%"%"
 1310 ENDPROC
 1320 DEF PROCSTATOK
 1330 CLS
 1340 PRINTTAB(1,4)"ROCKETS LAUNCHED ";Z%
 1350 PRINTTAB(1,5)"AIMING ABILITY ";H%"%"
 1360 PRINTTAB(1,6)"SURVIVAL CHANCE 100%"
 1370 PRINTTAB(1,7)"YOU SURVIVED THE ATTACK-C
ONGRATULATIONS"
 1380 FOR T%=1 TO 4000: NEXT
 1390 CLS
 1400 ENDPROC
 1410 VDU 11,9,127
 1420 DEF PROCRATING
 1430 PRINTTAB(1,16)"PRESS 'R' FOR OVERALL RA
TING"
 1440 PRINTTAB(1,17)"AND COMMANDERS' PROSPECT
S"
 1450 A$=GET$: IF A$<>"R" THEN 1450
 1460 CLS: PRINT: PRINT: PRINT: PRINT: PRINT
 1470 FOR ZZ%=1 TO CN%
 1480   E%(ZZ%)=(EE%(1,ZZ%)+EE%(2,ZZ%)+EE%(3,
ZZ%))/PN%
 1490   IF E%(ZZ%)<101 THEN P$="PROMOTION AND
 KNIGHTHOOD"
 1500   IF E%(ZZ%)<91 THEN P$="PROMOTION AND
DECORATION"
 1510   IF E%(ZZ%)<81 THEN P$="DECORATION FOR
 VALOUR"
 1520   IF E%(ZZ%)<71 THEN P$="MENTIONED IN D
ISPATCHES"
 1530   IF E%(ZZ%)<61 THEN P$="TO BE CONGRATU
LATED"
 1540   IF E%(ZZ%)<51 THEN P$="PUT IN COMMAND
 OF NAFFI"
```

```
 1550    IF E%(ZZ%)<41 THEN P$="TØ BE DEMOTED"
 1560    IF E%(ZZ%)<31 THEN P$="REDUCED TO THE
 RANKS"
 1570    IF E%(ZZ%)<21 THEN P$="TO BE CASHIERE
D"
 1580    IF E%(ZZ%)<11 THEN P$="TO BE SHOT IMM
EDIATELY"
 1590    PRINT: PRINT "COMMANDER NO.";ZZ%;" RA
TING  ";E%(ZZ%);"%"
 1600    PRINTP$: PRINT
 1610    NEXT ZZ%
 1620 ENDPROC
 1630 DEF PROCEND
 1640 PRINT: PRINT: PRINT
 1650 PRINT"PRESS 'N' FOR NEW CAMPAIGN"
 1660 A$=GET$: IF A$="" THEN 1660
 1670 IF A$<>"N" THEN END
 1680 ENDPROC
 1690 DEF PROCHROC
 1700 MOVE J%,K%
 1710 VDU 244,245,8,8,10,246,247
 1720 VDU 127,127,11,9,9,127,127
 1730 ENDPROC
 1740 DEF PROCMIS
 1750 MOVE U%,V%
 1760 VDU 252
 1770 ENDPROC
 1780 DEF PROCVROC
 1790 MOVE J%,K%
 1800 VDU 248,249,8,8,10,250,251
 1810 ENDPROC
 1820 DEF PROCMOVEMIS
 1830 A%=51-INKEY(0): *FX 15,1
 1840 IF A%=52 AND V%=180 THEN PRINTTAB(14,15
);"MISSILE ";Y%;" READY": GOTO 1910
 1850 IF A%=19 THEN V%=V%+40
 1860 IF A%<-2 OR A%>2 THEN A%=0
 1870 V%=V%+30: U%=U%-20*A%
 1880 PRINTTAB(12,16);"                      "
 1890 PRINTTAB(14,15);"                       "
 1900 PRINTTAB(24,27);"MISSILES SHOT ";Y%
 1910 ENDPROC
```

```
 192Ø DEF PROCHITROC
 193Ø SOUNDØ,-15,4,15:VDU8: PRINT"BOOM";
 194Ø FOR T%=Ø TO 1ØØØ: NEXT
 195Ø VDU 127,127,127,127
 196Ø X%=X%+1: V%=1Ø25: K%=-1: VDU4: PRINTTAB
(24,24);"ROCKETS HIT ";X%
 197Ø ENDPROC
```

TRUTH

PAUL BORRETT
Original program by M. Costello

GENERAL DESCRIPTION

TRUTH is a game that will test your ability to follow and apply a logical statement in the selection of a series of playing cards. There are three skill levels at which this game can be played, and this determines the complexity of the logical rules chosen. Although the basic level should present no problem, the advanced level could easily ensure that you go round in logical circles...

Having chosen the level of play, a randomly selected rule is displayed to you which states what constitutes a correct or an incorrect choice of card. Having absorbed that, the game begins and you have to choose your first card. The entire pack is displayed and you indicate your selection of the suit, and then the denomination by use of the cursor keys. The card chosen is graphically displayed and you are then told whether it fits the rule. Success requires that eight correct cards be chosen from the pack, and as some extra help your previous choices are always displayed on the screen in the same order as they were selected.

DETAILED DESCRIPTION

Lines 10-60 program heading and variable initialisation.
70-80 instructions are provided if required.
90-120 ask for the skill level and call appropriate procedures and display rule chosen.
130-180 main loop which displays the pack, allows the player to choose a card and signals if it is right or wrong. Also tests for winning, giving up or running out of cards.
190-460 applies the current rule to the card selected and determines outcome.
470-590 chooses a rule randomly for the basic level.

600-810 the basic level rules in text form.
820-950 chooses a rule for the intermediate level.
960-1160 the intermediate level rules in text form.
1170-1260 chooses a rule for the advanced level.
1270-1380 advanced level rules in text form.
1390-1410 string initialisation.
1420-2600 the logical tests for all 24 rules (E=0 means wrong, E=1 means right).
2610-2620 displays headings and suit names on screen.
2630-2650 displays what remains of the pack.
2660-2830 card selected was correct so tell the player the good news.
2840-2930 wrong choice!
2940-3110 variable initialisation and setting up of the pack from the data statements.
3120-3300 invites player to select a card. Tests for giving up (Z key pressed).
3310-3440 displays card chosen in graphical form.
3450-3490 clears some space on the screen and calls the card display procedure.
3500-3530 illegal choice - card already taken.
3540-3580 procedure to print text strings of unknown length making sure words are not split up when a new line is taken.
3600-3630 player gives up.
3640-3680 displays the chosen rule.
3690-3750 eight correct cards - success! Appropriate congratulatory noises are issued.
3760-3820 entire pack gone - player loses.
3830-3910 player is offered another game.
3920-4040 one screen full of instructions.

```
   10 *FX4,1
   20 *TV255,1
   30 MODE7
   40 DIMP$(13,3),H(13,3),D$(13),R$(13),S$(3)
,J1(12),J2(12),JV(3)
   50 CLS:PRINT'''"T R U T H":PRINT''''"Do yo
u want instructions (Y/N)";
   60 PROCINIT
```

```
   70 Z=GET:IFZ=78 THEN90
   80 IFZ=89 PROCINTRO ELSE70
   90 CLS:PRINT''''"T R U T H":PRINT''''"Pleas
e choose your level of play:":PRINT''"1.  "U1
$'"2.  "U2$'"3.  "U3$:PRINT''''"Type 1,2 or 3"
  100 Z$=GET$
  110 IFZ$="1"US$=U1$:PROCBASIC ELSEIFZ$="2"U
S$=U2$:PROCINTER ELSEIFZ$="3"US$=U3$:PROCADVA
N ELSE100
  120 CLS:C5=0:K=0
  130 VDU30:*FX15,1
  140 PROCLAYOUT:PROCCARDSLEFT:PROCGETKEY
  150 IFZ<>90 THEN170
  160 PROCGIVEUP:IFZ=89 PROCEND ELSE130
  170 PROCTEST:IFCT=0 PROCLOSE ELSEIFK=7 PROC
WIN
  180 FORA=0TO2:JV(A)=JV(A+1):NEXT:JV(3)=JT:G
OTO130
  190 DEFPROCTEST
  200 IFCT=52 E=1:GOTO450
  210 IFGC=1 PROCRULE1
  220 IFGC=2 PROCRULE2
  230 IFGC=3 PROCRULE3
  240 IFGC=4 PROCRULE4
  250 IFGC=5 PROCRULE5
  260 IFGC=6 PROCRULE6
  270 IFGC=7 PROCRULE7
  280 IFGC=8 PROCRULE8
  290 IFGC=9 PROCRULE9
  300 IFGC=10 PROCRULE10
  310 IFGC=11 PROCRULE11
  320 IFGC=12 PROCRULE12
  330 IFGC=13 PROCRULE13
  340 IFGC=14 PROCRULE14
  350 IFGC=15 PROCRULE15
  360 IFGC=16 PROCRULE16
  370 IFGC=17 PROCRULE17
  380 IFGC=18 PROCRULE18
  390 IFGC=19 PROCRULE19
  400 IFGC=20 PROCRULE20
  410 IFGC=21 PROCRULE21
  420 IFGC=22 PROCRULE22
```

```
  430 IFGC=23 PROCRULE23
  440 IFGC=24 PROCRULE24
  450 IFE=1 PROCRIGHT ELSEPROCWRONG
  460 CT=CT-1:ENDPROC
  470 DEFPROCBASIC
  480 PROCNULLSTRING
  490 GC=RND(8)
  500 GC=8
  510 IFGC=1 PROCMAKERULE1
  520 IFGC=2 PROCMAKERULE2
  530 IFGC=3 PROCMAKERULE3
  540 IFGC=4 PROCMAKERULE4
  550 IFGC=5 PROCMAKERULE5
  560 IFGC=6 PROCMAKERULE6
  570 IFGC=7 PROCMAKERULE7
  580 IFGC=8 PROCMAKERULE8
  590 ENDPROC
  600 DEFPROCMAKERULE1
  610 S1$=X3$:S2$=XA$:S3$=V5$:S4$=V1$:ENDPROC
  620 DEFPROCMAKERULE2
  630 S1$=X4$:S2$=CHR$8+V1$:S3$=X5$+" ":S4$=V
7$:ENDPROC
  640 DEFPROCMAKERULE3
  650 L=RND(13):N$=R$(L):L1=L+6:IFL1<14 NS$=R
$(L1) ELSEL1=L1-13:NS$=R$(L1)
  660 IFL1=13 Q$=R$(1) ELSEQ$=R$(L1+1)
  670 L2=L1+6:IFL2<14 QS$=R$(L2) ELSEL2=L2-13
:QS$=R$(L2)
  680 S1$=N$+" to "+NS$:S2$=CHR$8+V1$:S3$=Q$+
" to "+QS$:S4$=" "+V7$
  690 FORA=0TO12:J1(A)=0:J2(A)=0:NEXT:FORA=0T
O6:J1(A)=L:L=L+1:IFL=14 L=1
  700    NEXT:FORA=0TO5:J2(A)=L:L=L+1:IFL=14 L
=1
  710    NEXT:ENDPROC
  720 DEFPROCMAKERULE4
  730 S1$=X3$:S2$=V3$:S3$=XC$+XD$:S4$=S$(3)+"
 "+S$(1)+" "+S$(0)+" "+S$(2)+" "+S$(3)+" etc"
:ENDPROC
  740 DEFPROCMAKERULE5
  750 S1$=X3$:S2$=X3$:S3$=XC$:S4$=XD$:S5$=XE$
+" (first two always right)":ENDPROC
```

```
  760 DEFPROCMAKERULE6
  770 S1$=X3$:S2$=X6$:S3$=V8$+" ":S4$=X8$:S5$
=V1$:ENDPROC
  780 DEFPROCMAKERULE7
  790 S1$=X3$:S2$=X6$:S3$=V8$+" ":S4$=X7$:S5$
=V1$:ENDPROC
  800 DEFPROCMAKERULE8
  810 S1$=X2$:S2$=XF$:S3$=XG$:ENDPROC
  820 DEFPROCINTER
  830 PROCNULLSTRING
  840 GC=RND(10)
  850 IFGC=1 PROCMAKERULE9
  860 IFGC=2 PROCMAKERULE10
  870 IFGC=3 PROCMAKERULE11
  880 IFGC=4 PROCMAKERULE12
  890 IFGC=5 PROCMAKERULE13
  900 IFGC=6 PROCMAKERULE14
  910 IFGC=7 PROCMAKERULE15
  920 IFGC=8 PROCMAKERULE16
  930 IFGC=9 PROCMAKERULE17
  940 IFGC=10 PROCMAKERULE18
  950 GC=GC+8:ENDPROC
  960 DEFPROCMAKERULE9
  970 S1$=X3$:S2$=X3$:S3$=XC$:S4$=XD$:S5$=S$(
1)+" "+S$(3)+" "+S$(0)+" "+S$(2)+" "+S$(1)+"
etc":ENDPROC
  980 DEFPROCMAKERULE10
  990 S1$=X6$:S2$=X6$:S3$=XH$:ENDPROC
 1000 DEFPROCMAKERULE11
 1010 S1$=V3$:S2$="it":S3$=XC$:S4$="this rule
:"+NL$+"divide"+V1$+" by 4,"+NL$+"if remainde
r = 3 then play "+S$(3)+","+NL$+"if = 2 then
play "+S$(2)+","+NL$+"if = 1 then play "+S$(1
)+","+NL$+"else play "+S$(0):ENDPROC
 1020 DEFPROCMAKERULE12
 1030 S1$=X2$:S2$=CHR$8+V1$:S3$=X4$:S4$=" and
 "+X1$+" is correct if"+V1$+" is "+X5$:ENDPRO
C
 1040 DEFPROCMAKERULE13
 1050 S1$=X3$:S2$=X6$:S3$="4 to 6 ":S4$=X8$:S
5$=V1$:ENDPROC
 1060 DEFPROCMAKERULE14
```

```
 1070 S1$=X3$:S2$=X6$:S3$="8 to 10 ":S4$=X7$:
S5$=V1$:ENDPROC
 1080 DEFPROCMAKERULE15
 1090 S1$=X3$:S2$=V3$+" or "+X6$:S3$=V4$:S4$=
V1$:ENDPROC
 1100 DEFPROCMAKERULE16
 1110 L5=RND(4)-1:L6=L5:L5=L5+1:L7=L5 MOD4:L5
=L5+1:L8=L5 MOD4:L5=L5+1:L9=L5 MOD4:T6$=S$(L6
):T7$=S$(L7):T8$=S$(L8):T9$=S$(L9)
 1120 S1$=X4$+" "+X6$:S2$=V1$:S3$=T6$+" or "+
T7$+" and ":S4$=X5$+" "+X6$+" is correct if"+
V1$+" is "+T8$+" or "+T9$:ENDPROC
 1130 DEFPROCMAKERULE17
 1140 S1$=X1$:S2$=XF$:S3$=XG$:S4$=" ":S6$=X2$
:S7$=XF$:S8$=Y3$:ENDPROC
 1150 DEFPROCMAKERULE18
 1160 S1$=X3$:S2$=X6$:S3$="within the range o
f 2 less or 2 more than"+V1$:ENDPROC
 1170 DEFPROCADVAN
 1180 PROCNULLSTRING
 1190 GC=RND(6)
 1200 IFGC=1 PROCMAKERULE19
 1210 IFGC=2 PROCMAKERULE20
 1220 IFGC=3 PROCMAKERULE21
 1230 IFGC=4 PROCMAKERULE22
 1240 IFGC=5 PROCMAKERULE23
 1250 IFGC=6 PROCMAKERULE24
 1260 GC=GC+18:ENDPROC
 1270 DEFPROCMAKERULE19
 1280 S1$=X4$:S2$=CHR$8+V1$:S3$="same "+XA$+"
 as ":S4$=V2$+" one":S6$=X5$:S7$=CHR$8+V1$:S8
$="different "+XA$+" from ":S9$=V2$+" one":EN
DPROC
 1290 DEFPROCMAKERULE20
 1300 S1$="ace to 7":S2$=V2$+" 2":S3$=X1$:S6$
="8 to king":S7$=V2$+" 2":S8$=X2$:ENDPROC
 1310 DEFPROCMAKERULE21
 1320 S1$=X4$:S2$=XF$+" but 3":S3$=Y3$:S6$=X5
$:S7$=S2$:S8$=XG$:ENDPROC
 1330 DEFPROCMAKERULE22
 1340 S1$="Every third card":S2$="it":S3$=XC$
+XD$:S4$=S$(1)+" "+S$(0)+" "+S$(3)+" "+S$(2):
```

```
S5$="(other cards are always correct)":ENDPRO
C
 1350 DEFPROCMAKERULE23
 1360 S1$="picture "+X3$:S2$=CHR$8+V1$:S3$="n
ot "+S1$:X4$=V7$:ENDPROC
 1370 DEFPROCMAKERULE24
 1380 S1$=X2$:S2$=V6$+XG$+"s":S3$=X8$+" or eq
ual to ":S4$=V6$+Y3$+" cards":S6$=X1$:S7$=S4$
:S8$=X8$+" ":S9$=S2$:ENDPROC
 1390 DEFPROCNULLSTRING
 1400 S1$="":S2$="":S3$="":S4$="":S5$="":S6$=
"":S7$="":S8$="":S9$=""
 1410 ENDPROC
 1420 DEFPROCRULE1
 1430 E=0:IF(F=0ORF=3)AND(HS=1ORHS=2) E=1 ELS
EIF(F=1ORF=2)AND(HS=0ORHS=3) E=1
 1440 ENDPROC
 1450 DEFPROCRULE2
 1460 E=0:IFINT(G/2)<>INT(HD/2) E=1
 1470 ENDPROC
 1480 DEFPROCRULE3
 1490 E=0:FORA=0TO12:IFJ1(A)=G A=12:NEXT ELSE
NEXT:GOTO1520
 1500 FORA=0TO12:IFJ2(A)=HD E=1
 1510   NEXT:GOTO1550
 1520 FORA=0TO12:IFJ2(A)=G A=12
 1530   NEXT:FORA=0TO12:IFJ1(A)=HD E=1
 1540   NEXT
 1550 ENDPROC
 1560 DEFPROCRULE4
 1570 IF(F=3ANDHS=1)OR(F=1ANDHS=0)OR(F=0ANDHS
=2)OR(F=2ANDHS=3) E=1 ELSEE=0
 1580 ENDPROC
 1590 DEFPROCRULE5
 1600 IFCT=51 E=1:ENDPROC
 1610 L=1:IFHD=(L ORHD) LC=0 ELSELC=1
 1620 IFG=(L ORG) LF=0 ELSELF=1
 1630 IFG0=(G0 ORL)L0=0 ELSEL0=1
 1640 IF(LC=1 ANDLF=0 ANDL0=0)OR(LC=1 ANDLF=1
 ANDL0=0)OR(LC=0 ANDLF=1 ANDL0=1)OR(LC=0 ANDL
F=0 ANDL0=1) E=1 ELSEE=0
 1650 ENDPROC
```

```
 166Ø DEFPROCRULE6
 167Ø IFG=13 GT=Ø ELSEGT=G
 168Ø E=Ø:FORA=ØTO12:J1(A)=Ø:NEXT:FORA=ØTO2:J
1(A)=GT+1:GT=GT+1:IFGT=13 GT=Ø
 169Ø   NEXT:FORA=ØTO2:IFJ1(A)=HD E=1:A=2
 17ØØ   NEXT:ENDPROC
 171Ø DEFPROCRULE7
 172Ø IFG=1 GT=14 ELSEGT=G
 173Ø E=Ø:FORA=ØTO12:J1(A)=Ø:NEXT:FORA=ØTO2:J
1(A)=GT-1:GT=GT-1:IFGT=1 GT=14
 174Ø   NEXT:FORA=ØTO2:IFJ1(A)=HD E=1:A=2
 175Ø   NEXT:ENDPROC
 176Ø DEFPROCRULE8
 177Ø E=Ø:IFHS=Ø ORHS=3 E=1
 178Ø IFJV(3)=1 E=1
 179Ø ENDPROC
 18ØØ DEFPROCRULE9
 181Ø E=Ø:IF(F=1 ANDHS=3) E=1
 182Ø IF(F=3 ANDHS=Ø) E=1
 183Ø IF(F=Ø ANDHS=2) E=1
 184Ø IF(F=2 ANDHS=1) E=1
 185Ø ENDPROC
 186Ø DEFPROCRULE1Ø
 187Ø E=Ø:FORA=ØTO12:J1(A)=Ø:NEXT:FORA=ØTO2:J
1(A)=A+1:NEXT:J1(3)=5:J1(4)=7:J1(5)=11:J1(6)=
13
 188Ø FORA=ØTO6:IFHD=J1(A) E=1
 189Ø   NEXT:ENDPROC
 19ØØ DEFPROCRULE11
 191Ø E=Ø:IF(G=6 ORG=1Ø)ANDHS=2 E=1
 192Ø IF(G=5 ORG=9 ORG=13)ANDHS=1 E=1
 193Ø IF(G=7 ORG=11)ANDHS=3 E=1
 194Ø IF(G<5 ORG=8 ORG=12)ANDHS=Ø E=1
 195Ø ENDPROC
 196Ø DEFPROCRULE12
 197Ø E=Ø:L=1:IFG=(G ORL) L5=1 ELSEL5=Ø
 198Ø IF(HS=1 ORHS=2)ANDL5=Ø E=1
 199Ø IF(HS=Ø ORHS=3)ANDL5=1 E=1
 2ØØØ ENDPROC
 2Ø1Ø DEFPROCRULE13
 2Ø2Ø E=Ø:FORA=ØTO12:J1(A)=Ø:NEXT:L1=G+4:IFL1
>13 L1=L1-13
```

```
 2030 FORA=0TO2:J1(A)=L1:L1=L1+1:IFL1>13 L1=1
 2040   NEXT:FORA=0TO2:IFHD=J1(A) E=1
 2050   NEXT:ENDPROC
 2060 DEFPROCRULE14
 2070 E=0:FORA=0TO12:J1(A)=0:NEXT:L1=G-8:IFL1
<1 L1=L1+13
 2080 FORA=0TO2:J1(A)=L1:L1=L1-1:IFL1<1 L1=13
 2090   NEXT:FORA=0TO2:IFHD=J1(A) E=1
 2100   NEXT:ENDPROC
 2110 DEFPROCRULE15
 2120 E=0:IF(HS=F)OR(HD=G) E=1
 2130 ENDPROC
 2140 DEFPROCRULE16
 2150 E=0:L=1:IFHD=(L ORHD) L=0
 2160 IF(L=0 ANDF=L8)OR(L=0 ANDF=L9) E=1
 2170 IF(L=1 ANDF=L6)OR(L=1 ANDF=L7) E=1
 2180 ENDPROC
 2190 DEFPROCRULE17
 2200 E=0:IF(HS=0 ORHS=3)ANDJV(3)=1 E=1
 2210 IF(HS=1 ORHS=2)ANDJV(3)=2 E=1
 2220 ENDPROC
 2230 DEFPROCRULE18
 2240 E=0:FORA=0TO12:J1(A)=0:NEXT:L1=G-2:FORA
=0TO4:IFL1<1 L1=L1+13
 2250   J1(A)=L1:L1=L1+1:IFL1>13 L1=1
 2260   NEXT:FORA=0TO4:IFHD=J1(A) E=1
 2270   NEXT:ENDPROC
 2280 DEFPROCRULE19
 2290 E=0:IFCT>50 E=1
 2300 L=1:IF(F=0 ORF=3) AND(F0=0 ORF0=3) ANDH
D<>(L ORHD) E=1
 2310 IF(F=1 ORF=2) AND(F0=1 ORF0=2) ANDHD<>(
L ORHD) E=1
 2320 IF(F=0 ORF=3) AND(F0=1 ORF0=2) ANDHD=(L
 ORHD) E=1
 2330 IF(F=1 ORF=2) AND(F0=0 ORF0=3) ANDHD=(L
 ORHD) E=1
 2340 ENDPROC
 2350 DEFPROCRULE20
 2360 E=0:IFCT>49 E=1
 2370 IF(F=0 ORF=3) AND(HD<8) E=1
 2380 IF(F=1 ORF=2) AND(HD>7) E=1
```

```
 2390 ENDPROC
 2400 DEFPROCRULE21
 2410 E=0:IFCT>48 E=1
 2420 L=1:IFHD=(L ORHD)ANDJV(0)=1 E=1
 2430 IFHD<>(L ORHD)ANDJV(0)=2 E=1
 2440 ENDPROC
 2450 DEFPROCRULE22
 2460 E=1:IFCT>49 ENDPROC
 2470 DL=C5/3:IFDL<>INT(DL) ENDPROC
 2480 IFHS=0 ANDF1=1 ENDPROC
 2490 IFHS=3 ANDF1=0 ENDPROC
 2500 IFHS=2 ANDF1=3 ENDPROC
 2510 IFHS=1 ANDF1=2 ENDPROC
 2520 E=0:ENDPROC
 2530 DEFPROCRULE23
 2540 E=0:IFHD>10 ANDG<11 E=1
 2550 IFHD<11 ANDG>10 E=1
 2560 ENDPROC
 2570 DEFPROCRULE24
 2580 E=0:IFC6>=C5 AND(HS=1 ORHS=2) E=1
 2590 IFC6<C5 AND(HS=0 ORHS=3) E=1
 2600 ENDPROC
 2610 DEFPROCLAYOUT
 2620 PRINTCHR$129Y1$''''CHR$131Y2$''''CHR$1
32''''" "S$(0)'" "S$(1)'" "S$(2)'" "S$(3):ENDP
ROC
 2630 DEFPROCCARDSLEFT
 2640 FORP=12TO15:FORA=1TO13:PRINTTAB(8+A*2,P
);P$(A,P-12)
 2650      NEXT,:ENDPROC
 2660    DEFPROCRIGHT
 2670    PROCSHOWCARD
 2680    F3=F2:G3=G2:F2=F1:G2=G1:F1=F0:G1=G0:F
0=F:G0=G:F=HS:G=HD
 2690    JT=2:C5=C5+1:IFCT<>52 K=K+1
 2700    IFK>1 PRINTTAB(10,0);K+1;" in a row s
o far"
 2710    IFCT=52 THEN2770
 2720    FORI=129TO135
 2730      PRINTTAB(0,I-112);CHR$I;"Correct!!"
 2740      SOUND17,-15,200,5
 2750      J=INKEY(50)
```

```
 276Ø      NEXT
 277Ø    J=INKEY(2ØØ)
 278Ø    A$=R$(HD):IFLEFT$(A$,1)=" " A$=RIGHT$
(A$,LENA$-1)
 279Ø    IFRIGHT$(A$,1)<>" " A$=A$+" "
 28ØØ    A$="c"+STR$(53-CT)+"-"+A$+"of "+S$(HS
)
 281Ø    PRINTTAB(C1*2Ø MOD4Ø,INT(C1/2)+1);A$
 282Ø    C1=C1+1
 283Ø    ENDPROC
 284Ø    DEFPROCWRONG
 285Ø    PROCSHOWCARD
 286Ø    JT=1:C6=C6+1:SOUND1,-15,4Ø,1ØØ
 287Ø    VDU28,Ø,1Ø,39,6
 288Ø    A$=R$(HD):IFLEFT$(A$,1)=" " A$=RIGHT$
(A$,LENA$-1)
 289Ø    IFRIGHT$(A$,1)<>" " A$=A$+" "
 29ØØ    A$="c"+STR$(53-CT)+"-"+A$+"of "+S$(HS
)
 291Ø    PRINTTAB(M1*2Ø MOD4Ø,INT(M1/2));A$;
 292Ø    M1=M1+1:IFM1>12 M1=12
 293Ø    VDU26:J=INKEY(2ØØ):SOUND17,Ø,Ø,1:ENDP
ROC
 294Ø    DEFPROCINIT
 295Ø    F=Ø:FØ=Ø:F1=Ø:F2=Ø:F3=Ø
 296Ø    G=Ø:GØ=Ø:G1=Ø:G2=Ø:G3=Ø
 297Ø    C1=Ø:M1=Ø
 298Ø    NL$=CHR$13+CHR$1Ø
 299Ø    CT=52:FORA=ØTO3:P$(1,A)=" A":P$(1Ø,A)
=" T":P$(11,A)=" J":P$(12,A)=" Q":P$(13,A)="
K":FORB=2TO9:P$(B,A)=" "+STR$B:NEXT,
 3ØØØ      FORA=ØTO3:FORB=1TO13:H(B,A)=B:NEXT,
 3Ø1Ø        Y1$="CORRECT":Y2$="MISTAKES":Y3$=
"correct"
 3Ø2Ø        X1$="black":X2$="red":X3$="card":
X4$="even":X5$="odd":X6$="denomination":X7$="
less than":X8$="greater than":V1$=" last corr
ect card":V2$=CHR$8+V1$+" but"
 3Ø3Ø        V3$="suit":V4$="same as":V5$="dif
ferent from":V6$="total of ":V7$="and vice ve
rsa":V8$="1 to 3":U1$="BASIC":U2$="INTERMEDIA
TE":U3$="ADVANCED"
```

```
 3040        XA$="colour":XB$="it follows the
sequence":XC$="consistant with ":XD$="sequenc
e of: ":XE$="2 odd alternating with 2 even":X
F$="previous play":XG$="mistake":XH$="prime n
umber"
 3050        FORI=1TO13:READD$(I):NEXT
 3060        FORI=1TO13:READR$(I):NEXT
 3070        FORI=0TO3:READS$(I):NEXT
 3080        DATA" ACE "," TWO ","THREE","FOUR
 ","FIVE "," SIX ","SEVEN","EIGHT","NINE ","
TEN ","JACK ","QUEEN","KING "
 3090        DATAAce,Two,Three,Four,Five,Six,S
even,Eight,Nine,Ten,Jack,Queen,King
 3100        DATASpades,Hearts,Diamonds,Clubs
 3110        ENDPROC
 3120        DEFPROCGETKEY
 3130        *FX15,1
 3140        PRINTTAB(20,17);SPC255
 3150        PRINTTAB(0,17);CHR$133;"Use the c
ursor keys to choose the suit"';CHR$133;"and
press RETURN,or 'Z' to end";:L=12
 3160        PRINTTAB(0,L);CHR$130;CHR$8;
 3170        Z=GET:IFZ=138 ANDL<15 L=L+1:VDU13
5
 3180        IFZ=139 ANDL>12 L=L-1:VDU135
 3190        IFZ=90 ENDPROC
 3200        IFZ<>13 THEN3160
 3210        HS=L-12:FL=0:FORI=1TO13:IFH(I,HS)
<>0 FL=1
 3220          NEXT:IFFL=0 PROCCANT:GOTO3160
 3230        PRINTTAB(0,17);CHR$133;"Now use t
he cursor keys to choose the "';CHR$133;"deno
mination and press RETURN,or Z to"';CHR$133;"
change your choice of suit";:A=1
 3240        PRINTTAB(8+A*2,L);CHR$134;CHR$9;C
HR$135;CHR$8;CHR$8;
 3250        Z=GET:IFZ=136 ANDA>1 A=A-1:VDU8,1
35
 3260        IFZ=137 ANDA<13 A=A+1:VDU8,135
 3270        IFZ=90 PRINTTAB(0,L);CHR$135;TAB(
8+A*2,L);" ";TAB(10+A*2,L);" ":GOTO3150
 3280        IFZ<>13 THEN3240
```

```
 3290        HD=A:IFH(HD,HS)=Ø PROCCANT:GOTO32
5Ø
 3300        H(HD,HS)=Ø:P$(HD,HS)="   ":PRINTTA
B(Ø,17)SPC12Ø:ENDPROC
 3310        DEFPROCMAKECARD(A,B)
 3320        B$=RIGHT$(P$(A,B),1):H$=STRING$(8
,CHR$8)+CHR$1Ø
 3330        IFB=Ø A$=CHR$148:D$="   _   ":E$="
h"+CHR$255+CHR$253+" ":F$=" *o/ ":G$="  .$ "
 3340        IFB=1 A$=CHR$145:D$="       ":E$="
"+CHR$254+CHR$253+CHR$255+"4":F$=" k"+CHR$255
+CHR$255+"!":G$="   +!  "
 3350        IFB=2 A$=CHR$145:D$="   _   ":E$="
x"+CHR$255+CHR$253+"Ø":F$=" "+CHR$34+"o' ":G$
="       "
 3360        IFB=3 A$=CHR$148:D$="   _   ":E$="
 o% ":F$=" "+CHR$255+"nn5":G$="  .$ "
 3370        C$=A$+CHR$1Ø4+CHR$124+CHR$124+CHR
$124+CHR$124+CHR$124+CHR$52+H$
 3380        C$=C$+A$+CHR$1Ø6+D$+CHR$53+H$
 3390        C$=C$+A$+CHR$1Ø6+E$+CHR$53+H$
 3400        C$=C$+A$+CHR$1Ø6+F$+CHR$53+H$
 3410        C$=C$+A$+CHR$1Ø6+G$+CHR$53+H$
 3420        C$=C$+A$+CHR$1Ø6+D$(A)+CHR$53+H$
 3430        C$=C$+A$+CHR$42+CHR$47+CHR$47+CHR
$47+CHR$47+CHR$47+CHR$37
 3440        ENDPROC
 3450        DEFPROCSHOWCARD
 3460        PRINTTAB(2Ø,17)SPC255
 3470        PROCMAKECARD(HD,HS)
 3480        PRINTTAB(2Ø,17);C$
 3490        ENDPROC
 3500        DEFPROCCANT
 3510        SOUND1,-15,4Ø,255
 3520        PRINTTAB(Ø,22);CHR$134;"You can't
 choose that":Z$=INKEY$(3ØØ):SOUND17,Ø,Ø,1
 3530        PRINTTAB(Ø,22);CHR$152;SPC4Ø:ENDP
ROC
 3540        DEFPROCSPLIT(A$)
 3550        IFASCA$>96 A$=CHR$(ASC(LEFT$(A$,1
))-32)+RIGHT$(A$,LENA$-1)
 3560        A$=A$+" ":REPEAT:L=INSTR(A$," "):
```

```
IFPOS+L>38 PRINT
 357Ø          PRINTLEFT$(A$,L);:A$=MID$(A$,L+
1):UNTILLENA$<2
 358Ø        PRINT:ENDPROC
 359Ø        END
 36ØØ        DEFPROCGIVEUP
 361Ø        PRINTTAB(Ø,17)SPC8Ø:*FX15,1
 362Ø        PRINTTAB(Ø,17);CHR$134;"Are you s
ure (Y/N)";
 363Ø        Z=GET:ENDPROC
 364Ø        DEFPROCSHOWRULE
 365Ø        PRINTTAB(Ø,17)SPC2ØØ;SPC8Ø;
 366Ø        PRINTTAB(Ø,16):PROCSPLIT(S1$+" is
 correct only if "+S2$+" is "+S3$+S4$+S5$)
 367Ø        IFS6$<>"" PRINT:PROCSPLIT("and "+
S6$+" is correct only if "+S7$+" is "+S8$+S9$
)
 368Ø        ENDPROC
 369Ø        DEFPROCWIN
 37ØØ        PRINTTAB(Ø,17)SPC2ØØ;SPC8Ø;
 371Ø        PRINTTAB(Ø,17)CHR$141;CHR$129;"Yo
u've done it!"
 372Ø        PRINTTAB(Ø,18)CHR$141;CHR$129;"Yo
u've done it!"
 373Ø        FORI=16ØTO23ØSTEP2:SOUND&2Ø1,-15,
I,2:SOUND&2Ø2,-15,I+2,2:SOUND&2Ø3,-15,I+4,2:N
EXT
 374Ø        J=INKEY(3ØØ)
 375Ø        PROCEND
 376Ø        DEFPROCLOSE
 377Ø        PRINTTAB(Ø,17)SPC2ØØ;SPC8Ø;
 378Ø        PRINTTAB(Ø,17)CHR$141;CHR$129;"Al
l the cards have gone,you lose!"
 379Ø        PRINTTAB(Ø,18)CHR$141;CHR$129;"Al
l the cards have gone,you lose!"
 38ØØ        FORI=15ØTO8ØSTEP-2:SOUND&2Ø1,-15,
I,2:SOUND&2Ø2,-15,I+2,2:SOUND&2Ø3,-15,I+4,2:N
EXT
 381Ø        J=INKEY(3ØØ)
 382Ø        PROCEND
 383Ø        DEFPROCEND
 384Ø        PRINTTAB(13,24);CHR$134;"Press a
```

```
key";:*FX15,1
 3850        Z=GET
 3860        PRINTTAB(0,17);SPC200;SPC119;
 3870        PRINTTAB(0,17);"Do you want to pl
ay again (Y/N) ?";
 3880        Z=GET
 3890        IFZ=89 THEN60
 3900        IFZ<>78 THEN3880
 3910        END
 3920        DEFPROCINTRO
 3930        CLS:PRINT'''"T R U T H"
 3940        PRINT''"The idea of this game is
to select eightcards in a row,that all follow
 the same basic rule"
 3950        PRINT''"For example,if the rule w
as:-"
 3960        PRINT"Play black and red alternat
ly"
 3970        PRINT'"Therefore:-"
 3980        PRINT"Ace of Spades,Four of Heart
s"
 3990        PRINT"Two of Clubs ,Six of Diamon
ds"
 4000        PRINT"King of Clubs,Three of Hear
ts"
 4010        PRINT"Ten of Spades,Queen of Diam
onds"
 4020        PRINT'"Would win,although they ar
e not all thateasy"
 4030        PRINT'"Press a key to play";:*FX1
5,1
 4040        Z=GET:ENDPROC
```

PAUL BORRETT
Original program by D. Denholm

GENERAL DESCRIPTION

This challenging game is based on a car race. You pilot a car round a five-lane track, trying to hit as many dots as possible. However, to hinder you a mad driver is travelling in the opposite direction trying to crash his car into yours!

Try and cover as many dots as you can before he hits you. Controls are by two keys: Z to move in a lane, and M to move out a lane, and you are in the car travelling anti-clockwise.

If you manage to eradicate all the dots successfully a new track will appear.

DETAILED DESCRIPTION

Lines 10-290 main body of program from which the procedures are controlled.
300-440 PROC UDG - defines graphics used.
450-630 PROC SCREEN - outlines colour and on-screen display, and sound.
640-730 PROC COM - defines maniac's car and position.
740-830 PROC HUM - defines player's car and position.

840-880 FN CH(X)] - defines functions used in
890-910 FN A(X, Y)] movement of cars.
920-960 PROC D] - deals with movement of cars round
970-1010 PROC R] track.
1020-1100 PROC L (S)] - deals with further movement of
1110-1250 PROC M (S)] cars, including changing lanes.
1260-1320 gives chance of another go if RETURN is pressed.
1330-1410 PROC INTRO - displays introduction.

```
LIST
    10 *TV255,1
    20 MODE1:VDU23,1,0;0;0;0;:REM FOR O.S
. VERSION 0.1 VDU23;8202;0;0;0;
    30 PROCINTRO
    40 PROCUDG:HI=0
    50 VDU19,2,2,0,0,0,19,3,5,0,0,0
    60 SC=0:F=0
    70 PROCSCREEN
    80 XA=0:XD=0:YA=0:YD=0
    90 PROCCOM:PROCHUM
   100 XA=CA:XD=CD
   110 IFCD>13ANDCD<18 PROCL(1):GOTO140
   120 IFCA>16ANDCA<23 PROCL(2):GOTO140
   130 F=1
   140 CA=CA+C1:CD=CD+C2
   150 Z%=FNA(CA,CD):IFZ%=3 CA=CA-C1:CD=C
D-C2:PROCD
   160 IFZ%=1 DT=DT-1:SOUND1,-15,50,1:IFD
T=0 PROCSCREEN
   170 IFZ%=2 THEN1260
   180 PROCCOM
   190 YA=HA:YD=HD
   200 IFHD>13ANDHD<18 PROCM(1)
   210 IFHA>16ANDHA<23 PROCM(2)
   220 HA=HA+H1:HD=HD+H2
   230 Z%=FNA(HA,HD):IFZ%=3 HA=HA-H1:HD=H
D-H2:PROCR
   240 IFZ%=1 DT=DT-1:SC=SC+10:SOUND1,-15
,200,1:IFDT=0 PROCSCREEN
   250 IFZ%=2 THEN1260
   260 PROCHUM
```

```
  270 PRINTTAB(13,15);SC
  280 GOTO100
  290 END
  300 DEFPROCUDG
  310 VDU23,224,0,0,0,0,24,24,0,0
  320 VDU23,225,0,238,68,255,255,68,238,
0
  330 VDU23,226,24,90,126,90,24,90,126,9
0
  340 VDU23,227,0,119,34,255,255,34,119,
0
  350 VDU23,228,90,126,90,24,90,126,90,2
4
  360 VDU23,229,60,60,102,102,110,110,12
6,126,23,230,118,118,102,102,60,60,0,0
  370 VDU23,231,24,24,56,56,24,24,24,24,
23,232,24,24,24,24,126,126,0,0
  380 VDU23,233,60,60,102,102,6,6,12,12,
23,234,24,24,48,48,126,126,0,0
  390 VDU23,235,60,60,102,102,6,6,28,28,
23,236,6,6,102,102,60,60,0,0
  400 VDU23,237,12,12,28,28,60,60,108,10
8,23,238,126,126,12,12,12,12,0,0
  410 VDU23,239,126,126,96,96,124,124,6,
6,23,240,6,6,102,102,60,60,0,0
  420 VDU23,241,85,170,149,140,248,24,25
5,241,23,242,192,170,163,23,255,23,255,1
49
  430 ENVELOPE1,5,16,12,8,1,1,1,100,-1,0
,-10,120,0
  440 ENDPROC
  450 DEFPROCSCREEN
  460 DT=218:CA=16:CD=27:HA=23:HD=27:C1=
-1:H1=1:C2=0:H2=0:COLOUR1
  470 CLS:A$=STRING$(8,CHR$224+" ")+"
 "+STRING$(8," "+CHR$224)
  480 MOVE48,100:DRAW48,900:DRAW1232,900
:DRAW1232,100:DRAW48,100
  490 FORI=4TO12STEP2:PRINTTAB(2,I);A$;T
AB(2,I+15);A$:NEXT
  500 FORI=1TO5:PRINTTAB(2,I*2+3);STRING
$(I,CHR$224+CHR$9);TAB(2,28-I*2);STRING$
```

```
(I,CHR$224+CHR$9);TAB(39-I*2,I*2+3);STRI
NG$(I,CHR$224+CHR$9);TAB(39-I*2,28-I*2);
STRING$(I,CHR$224+CHR$9):NEXT
  510 COLOUR3:FORI%=0TO3
  520    MOVE112+I%*64,440:DRAW112+I%*64,
172+I%*64:DRAW540,172+I%*64
  530    MOVE736,844-I%*64:DRAW1168-I%*64
,844-I%*64:DRAW1168-I%*64,576
  540    MOVE112+I%*64,576:DRAW112+I%*64,
844-I%*64:DRAW540,844-I%*64
  550    MOVE736,172+I%*64:DRAW1168-I%*64
,172+I%*64:DRAW1168-I%*64,440
  560    NEXT
  570 MOVE368,428:DRAW368,588:DRAW912,58
8:DRAW912,428:DRAW368,428
  580 PRINTTAB(13,14);"Score    High";TA
B(13,15);SC;TAB(22,15);HI;TAB(13,17);"Z-
In      M-Out"
  590 SOUND1,1,200,120
  600 TIME=-100:REPEAT:T%=5-(TIME/100)
  610    VDU31,19,16,T%*2+229,31,19,17,T%
*2+230
  620    UNTILTIME>500:VDU31,19,16,32,31,
19,17,32
  630 ENDPROC
  640 DEFPROCCOM
  650 COLOUR2
  660 IFC1=-1 Z=227
  670 IFC1=1 Z=225
  680 IFC2=-1 Z=226
  690 IFC2=1 Z=228
  700 VDU31,CA,CD,Z
  710 VDU31,XA,XD,32
  720 COLOUR3
  730 ENDPROC
  740 DEFPROCHUM
  750 COLOUR2
  760 IFH1=-1 Z=227
  770 IFH1=1 Z=225
  780 IFH2=-1 Z=226
  790 IFH2=1 Z=228
  800 VDU31,HA,HD,Z
```

```
  810 VDU31,YA,YD,32
  820 COLOUR3
  830 ENDPROC
  840 DEFFNCH(X)
  850 Y=X DIV40:X=X MOD40
  860 VDU31,X,Y:A%=135:C=USR(&FFF4)AND&F
FFF:C=C DIV256
  870 IFC>127 ANDC<224 C=C+96
  880 =C
  890 DEFFNA(X,Y)
  900 X=32*X+16:Y=32*(32-Y)-20:MOVEX,Y
  910 =POINT(X,Y)
  920 DEFPROCD
  930 IFC1=-1 C1=0:C2=-1:ENDPROC
  940 IFC1=1 C1=0:C2=1:ENDPROC
  950 IFC2=1 C1=-1:C2=0:ENDPROC
  960 IFC2=-1 C1=1:C2=0:ENDPROC
  970 DEFPROCR
  980 IFH1=1 H1=0:H2=-1:ENDPROC
  990 IFH1=-1 H1=0:H2=1:ENDPROC
 1000 IFH2=-1 H1=-1:H2=0:ENDPROC
 1010 IFH2=1 H1=1:H2=0:ENDPROC
 1020 DEFPROCL(S)
 1030 IFF=0 ENDPROC
 1040 IFRND(2)=1 ENDPROC
 1050 F=0:Z=RND(3)-2:IFZ=0 ENDPROC
 1060 IFS=1THEN1090
 1070 IFFNA(CA,CD+Z)=0 CD=CD+Z+Z
 1080 ENDPROC
 1090 IFFNA(CA+Z,CD)=0 CA=CA+Z+Z
 1100 ENDPROC
 1110 DEFPROCM(S)
 1120 K=0:IFINKEY(-98)K=1 ELSEIFINKEY(-1
02)K=2
 1130 IA=0:ID=0:OA=0:OD=0
 1140 IFS=2THEN1180
 1150 IFHA<20 IA=-(K=1):OA=(K=2)
 1160 IFHA>20 IA=(K=1):OA=-(K=2)
 1170 GOTO1200
 1180 IFHD>15 ID=(K=1):OD=-(K=2)
 1190 IFHD<15 ID=-(K=1):OD=(K=2)
 1200 X=HA+IA+OA+IA+OA:Y=HD+ID+OD+ID+OD
```

```
 1210 IFY<3ORY>27ORX<2ORX>38 ENDPROC
 1220 IFX>11 ANDX<28 ANDY>13 ANDY<18 END
PROC
 1230 IFFNA(X,Y)<>0 THENENDPROC
 1240 HA=X:HD=Y
 1250 ENDPROC
 1260 FORI=1TO40:VDU31,HA,HD,241,31,HA,H
D,42,31,HA,HD,242
 1270   SOUND0,-15,4,1:NEXT
 1280 IFSC>HI HI=SC:PRINTTAB(22,15);HI
 1290 PRINTTAB(13,17);"Press   Return"
 1300 REPEAT:UNTILGET$=CHR$13
 1310 GOTO60
 1320 ENDPROC
 1330 DEFPROCINTRO
 1340 CLS:PRINT''"Race Chase"'"_________
_"
 1350 PRINT''"  In this game,you are in
control of    a car that is hurtling aro
und a         track."
 1360 PRINT''"  Unfortunately,there is a
 maniac in     another car driving in th
e opposite     direction,trying to kill
you."
 1370 PRINT''"  You might care to hit as
 many dots as possible before the inevit
able happens, after all,you never know w
hat they mighttake into account when you
 get to       Heaven."
 1380 PRINT''"    Key 'Z' will move you
in one lane"
 1390 PRINT'"    Key 'M' will move you o
ut one lane."
 1400 PRINT'''"      Press a key to play
";:A$=GET$
 1410 ENDPROC
```

BEEBTREK

PAUL BORRETT
Original author unknown

GENERAL DESCRIPTION

This game is based on that evergreen television series 'Startrek'. You are at the helm of the Enterprise charged with the arduous task of ridding the galaxy of the Klingon menace. Your arsenal consists of phasers and photon torpedoes, but it is not a one-way fight as the Klingons can have a nasty bite as well.

After the instructions have been displayed, you are invited to choose a proficiency level and your mission objectives are then presented. Choose the level well since this determines the enemy strength and how much time the mission is to last. When the Enterprise's energy gets low, you can replenish both that and its complement of torpedoes by finding a starbase and docking with it. However, on the higher skill levels these are few and far between, and traversing the galaxy efficiently becomes important if the deadline is to be beaten.

DETAILED DESCRIPTION

Arrays: K – contains position and energy levels of the Klingons in the current sector.
G – number of stars, Klingons and starbases for each sector in the galaxy.
S – contents of the current sector.

Lines 10-30 disable cursor editing keys and set mode.
40-90 calls to the procedures that give instructions and set up variables.
100 remove cursor.
110-180 main game loop repeated until time runs out or all the baddies are destroyed.

190-250 introductory title.
260-410 sets up global variables and the special characters to represent the ships.
420-740 two-screen display of instructions.
750-920 ask for the skill level.
930-1070 fill the galaxy with objects randomly.
1080-1290 display current sector graphically along with data and energy level readouts.
1300-1630 get a command line from the keyboard and call the appropriate procedure.
1640-1750 commiserate with player if either time allotted or energy has run out.
1760-1790 pause routine to allow reading at human speed.
1800-2040 randomly allocate co-ordinates for the objects in the current sector.
2050-2100 move the Enterprise around the current sector.
2110-2140 moving uses up energy - this calculates it.
2150-2180 restore energy/torpedoes when docked.
2190 calculates distance between you and a Klingon.
2200-2400 long range sensor scan.
2410-2500 does the Klingon fire back?
2510-2570 how much energy to divert to the shields?
2580-2650 general numeric input procedure. Reads up to a six digit number ignoring non-numerics.
2660-2810 fires phasers and checks if Klingon is dead.
2820-3040 fires a photon torpedo in the direction specified by the arrow keys.
3050-3130 Success! Displays a congratulatory message and offers another game.

```
  10 *TV255,1
  20 *FX4,1
  30 MODE7
  40 PROCTitle
  50 PROCInit
  60 PROCIntro
  70 PROCSkill
  80 PROCSetup
  90 MODE5:VDU19,2,2,0,0,0
 100 VDU23,1,0;0;0;0;:REM FOR O.S. 0.1  VDU2
```

```
3;8202;0;0;0
  110 REPEAT
  120   PROCKeys
  130   *FX15,1
  140   PROCDisplay
  150   PROCEnergyloss
  160   PROCAliens
  170   PROCDead
  180   UNTILFALSE
  190 DEFPROCTitle
  200 CLS:PRINT''''CHR$141;CHR$129;"Beebtrek"
  210 PRINTCHR$141;CHR$129;"Beebtrek"
  220 PRINT''''CHR$130;"or,How to master the g
alaxy in one"
  230 PRINTCHR$130;"easy lesson."
  240 Z$=INKEY$(500)
  250 ENDPROC
  260 DEFPROCInit
  270 Energy=50000:Shields=0:K=0:Torps=10
  280 Stardate=RND(50)*100+1000
  290 GX=RND(8):GY=RND(8)
  300 SX=RND(8):SY=RND(8)
  310 VDU23,224,0,0,219,90,126,90,219,0
  320 VDU23,225,0,112,248,32,47,40,255,0
  330 VDU23,226,24,24,24,231,231,24,24,24
  340 VDU23,227,0,112,248,32,47,40,255,0
  350 K$=CHR$17+CHR$1+CHR$224+CHR$17+CHR$3
  360 E$=CHR$17+CHR$2+CHR$225+CHR$17+CHR$3
  370 B$=CHR$17+CHR$2+CHR$226+CHR$17+CHR$3
  380 T$=CHR$17+CHR$1+"+"+CHR$17+CHR$3
  390 S$="*"
  400 NS=0
  410 ENDPROC
  420 DEFPROCIntro
  430 CLS:PRINT'"Beebtrek"
  440 PRINT'"In this game you have to patrol
the"
  450 PRINT"galaxy,killing klingons,in order
to"
  460 PRINT"make space a safe place for decen
t"
  470 PRINT"earthlings."
```

```
  480 PRINT''"You move around by use of the c
ursor"
  490 PRINT"keys,firing phasers by pressing '
P'"
  500 PRINT"when you will be asked to input a
ny"
  510 PRINT"amount of energy,up to your prese
nt"
  520 PRINT"amount."
  530 PRINT'"And firing the photon torpedoes
by"
  540 PRINT"pressing 'T' when you will be ask
ed"
  550 PRINT"to push a cursor key,to indicate
the"
  560 PRINT"direction."
  570 PROCReady
  580 CLS:PRINT'"Beebtrek"
  590 PRINT'"If you need more energy,move you
r ship"
  600 PRINT"so it is next to a starbase,upon
which"
  610 PRINT"you will get a top up of energy."
  620 PRINT"Press 'S' to charge Shields up to
 a"
  630 PRINT"maximum of 10000 units."
  640 PRINT'"When you hit,or are being hit by
 a"
  650 PRINT"Klingon,the amount of damage depe
nds"
  660 PRINT"on how close you are to each othe
r."
  670 PRINT'"Check the long range scan with '
L'"
  680 PRINT"You will then be shown a three fi
gure"
  690 PRINT"number for the sectors above,belo
w,to"
  700 PRINT"the left and right of your presen
t"
  710 PRINT"sector.The number 251 would mean"
```

```
  720 PRINT"2 klingons,5 stars,and 1 Starbase
."
  730 PROCReady
  740 ENDPROC
  750 DEFPROCSkill
  760 CLS:PRINT''"Which skill level do you wa
nt:-"
  770 PRINT'"1. Cadet"
  780 PRINT'"2. Corporal"
  790 PRINT'"3. Major"
  800 PRINT'"4. Admiral"
  810 PRINT'"5. Air-commodore"
  820 PRINT''"1-5 ?";
  830 Z$=GET$:IFZ$<"1" ORZ$>"5" THEN830
  840 Klingons=INT(5+VAL(Z$)*(1+RND(2)*3))
  850 Starbases=8-VAL(Z$)
  860 Enddate=Stardate+VAL(Z$)*600
  870 PRINT''"O.K. You have to destroy ";Klin
gons;" Klingons"
  880 PRINT'"In ";Enddate-Stardate;" Stardate
s"
  890 PRINT'"With ";Starbases;" Starbases"
  900 PRINT'"Press a key to play";:*FX15,1
  910 Z$=GET$
  920 ENDPROC
  930 DEFPROCSetup
  940 DIMK(10,3),G(8,8),S(8,8)
  950 FORI%=1TO100
  960   X=RND(8):Y=RND(8)
  970   G(X,Y)=G(X,Y)+10
  980   NEXT
  990 FORI%=1TOKlingons
 1000   X=RND(8):Y=RND(8)
 1010   G(X,Y)=G(X,Y)+100
 1020   NEXT
 1030 FORI%=1TOStarbases
 1040   X=RND(8):Y=RND(8)
 1050   G(X,Y)=G(X,Y)+1
 1060   NEXT
 1070 ENDPROC
 1080 DEFPROCDisplay
 1090 IFNS<>1 PROCFillsector:NS=1
```

```
 1100 MOVE376,896:DRAW896,896:DRAW896,632:DRA
W376,632:DRAW376,896
 1110 FORI=1TO8
 1120   FORJ=1TO8
 1130     IFS(I,J)=1 PRINTTAB(I+5,J+3);S$
 1140     IFS(I,J)=2 PRINTTAB(I+5,J+3);E$
 1150     IFS(I,J)=3 PRINTTAB(I+5,J+3);K$
 1160     IFS(I,J)=4 PRINTTAB(I+5,J+3);B$
 1170     IFS(I,J)=5 PRINTTAB(I+5,J+3);T$
 1180     NEXT,
 1190   COLOUR1
 1200   PRINTTAB(0,15);"Stardate....";Stardat
e;"   "
 1210   PRINTTAB(0,17);"Energy......";Energy;
"     "
 1220   PRINTTAB(0,19);"Shields.....";Shields
;"     "
 1230   PRINTTAB(0,21);"Klingons....";Klingon
s;" "
 1240   PRINTTAB(0,23);"Torpedoes...";Torps;"
 "
 1250   COLOUR2
 1260   PRINTTAB(0,25);"Sector......";GX;",";
GY
 1270   PRINTTAB(0,27);"Co-ords.....";SX;",";
SY
 1280   COLOUR3
 1290   ENDPROC
 1300   DEFPROCKeys
 1310   Z=INKEY(0)
 1320   IFZ<>136 THEN1390
 1330   IFSX>1THEN1360
 1340   IFGX=1 THENENDPROC
 1350   GX=GX-1:SX=8:NS=0:ENDPROC
 1360   IFS(SX-1,SY)=4 PROCBase
 1370   IFS(SX-1,SY)=0 THENPROCMoveship(-1,0)
 1380   ENDPROC
 1390   IFZ<>137 THEN1460
 1400   IFSX<8 THEN1430
 1410   IFGX=8 THENENDPROC
 1420   GX=GX+1:SX=1:NS=0:ENDPROC
 1430   IFS(SX+1,SY)=4 PROCBase
```

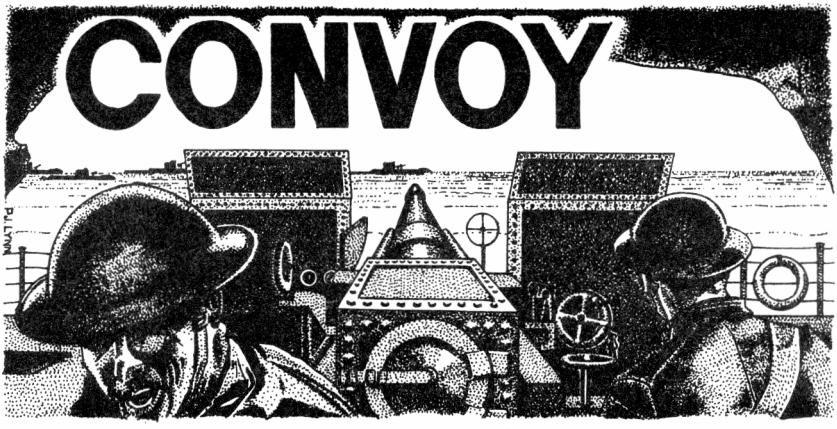
```
 1440   IFS(SX+1,SY)=0 THENPROCMoveship(1,0)
 1450   ENDPROC
 1460   IFZ<>138 THEN1520
 1470   IFSY<8 THEN1500
 1480   IFGY=8 THENENDPROC
 1490   GY=GY+1:SY=1:NS=0:ENDPROC
 1500   IFS(SX,SY+1)=4 PROCBase
 1510   IFS(SX,SY+1)=0 THENPROCMoveship(0,1)
 1520   IFZ<>139 THEN1580
 1530   IFSY>1 THEN1560
 1540   IFGY=1 THENENDPROC
 1550   GY=GY-1:SY=8:NS=0:ENDPROC
 1560   IFS(SX,SY-1)=4 PROCBase
 1570   IFS(SX,SY-1)=0 THENPROCMoveship(0,-1)
 1580   IFZ=83 PROCShields
 1590   IFZ=84 PROCTorp
 1600   IFZ=80 PROCPhasers
 1610   IFZ=76 PROCLong
 1620   IFKlingons=0 PROCCongrats
 1630   ENDPROC
 1640   DEFPROCDead
 1650   IFStardate>Enddate THEN1670
 1660   IFEnergy>0 ENDPROC
 1670   FORI=1TO3000:NEXT:CLS
 1680   COLOUR2:PRINT"Sorry!!!"
 1690   PRINT''"You didn't quite    manage it
."
 1700   PRINT''"You still had ";Klingons'"Kli
ngons left in"
 1710   PRINT"the end."
 1720   PRINT'''"Would you like"''"another try
 (Y/N)?";
 1730   Z$=GET$:IFZ$="Y" RUN
 1740   IFZ$<>"N"THEN1730
 1750   CLS:PRINT"Bye!":END
 1760   DEFPROCReady
 1770   PRINTTAB(7,22);"Press a key when read
y";:*FX15,1
 1780   Z$=GET$:CLS
 1790   ENDPROC
 1800   DEFPROCFillsector
 1810   IFNS=2 THEN1910
```

```
 1820   FORI=1TO8:FORJ=1TO8:S(I,J)=0:NEXT,
 1830     VDU28,6,11,13,4:CLS:VDU26:S(SX,SY)=
2
 1840     SOUND1,-15,200,10:SOUND2,-15,220,9
 1850     Z=(G(GX,GY)MOD10)
 1860     IFZ=0 THEN1910
 1870     FORI=1TOZ
 1880       X=RND(8):Y=RND(8)
 1890       IFS(X,Y)<>0 THEN1880
 1900       S(X,Y)=4:NEXT
 1910     Z=(G(GX,GY)DIV100)
 1920     K=0:IFZ=0 THEN1970
 1930     K=Z:FORI=1TOZ
 1940       X=RND(8):Y=RND(8)
 1950       IFS(X,Y)<>0 THEN1940
 1960       S(X,Y)=3:K(I,1)=X:K(I,2)=Y:K(I,3)
=1000:NEXT
 1970     IFNS=2 ENDPROC
 1980     Z=(G(GX,GY)MOD100DIV10)
 1990     IFZ=0 ENDPROC
 2000     FORI=1TOZ
 2010       X=RND(8):Y=RND(8)
 2020       IFS(X,Y)<>0 THEN2010
 2030       S(X,Y)=1:NEXT
 2040     ENDPROC
 2050     DEFPROCMoveship(X,Y)
 2060     SOUND1,-15,200,2
 2070     Energy=INT(Energy-Energy/150)
 2080     S(SX,SY)=0:PRINTTAB(SX+5,SY+3);" "
 2090     SX=SX+X:SY=SY+Y:PRINTTAB(SX+5,SY+3)
;E$
 2100     ENDPROC
 2110     DEFPROCEnergyloss
 2120     Energy=INT(Energy-Energy/250)
 2130     Stardate=Stardate+1
 2140     ENDPROC
 2150     DEFPROCBase
 2160     Energy=50000:Shields=0:Torps=10
 2170     SOUND&201,-15,100,30:SOUND&202,-15,
150,30:SOUND&203,-15,200,30
 2180     ENDPROC
 2190     DEFFNA(X)=SQR((SX-K(X,1))^2+(SY-K(X
```

```
,2))^2)
 2200      DEFPROCLong
 2210      Z$=STR$(G(GX-1,GY))
 2220      IFLENZ$<3 Z$="Ø"+Z$:GOTO2220
 2230      PRINTTAB(2,7);Z$
 2240      IFGX=8 Z$="ØØØ":GOTO2270
 2250      Z$=STR$(G(GX+1,GY))
 2260      IFLENZ$<3 Z$="Ø"+Z$:GOTO2260
 2270      PRINTTAB(15,7);Z$
 2280      Z$=STR$(G(GX,GY-1))
 2290      IFLENZ$<3 Z$="Ø"+Z$:GOTO2290
 2300      PRINTTAB(8,2);Z$
 2310      IFGY=8 Z$="ØØØ":GOTO2340
 2320      Z$=STR$(G(GX,GY+1))
 2330      IFLENZ$<3 Z$="Ø"+Z$:GOTO2330
 2340      PRINTTAB(8,13);Z$
 2350      PRINTTAB(3,Ø);"Press a key";:*FX15,
1
 2360      Z$=GET$
 2370      PRINTTAB(2,7);"    ";TAB(15,7);"    "
 2380      PRINTTAB(8,2);"    ";TAB(8,13);"    "
 2390      PRINTTAB(3,Ø);SPC11
 2400      ENDPROC
 2410      DEFPROCAliens
 2420      IFK=Ø ENDPROC
 2430      IFK=1 Z=1 ELSEZ=RND(K)
 2440      IFRND(1)<.8 THENENDPROC ELSEPRINTTA
B(3,Ø);"Klingon fires"
 2450      SOUNDØ,-15,4,3Ø
 2460      Shields=Shields-INT(3ØØØ/FNA(Z))
 2470      IFShields<Ø Energy=Energy-INT((Ø-Sh
ields)*1Ø):Shields=Ø
 2480      FORI=1TO2ØØØ:NEXT
 2490      PRINTTAB(3,Ø);SPC13
 2500      ENDPROC
 2510      DEFPROCShields
 2520      Energy=Energy+Shields
 2530      PRINTTAB(2,Ø);"How much ?";SPC7;TAB
(12,Ø);
 2540      PROCInput:IFZ>1ØØØØ THEN253Ø
 2550      Shields=Z:Energy=Energy-Z
 2560      PRINTTAB(2,Ø);SPC17
```

```
 2570      ENDPROC
 2580      DEFPROCInput
 2590      A$=""
 2600      Z$=GET$
 2610      IFZ$=CHR$13 Z=VALA$:ENDPROC
 2620      IFZ$>="0"ANDZ$<="9"ANDLENA$<6 THENA
$=A$+Z$:PRINTZ$;
 2630      IFZ$=CHR$127 ANDA$=""THEN2600
 2640      IFZ$=CHR$127 A$=LEFT$(A$,LEN(A$)-1)
:PRINTZ$;
 2650      GOTO2600
 2660      DEFPROCPhasers
 2665      IF G(GX,GY) < 100 THEN ENDPROC
 2670      PRINTTAB(2,0);"How much ?";SPC7;TAB
(12,0);
 2680      PROCInput:IFZ>Energy THEN2670
 2690      FORI=1TOK
 2700        K(I,3)=K(I,3)-Z/FNA(I)
 2710        IFK(I,3)>0 THEN2750
 2720        SOUND0,-15,6,10:PRINTTAB(K(I,1)+5
,K(I,2)+3);" "
 2730        G(GX,GY)=G(GX,GY)-100:Klingons=Kl
ingons-1
 2740        S(K(I,1),K(I,2))=0
 2750        NEXT
 2760      J=0:FORI=1TOK:IFK(I,3)<=0 THENJ=I
 2770        NEXT:IFJ=0 THEN2800
 2780      IFK<>J K(J,1)=K(K,1):K(J,2)=K(K,2):
K(J,3)=K(K,3)
 2790      K=K-1:IFK>0 THEN2760
 2800      PRINTTAB(2,0);SPC17
 2810      ENDPROC
 2820      DEFPROCTorp
 2830      IFTorps=0 ENDPROC
 2840      PRINTTAB(2,0);"Press cursor key";
 2850      Z=GET:IFZ<136 THEN2850
 2860      X=0:Y=0:IFZ=136 X=-1
 2870      IFZ=137 X=1
 2880      IFZ=138 Y=1
 2890      IFZ=139 Y=-1
 2900      X1=SX:Y1=SY:Torps=Torps-1
 2910      F=0:REPEAT:X1=X1+X:Y1=Y1+Y
```

```
 2920        IFX1<1 ORX1>8 ORY1<1 ORY1>8 F=1:G
OTO3020
 2930        PRINTTAB(X1+5,Y1+3);T$
 2940        FORI=1TO100:NEXT:PRINTTAB(X1+5,Y1
+3);" "
 2950        IFS(X1,Y1)<>3 THEN3010
 2960        SOUND0,-15,5,10
 2970        G(GX,GY)=G(GX,GY)-100:Klingons=Kl
ingons-1
 2980        J=0:FORI=1TOK:IFK(I,1)=X1 ANDK(I,
2)=Y1 THENJ=I
 2990          NEXT:IFK<>J K(J,1)=K(K,1):K(J,2
)=K(K,2):K(J,3)=K(K,3)
 3000        K=K-1
 3010        IFS(X1,Y1)<>0 S(X1,Y1)=0:F=1
 3020        UNTILF=1
 3030      PRINTTAB(2,0);SPC17
 3040      ENDPROC
 3050      DEFPROCCongrats
 3060      COLOUR1
 3070      CLS:PRINT'"Congratulations"
 3080      PRINT'"You've cleared the  galaxy o
f Klingons."
 3090      PRINT''"Bet you couldn't do it agai
n though..."
 3100      PRINT'"If you think you can press Y
 else N ";
 3110      Z$=GET$:IFZ$="Y" RUN
 3120      IFZ$<>"N"THEN3110
 3130      END
```



JAMES BYRNE
Original program by Jeff Aughton

GENERAL DESCRIPTION

You are the commander of a large land-based gun which must be used to shell a convoy of submarines. The submarines are trying to find their way through some narrow straits and your task is to destroy them all before they escape from your range. If you successfully destroy a fleet of subs then you can go on to try your aim at larger convoys.

You aim your gun by controlling its elevation and bearing. The elevation of your gun affects how far the shells travel, and the bearing affects their direction. Instructions on which keys are used to effect this control appear on the screen when the program is first run.

DETAILED DESCRIPTION

Lines 10-390 this calls the procedures to set up the user-defined characters and give instructions, then initialises the variables used and draws the screen display.
400-450 check keyboard and alter angles.
460-520 start shell from gun.

530-590 move submarines.
600-760 shell lands, test for submarine and give appropriate message.
770-910 end of game, if all ships were destroyed you can go on to try with a larger convoy of submarines.
920-950 give display for submarine escaping.
960-990 procedure to delay for time measured in hundredths of seconds.
1000-1020 data for messages, and positions for subs.
1040-1240 procedure to give instructions.
1250-1400 procedure to define characters, clear screen, set window and remove flashing cursor. Define envelopes for sound.
1410-1460 procedure to place characters on screen.
1470-1530 function to read character from screen.
1540-1570 procedure to change text colour.

```
>L.
   10  REM CONVOY
   20 MODE 1
   30  PROCsetup
   40  PROCinstructions
   50  N=3:U=Ø.Ø58:V=Ø.Ø4:P=PI/18Ø:L=33733
   60  DIM M(13),S(9),T(6)
   70  HIT%=Ø:BEARING%=Ø:ELEVATION%=1Ø:K=Ø:R=
Ø:T=Ø:TS=Ø:COL=Ø
   80  RESTORE:FOR I%=Ø TO 13:READ M(I%):NEXT
   90  REM DRAW SCREEN
  100  COLOUR 13Ø
  110  VDU 28,Ø,24,39,Ø : CLS
  120  FOR I%=Ø TO 39
  122   IF RND(1Ø)<5 CH$="." ELSE CH$=CHR$234
  130    PRINT TAB(RND(39),RND(24))CH$
  140    NEXT
  150  VDU 26
  160  COLOUR 128
  170  PROCCOL:FOR I%=1 TO 19:PRINT"     ";
  180    NEXT:PRINT"     "
  190  PRINTTAB(29,2)"            "
  200  PRINTTAB(31,3)"          "
  210  PRINTTAB(33,4)"        "
  220  PRINTTAB(36,5)"     "
```

```
  230  PRINTTAB(37,6)"    "
  240  PRINTTAB(0,18)"   "
  250  PRINT"     "
  260  PRINT"      "
  270  PRINT"         "
  280  PRINT"            "
  290  PRINT"            "
  300  COLOUR 0
  310  PRINT"    ";CHR$(227);"        "
  320  PROCCOL
  330  COLOUR 130
  340  FOR I%=0 TO 24
  350    PRINTTAB(39,I%)CHR$(226):NEXT
  360  FOR I%=12 TO 38:PRINTTAB(I%,24)CHR$(22
6):NEXT
  370  FOR I%=1 TO N:READ X:S(I%)=X+32888
  380    COLOUR 3:COLOUR 130
  390    PROCPOKE(S(I%),225):NEXT:SB=TIME
  400  PROCCOL:PRINTTAB(2,1)"HITS ";HIT%;" EL
EVATION ";
  410  PRINT;ELEVATION%;"  BEARING ";BEARING%
;" ":PROCCOL
  420  K=K+1:IF K>N THEN K=1
  430  ELEVATION%=ELEVATION%-(ELEVATION%>10)*
(INKEY(-98))+(ELEVATION%<85)*(INKEY(-67))
  440  BEARING%=BEARING%-(BEARING%>-5)*(INKEY
(-103))+(BEARING%<90)*(INKEY(-104))
  450  IF(NOT INKEY(-74))OR R THEN 530
  460  SOUND 0,4,6,10
  470  FOR I%=1 TO 7
  480    TS=200*SIN(ELEVATION%*P):NEXT
  490  T=TIME
  500  R=23*SIN(2*ELEVATION%*P)
  510  X=INT(1.2*R*SIN(BEARING%*P)+.5)
  520  Y=INT(R*COS(BEARING%*P)+.5)
  530  IF R AND(TIME>T+TS) THEN 620
  540  D=-(RND(1)<U)-40*(RND(1)<V)
  550  M=S(K)+D:IF FNPEEK(M)=130 THEN 920
  560  CH%=FNPEEK(M)
  570  IF NOT( CH%=128 OR CH%=126 ) THEN 400
  580  COLOUR 3:COLOUR 130
  590  PROCPOKE(S(K),32):S(K)=M
```

```
  600  S(K)=M
  610  PROCPOKE(S(K),225):GOTO 400
  620  REM SHELL LANDS
  630  COLOUR 3:COLOUR 130
  640  R=0:W=3:Q=L+X-40*Y
  650  IF FNPEEK(Q)=129 THEN W=10
  660  FOR I=1 TO 5:PROCPOKE(Q,228+I):PROCDEL
AY(7)
  670    PROCPOKE(Q,32):PROCDELAY(7):NEXT
  680  FOR I=-3 TO 3:T(I+3)=FNPEEK(Q+I)
  690    PROCPOKE(Q+I,M(I+W)):NEXT
  700  IF W=3 SOUND 0,4,4,20:GOTO 740
  710  SOUND 0,4,6,20
  720  FOR I=1 TO N:IF S(I)<>Q THEN NEXT
  730  S(I)=S(N):N=N-1:HIT%=HIT%+1:T(3)=228:I
=N+1:NEXT
  740  PROCDELAY(90)
  750  FOR I=-3 TO 3:PROCPOKE(Q+I,T(I+3))
  760    NEXT:IF N THEN 400
  770  REM ALL DESTROYED
  780  T=INT((TIME-SB)/100)
  790  CLS:PROCCOL:PRINT TAB(13,2)"** GAME OV
ER **"
  800  PRINT'"YOU DESTROYED ";HIT%;" SHIPS"
  810  PRINT'"YOUR TIME WAS ";T;" SECS"
  820  IF K THEN 880
  830  PRINT"BECAUSE OF YOUR MISERABLE"
  840  PRINT"FAILURE YOU MUST TYPE RUN"
  850  PRINT"IF YOU WANT TO PLAY AGAIN"
  860  *FX 15,1
  870  END
  880  PRINT'"PRESS SPACE FOR A HARDER GAME"
  890  REPEAT:KEY=GET:UNTILKEY=32
  900  IF HIT%=9 HIT%=8:U=2*U:V=2*V
  910 N=N+1:PROCCOL:GOTO 70
  920  PROCPOKE(S(K),32):FOR I=0 TO 9:PROCCOL
:SOUND 1,-15,100,2:PROCPOKE(M,225)
  930    PRINTTAB(10,12)"** SHIP ESCAPES **"
  940    PROCDELAY(7):PROCCOL:PROCPOKE(M,225)
:PROCDELAY(7):SOUND 1,-15,50,2:NEXT
  950  K=0:GOTO 780
  960  DEF PROCDELAY(WAIT)
```

```
  970  NOW=TIME
  980  REPEAT UNTIL TIME>NOW+WAIT
  990  ENDPROC
 1000  DATA 83,80,76,65,83,72,33
 1010  DATA 32,66,79,79,77,33,32
 1020  DATA 2,7,43,129,50,83,166,89,125
 1030  END
 1040  DEF PROCinstructions
 1050  CLS:PRINTTAB(13,0)"** CONVOY **"
 1060  PRINT'"A CONVOY OF SUBMARINES IS SAILI
NG S-E"
 1070  PRINT"AND YOU HAVE TO DESTROY THEM ALL
 USING"
 1080  PRINT"THE GUN (";CHR$(227);") IN THE B
OTTOM CORNER OF THE"
 1090  PRINT"SCREEN."
 1100  PRINT'"ELEVATION IS MEASURED IN DEGREE
S(10-85)"
 1110  PRINT"FROM THE HORIZONTAL."
 1120  PRINT"TO INCREASE PRESS X"
 1130  PRINT"TO DECREASE PRESS Z."
 1140  PRINT"MAXIMUM RANGE OCCURS WHEN THE EL
EVATION"
 1150  PRINT"IS 45 DEGREES."
 1160  PRINT'"BEARING IS MEASURED IN DEGREES
FROM"
 1170  PRINT"-5 THROUGH 0 (NORTH) TO 90 (EAST
)."
 1180  PRINT"INCREASE BY PRESSING >"
 1190  PRINT"DECREASE BY PRESSING <"
 1200  PRINT"TO FIRE (AND START THE GAME) PRE
SS"
 1210  PRINT"RETURN"
 1220  REPEAT:UNTIL GET=13
 1230  ENDPROC
 1240  DEF PROCsetup
 1250  VDU 23,224,255,255,255,255,255,255,255
,255
 1260  VDU 23,225,0,0,14,14,12,76,76,255
 1270  VDU 23,226,255,129,129,129,129,129,129
,255
 1280  VDU 23,227,0,0,24,24,126,255,255,255
```

```
 129Ø  VDU 23,228,6Ø,66,129,153,153,129,66,6Ø
 13ØØ  VDU 23,229,Ø,Ø,Ø,&18,&1Ø,Ø,Ø,Ø
 131Ø  VDU 23,23Ø,Ø,&2Ø,&18,&1C,&28,Ø,Ø,Ø
 132Ø  VDU 23,231,Ø,&42,&24,&18,&CØ,&1A,&4Ø,Ø
 133Ø  VDU 23,232,Ø,&44,&AØ,&2Ø,Ø,&4,&62,&2Ø
 134Ø  VDU 23,233,Ø,&42,Ø,&Ø8,Ø,Ø,&44,&8Ø
 135Ø  VDU 23,234,Ø,Ø,Ø,&66,&99,Ø,Ø,Ø
 136Ø  VDU 12,28,Ø,29,39,3
 137Ø  VDU 23;1Ø,32;Ø;Ø;Ø;Ø;
 138Ø VDU 19,2,4,Ø,Ø,Ø,19,1,2,Ø,Ø,Ø
 139Ø  ENVELOPE 4,3,Ø,Ø,Ø,1,1,1,12Ø,-5,-5,-1Ø
,12Ø,9Ø
 14ØØ  ENDPROC
 141Ø  DEF PROCPOKE(LOC,CHAR)
 142Ø  LOCAL XC,YC
 143Ø  YC=INT((LOC-32768)/4Ø)
 144Ø  XC=LOC-32768-4Ø*YC
 145Ø  PRINTTAB(XC,YC)CHR$(CHAR)
 146Ø  ENDPROC
 147Ø  DEF FNPEEK(LOC)
 148Ø  LOCAL XC,YC
 149Ø  YC=INT((LOC-32768)/4Ø)
 15ØØ  XC=LOC-32768-4Ø*YC
 151Ø  VDU 31,XC,YC
 152Ø  A%=135
 153Ø  =(USR(&FFF4) AND &FFFF)DIV &1ØØ
 154Ø  DEF PROCCOL
 155Ø  COL=NOT COL
 156Ø  IF COL=Ø COLOUR 1:COLOUR 128:ENDPROC
 157Ø  COLOUR Ø:COLOUR 129:ENDPROC
```

SEARCH AND RESCUE

PAUL BORRETT
Original program by Geoffrey Little and Michael Orton

GENERAL DESCRIPTION

This is an exciting game, with good graphics and sound, requiring some skill in successfully piloting a search and rescue helicopter on its rescue mission.

By using four different keys: up = ':', down = '/', left = 'Z', and right = 'X', you move the helicopter into a position to pick up men from a ship in distress and deliver them safely to land. However, you only have a limited amount of fuel on board, so don't stray away from the land for too long and beware of crashing into the sea!

DETAILED DESCRIPTION

Lines 10-480 main body of program, defining original characters, colours and sound. The procedures are controlled from here.
490-690 PROC LADDER - drops rope to pick up man.
700-760 PROC FILLUP - fills helicopter with fuel on depression of SHIFT key.
770-800 PROC DISPLAY - displays helicopter pad scenario.
810-830 PROC SCORE - displays score on screen.
840-880 PROC UPDATE - updates screen display, for example, fuel used.
890-900 PROC FILLBOAT - fills boat with men.
910-920 PROC AGAIN - gives option of another go.
930-940 PROC DELAY (N) - gives a short delay.
950-980 PROC KEY - press any key to continue routine.
990-1000 PROC TITLE - displays title.
1010-1050 PROC INTRO - displays introduction.

1060-1240 PROC PAGE1 - displays first page of instructions.
1250-1460 PROC PAGE2 - displays second page of instructions.
1470-1620 PROC PAGE3 - displays third page of instructions.
1630-1690
1700-1730 PROC TEST - displays lowest height for dropping man.
1740-2040 PROC UDG - creates helicopter and men from given characters.
2050-2100 varies sound pitch and amplitude.
2110-2200 defines functions used.

```
LIST
   10 @%=0:*TV255,1
   20 PROCUDG
   30 MODE4:L=0:HI=0:SC=0:FU=0:FR=0:GCOL
0,0:VDU19,1,6,0,0,0:VDU23,1,0;0;0;0;:REM
 IF OS 0.1 CHANGE TO VDU23;8202;0;0;0
   40 DIML$(4),D$(25),H$(1,1,1),A$(37),H
D(9),VD(9),F$(1)
   50 PROCTITLE:PROCINTRO
   60 $&B00=CHR$246+R$+"          "+RO$+CHR
$250+"  "+U$+STRING$(10,L$)+CHR$251+";;;;
;"+R$+CHR$247+CHR$252+RO$+CHR$250+"  "+ST
RING$(3,L$)+U$+CHR$233+CHR$253+"  "
   70 D$(0)=CHR$30:FORX=1TO25:D$(X)=D$(X
-1)+D$:NEXT
   80 FORX=1TO37:A$(X)=A$(X-1)+CHR$9:NEX
T
   90 IFQ$="Y"ORQ$="y"PROCPAGE1
  100 L$(0)=STRING$(3,CHR$232)+L$+L$+L$+
D$+STRING$(3,CHR$232):L$(1)=R$+L$(0)+RO$
  110 FORX=2TO4:L$(X)=L$(0):NEXT
  120 H$(0,0,0)=CHR$228+CHR$233+CHR$228+
D$+L$+L$+L$+CHR$224+R$+CHR$227+RO$+CHR$2
37
  130 H$(1,0,0)="  "+CHR$233+"  "+D$+L$+L$
+L$+CHR$224+R$+CHR$227+RO$+CHR$237
  140 H$(0,1,0)=CHR$228+CHR$233+CHR$228+
D$+L$+L$+L$+CHR$236+R$+CHR$227+RO$+CHR$2
24
```

```
  150 H$(1,1,0)="  "+CHR$233+"  "+D$+L$+L$
+L$+CHR$236+R$+CHR$227+RO$+CHR$224
  160 H$(0,0,1)=CHR$228+CHR$233+CHR$228+
D$+L$+L$+L$+CHR$224+R$+";"+RO$+CHR$237
  170 H$(1,0,1)="  "+CHR$233+"  "+D$+L$+L$
+L$+CHR$224+R$+";"+RO$+CHR$237
  180 H$(0,1,1)=CHR$228+CHR$233+CHR$228+
D$+L$+L$+L$+CHR$236+R$+";"+RO$+CHR$224
  190 H$(1,1,1)="  "+CHR$233+"  "+D$+L$+L$
+L$+CHR$236+R$+";"+RO$+CHR$224
  200 F$=CHR$235+STRING$(20,CHR$228)+CHR
$234+A$(18)+CHR$238+"Thankyou for playin
g"+CHR$239+A$(18)
  210 F$=F$+CHR$236+STRING$(20,CHR$227)+
CHR$237+CHR$30:EM$=R$+"E"+L$+D$+"M"+L$+D
$+"P"+L$+D$+"T"+L$+D$+"Y"+L$+D$+"  "+L$+D
$+"  "+RO$
  220 X$="ABDEDBABDFDBAACDEDBAABDEFDCDBA
ABCDBAABCA"
  230 FORX=1TO40:W$=W$+CHR$(ASC(MID$(X$,
X,1))+159):NEXT
  240 IFQ$="Y"ORQ$="y"PROCPAGE2
  250 F$(0)=R$:F$(1)=RO$
  260 E$="    "+D$+L$+L$+L$+"    "
  270 EX$=CHR$241+CHR$240+CHR$242+D$+L$+
L$+L$+CHR$242+"V"+CHR$241+U$+L$+L$+L$+CH
R$235+CHR$243+CHR$234+D$+L$+L$+L$+CHR$23
6+CHR$245+CHR$237+U$+L$+L$+L$
  280 C$=R$+"  "+CHR$246+RO$+CHR$233+"
  "+R$+CHR$247+CHR$248+RO$+"  "+U$+STRING$
(10,L$)+R$+CHR$246+RO$+"        "+CHR$233+
CHR$242+U$+L$+L$+"   "+D$+D$+STRING$(9,L$
)
  290 C1$=U$+"  "+D$+L$+"            "+U$+L$
+L$+"    "
  300 FB$=STRING$(40," ")
  310 IFQ$="Y"ORQ$="y"PROCPAGE3
  320 SOUND17,0,0,1:VV=0:HV=0:VP=6:HP=0:
R=0:D=0:M=0:SC=0:FU=500:LP=604:PROCFILLB
OAT
  330 CLS:SOUND3,1,200,255:PROCDISPLAY
  340 IFINKEY(-99)PROCLADDER
```

```
  350 Q2=0:Q3=0:IFINKEY(-98) Q2=-0.32
  360 IFINKEY(-67) Q2=0.32
  370 IFINKEY(-105) Q3=0.32
  380 IFINKEY(-73) Q3=-0.32
  390 V=VV+Q3:VV=V-0.32*((V<-2.1)-(V>2.1
)):V=HV+Q2
  400 HV=V-0.32*((V<-2.1)-(V>2.1)):FU=IN
T((FU-ABS(HV)-ABS(VV+.2)-M)*.995)
  410 PRINTD$(23)A$(28)FU" ":IFFU<.1 FU=
0:GOTO610
  420 D=-(HV<0):R=1-R:V=VP:H=HP
  430 P=VP+VV:IFP>1 ANDP<22 VP=P
  440 P=HP+HV:IFP>0 ANDP<36 HP=P
  450 PRINTD$(V)A$(H)E$:IFFNC(VP)<>32ORF
ND(VP)<>32 THEN650
  460 PRINTD$(VP)A$(HP)H$(R,D,M):SOUND16
,-15,4,2:SOUND17,-15,5,1
  470 IFINT(HP)=0 ANDINT(VP)=6 ANDINKEY(
-1) PROCFILLUP
  480 PROCUPDATE:GOTO340
  490 DEFPROCLADDER
  500 FU=FU-5:IF-M THEN570
  510 IFFNB(VP+2)<>32ENDPROC
  520 P=VP+1:REPEAT:P=P+1:IFFNB(P)=32 PR
INTD$(P)A$(HP+1)CHR$239;:UNTILP=VP+6:P=P
+1 ELSEUNTIL-1
  530 IFFNB(P)<>59 M$=CHR$239:GOTO550
  540 ?(2845+(INT(HP)-INT(BP))+1)=32:M$=
";":M=1:PRINTD$(21)A$(BP)$&B00:SOUND1,2,
100,50
  550 FORP=P-1TOVP+2STEP-1:PRINTD$(P)A$(
HP+1)M$;
  560    PROCDELAY(20):PRINTL$" ":NEXT:EN
DPROC
  570 M=0:R=1-R:P=VP+2:PRINTD$(VP)A$(HP)
H$(R,D,M)D$L$L$;
  580 IFFNB(P)=32 PRINT";"L$;:PROCDELAY(
10):PRINT" "D$L$;:P=P+1:GOTO580
  590 IFP-VP>4 ORFNB(P)<>153 VDU30:PRINT
"SPLAT":PROCDELAY(600):SOUND1,-15,30,100
:VDU30:PRINTSPC5:GOTO390
  600 SC=SC+1:PRINTD$(23)A$(6)SC:FORP=HP
```

```
 TO1STEP-1:J=FNG(33568+P):PRINTTAB(P,2Ø)
";"L$;:PROCDELAY(5Ø):VDUJ:NEXT:SOUND1,2,
2ØØ,5Ø:GOTO39Ø
  61Ø IFHP<9 ANDFNB(VP+2)<>32 THEN75Ø
  62Ø PRINTD$(VP)A$(HP)E$:VV=VV-.3*(VV<.
7):VP=VP-VV*((VP+VV)>Ø):HV=HV*.9:HP=HP+H
V
  63Ø IFHP<ØORHP>37ORFNC(VP)<>32ORFND(VP
)<>32THEN65Ø
  64Ø PRINTD$(VP)A$(HP)H$(R,D,M):PROCDEL
AY(2Ø):GOTO62Ø
  65Ø PRINTD$(VP)A$(INT(FNE(HP)));:FORX=
1TO3Ø:SOUNDØ,-15,RND(3)+3,1:PRINTEX$;:NE
XT:PRINTE$
  66Ø SOUND19,Ø,Ø,1:IFSC>HI HI=SC:SOUND1
,2,23Ø,1ØØ:PRINTTAB(14,Ø)"New High Score
"
  67Ø SOUND17,3,2ØØ,255:PROCDISPLAY:PROC
UPDATE:*FX15,1
  68Ø PROCUPDATE:PROCAGAIN:QW$=INKEY$(Ø)
:Q=(QW$="Y"ORQW$="y")+(QW$="N"ORQW$="n")
*2:ON1-Q GOTO68Ø,32Ø,69Ø
  69Ø PRINTD$(11)A$(11)F$:SOUND17,Ø,Ø,1:
END
  7ØØ DEFPROCFILLUP
  71Ø IFNOTINKEY(-1)ENDPROC
  72Ø PRINTD$(23)A$(28)FU" "
  73Ø IFLP>38Ø FU=FU+36:MOVE32,LP:DRAW6Ø
,LP:LP=LP-4:SOUND1,-15,65Ø-LP,1:GOTO71Ø
  74Ø PRINTD$(13)A$(1)EM$:ENDPROC
  75Ø PRINTD$(VP)A$(HP)H$(R,D,M)
  76Ø PRINTD$(15)A$(12)"You Ran Out Of F
uel"D$A$(22)"But Landed Safely":GOTO66Ø
  77Ø DEFPROCDISPLAY
  78Ø VDU31,Ø,8,233,228,233,1Ø,8,8,8,232
,232,232,1Ø,8,8,8,232,232,232,1Ø,8,8,8,2
46,248,247,1Ø,8,8,8
  79Ø PRINTR$"o  "STRING$(5,(D$+L$+L$+L$
+"    "))D$L$L$L$"  o"D$L$L$L$"    "D$L$L$
L$" "RO$" "CHR$248D$L$L$L$STRING$(9,CHR$
249)
  8ØØ PROCSCORE:ENDPROC
```

```
  810 DEFPROCSCORE
  820 PRINTD$(23)"Score:";SC;"      Helic
opter Fuel:"FU''"Hi-score:"HI"   "
  830 PROCUPDATE:ENDPROC
  840 DEFPROCUPDATE
  850 WP=RND(38):PRINTD$(22)MID$(W$,WP)L
EFT$(W$,WP-1);
  860 BP=BP-RND(1)/3:IFBP<31 PRINTD$(21)
A$(BP)$&B00
  870 PRINTD$(9)L$(L):L=L+1:L=-(L<5)*L:I
FINT(BP)<>9 ENDPROC
  880 PRINTD$(21)A$(BP);:PROCDELAY(150):
PRINTC$;:PROCDELAY(150):PRINTC1$:PROCFIL
LBOAT:ENDPROC
  890 DEFPROCFILLBOAT
  900 FORX=2846TO2850:?X=59:NEXT:BP=RND(
10)+30:ENDPROC
  910 DEFPROCAGAIN
  920 PRINTD$(2)A$(12)"Press 'Y' To Rest
art":PRINTD$(4)A$(14)"Press 'N' To Stop"
:PROCDELAY(50):ENDPROC
  930 DEFPROCDELAY(N)
  940 FORZ=1TON:NEXTZ:ENDPROC
  950 DEFPROCKEY
  960 PRINT''"       Press Any Key To cont
inue"CHR$30
  970 QW$=GET$
  980 PROCTITLE:ENDPROC
  990 DEFPROCTITLE
 1000 CLS:PRINTTAB(11,0)R$"Search & Resc
ue"RO$:ENDPROC
 1010 DEFPROCINTRO
 1020 PRINT'TAB(13)"(c). 1982"'TAB(13)"B
y G.J.Little"'TAB(13)"& M.Orton"
 1030 PRINT''''''"   Do you want instruc
tions ? ";
 1040 Q$=GET$:QW$="No":IFQ$="Y"ORQ$="y"Q
W$="Yes"
 1050 PRINTQW$:ENDPROC
 1060 DEFPROCPAGE1:PROCTITLE
 1080 PRINT'"   You are the pilot of a s
earch and"
```

```
 1090 PRINT" rescue helicopter. The heli
copter is"
 1100 PRINT" controlled using the follow
ing keys"
 1110 PRINT" to accelerate you in your c
hosen         direction"
 1120 PRINT''SPC13"':' - Up"
 1130 PRINT'SPC2"'Z' - Left"SPC12"'X' -
Right"
 1140 PRINT'SPC13"'/' - Down"
 1150 PRINT'"    You may also use a combi
nation of"
 1160 PRINT" keys,i.e. 'X' and '/' to go
 down and"
 1170 PRINT" to the right at the same ti
me."
 1180 PRINT'"    Other controls are the S
HIFT key"
 1190 PRINT" and the SPACE BAR."
 1200 PRINT'"    The SHIFT key operates r
efuelling"
 1210 PRINT" from the top of the lightho
use only."
 1220 PRINT" It is possible to return th
ere to"
 1230 PRINT" refuel during sorties."
 1240 PROCKEY:ENDPROC
 1250 DEFPROCPAGE2:PROCTITLE
 1260 PRINT'"    The SPACE BAR operates t
he rope,"
 1270 PRINT" dropping the passenger when
 one is"
 1280 PRINT" carried or lowering the rop
e when one"
 1290 PRINT" is not."
 1300 PRINT'"    The idea is to manoeuvre
 the"
 1310 PRINT" helicopter above the approa
ching ship"
 1320 PRINT" and lower the rope onto one
 of the"
 1330 PRINT" men on deck,who will then c
```

```
limb up"
 1340 PRINT" into the helicopter. To bri
ng the man"
 1350 PRINT" to safety you must again ma
noeuvre so"
 1360 PRINT" that you are situated above
 the rock"
 1370 PRINT" platform next to the lighth
ouse. Then"
 1380 PRINT" simply depress the SPACE BA
R to drop"
 1390 PRINT" the passenger who will then
 run into"
 1400 PRINT" the lighthouse."
 1410 PRINT'"   If you begin to run low
on fuel,"
 1420 PRINT" move the helicopter to be s
tationary"
 1430 PRINT" above the helicopter platfo
rm (Start"
 1440 PRINT" Position) and depress the S
HIFT key"
 1450 PRINT" until you think you have en
ough fuel."
 1460 PROCKEY:ENDPROC
 1470 DEFPROCPAGE3:PROCTITLE
 1480 PRINT'"   POINTS TO NOTE:-"
 1490 PRINT''"   The rate of fuel consum
ption is"
 1500 PRINT" related to the amount of fu
el carried,";
 1510 PRINT" carrying a passenger, lower
ing the"
 1520 PRINT" rope and your speed."
 1530 PRINT'"   The keys accelerate you.
 Thus to"
 1540 PRINT" change direction you will s
low down,"
 1550 PRINT" turn around and then accele
rate."
 1560 PRINT'"   Passengers being fairly
fragile"
```

```
 1570 PRINT" cannot be dropped above the
 height of"
 1580 PRINT" the lowest lighthouse windo
w."
 1590 PRINT'"   When you run out of fuel
,if you are"
 1600 PRINT" positioned above the helico
pter,or"
 1610 PRINT" rock platform you will land
 safely."
 1620 PROCKEY
 1630 BP=25:PROCDISPLAY
 1640 FI$=CHR$233+CHR$228+CHR$233+RO$:FD
=8:FA=0:FQ$="Helicopter Pad":FL=24:PROCT
EST
 1650 FI$=STRING$(7," "+L$+D$):FD=13:FA=
1:FQ$="Fuel Gauge":FL=25:PROCTEST
 1660 FI$="o":FD=18:FA=2:FQ$="Minimum He
ight For Dropping Man":FL=24:PROCTEST
 1670 FI$=";":FD=20:FA=26:FQ$="Men":FL=2
5:PROCTEST
 1680 PRINTD$(5)A$(8)"Press Any Key To S
tart"
 1690 QW$=GET$:ENDPROC
 1700 DEFPROCTEST
 1710 PRINTD$(5)A$(19-LEN(FQ$)/2)FQ$
 1720 FORX=1TOFL:PRINTD$(FD)A$(FA)F$(FR)
FI$:FR=1-FR:PROCDELAY(100):NEXT
 1730 PRINTD$(5)A$(19-LEN(FQ$)/2)LEFT$(F
B$,LEN(FQ$)):ENDPROC
 1740 DEFPROCUDG
 1750 VDU23,224,255,0,0,0,0,0,0,0
 1760 VDU23,225,0,255,0,0,0,0,0,0
 1770 VDU23,226,0,0,255,0,0,0,0,0
 1780 VDU23,227,0,0,0,255,0,0,0,0
 1790 VDU23,228,0,0,0,0,255,0,0,0
 1800 VDU23,229,0,0,0,0,0,255,0,0
 1810 VDU23,230,0,0,0,0,0,0,255,0
 1820 VDU23,231,0,0,0,0,0,0,0,255
 1830 VDU23,232,129,66,36,24,24,36,66,12
9
 1840 VDU23,233,0,0,0,0,255,24,24,24
```

```
 1850 VDU23,234,0,0,0,0,240,8,4,4
 1860 VDU23,235,0,0,0,0,15,16,32,32
 1870 VDU23,236,32,32,16,15,0,0,0,0
 1880 VDU23,237,4,4,8,240,0,0,0,0
 1890 VDU23,238,32,32,32,32,32,32,32,32
 1900 VDU23,239,4,4,4,4,4,4,4,4
 1910 VDU23,240,24,60,90,153,24,24,24,24
 1920 VDU23,241,128,64,32,16,8,4,2,1
 1930 VDU23,242,1,2,4,8,16,32,64,128
 1940 VDU23,243,24,24,24,24,255,0,0,0
 1950 VDU23,244,24,24,24,255,0,0,0,0
 1960 VDU23,245,0,0,0,255,24,24,24,24
 1970 VDU23,246,255,127,63,31,15,7,3,1
 1980 VDU23,247,255,254,252,248,240,224,
192,128
 1990 VDU23,248,255,255,255,255,255,255,
255,255
 2000 VDU23,249,170,85,170,85,170,85,170
,85
 2010 VDU23,250,192,192,192,192,192,192,
192,192
 2020 VDU23,251,0,0,0,0,0,0,255,255
 2030 VDU23,252,24,24,24,24,255,24,24,24
 2040 VDU23,253,128,128,128,128,128,128,
128,128
 2050 ENVELOPE2,3,0,0,0,0,0,0,121,-10,-5
,-2,120,120
 2060 ENVELOPE1,2,80,50,80,20,30,20,20,0
,0,-127,75,0
 2070 ENVELOPE3,4,16,12,8,10,8,10,10,0,0
,0,120,0
 2080 R$=CHR$17+CHR$0+CHR$17+CHR$129:RO$
=CHR$17+CHR$1+CHR$17+CHR$128
 2090 L$=CHR$8:U$=CHR$11:D$=CHR$10
 2100 ENDPROC
 2110 DEFFNG(X)
 2120 I=X-32768
 2130 VDU31,I MOD40,I DIV40
 2140 A%=135:C=USR(&FFF4) AND&FFFF:C=C D
IV256
 2150 =C
 2160 DEFFNA(X)=32768+INT(X)*40+HP
 2170 DEFFNB(X)=FNG(FNA(X)+1)
```

```
 2180 DEFFNC(X)=FNG(FNA(X+1)+2*(1-D))
 2190 DEFFND(X)=FNG(FNA(X+1)+2*D)
 2200 DEFFNE(X)=(X+(X<0)*X+(X>37)*(X-37)
)
```

HINTS AND TIPS

BBC HAND RESET

Pressing the BREAK key on the BBC Computer returns the system to Mode 7, and deletes any program from memory. The message

BBC Computer
BASIC
>

appears on the screen. However, this does not reset the clock or clear the definitions of the user-defined function keys. This is what the User Guide calls a 'soft reset'.

According to the User Guide a 'hard reset' (clock reset and user-defined keys reset) can be obtained in operating system version 1.0 and above by pressing BREAK and CTRL keys together. Users with OS 0.1 can obtain a hard reset by pressing the BREAK key twice in quick succession. This causes the CHR$(7) 'bell' to be printed and the message

BBC Computer 32K
BASIC
>

to appear. The variable TIME will be set to 0 and all the user-defined keys will be cleared.

HOME SERVICE

If you would like to know which version of the operating system is installed in your machine, try talking to it. *FX0 will tell you the details of the ROM containing the OS e.g. OS 0.1 EPROM.

*HELP will give you details of all the ROMs fitted to your machine and may even give you further information on their contents, all without so much as lifting a screwdriver.

BBC RESTARTER

Although the BBC micro has no equivalent of the Microsoft CONT, restarting a program which has been halted (e.g. by accidentally pressing the 'escape' key) can be achieved using GOTO.

It is worth noting, however, that if a line has been edited or deleted then the variables, except A%-Z% will all have been lost. Restarting the program will work if no editing has occurred,

except in the cases where the error has taken place inside a loop or inside a subroutine or procedure.

LIGHTEN THOSE LONG LOADS

If you're using your BBC micro for word processing with a tape system, you can set into problems with long documents. Using either Wordwise or View it is possible to LOAD only parts of long files. To do this, you need to know roughly the position in the file of the piece you want. Then use *OPT 2,0 to make the tape ignore errors and Fast Forward/Rewind to read only the blocks that you want.

The BBC will LOAD this part, corrupting only the first and last blocks. As a rough guide, you set around 50 words in a block.

BBC USER GUIDE PASSES THE TON

If you type in the program on page 61 of the BBC User Guide exactly as it is printed you will find it stops at line 100. The solution is to take out the space between RND and (4).

CUT BACK ON BRACKETS

On the BBC micro the message "Missing)" can sometimes be caused by having too many brackets rather than one missing, as in the following, which uses numbers stored in arrays as graphics co-ordinates:-

```
4060 DIM Xpoint(100), Ypoint(100)
4070 FOR count=1 TO 100
4080 MOVE (Xpoint(count), Ypoint(count))
4090 VDU 18,0,0,227:NEXT count
```

This will cause the error message "Missing) at line 4080", when in fact the line should be:-

```
4080 MOVE Xpoint(count), Ypoint(count)
```

This slip is easily made as, if one were using text co-ordinates, an equivalent line would be:-

```
4080 PRINT TAB (Xpoint(count), Ypoint(count));CHR$227;
```

This does not occur when using brackets in calculations.

BBC 'BAD PROGRAM' FIX

If you have trouble loading tapes, perhaps because of a dropout or some other gremlin, or write programs which go crashing through the Basic buffer and bring up the message 'Bad Program',

you are then in a situation where most of the program may be intact but you can not LIST because the Bad Program flag will not go away. If you wish to recover at least something, then try typing the following as a command line, that is, one without a line number. (Any lines starting with a number would be fatal at this stage.)

A=PAGE:REP.B=A?3:IFA?B=13 A=A+B:U.0 ELSE U.
TRUE:A?B=13:A?(B+1)=255[RETURN]

When the prompt returns type: END[RETURN]

You should then be free to LIST the remaining, if slightly corrupted, part of the program. If the prompt fails to return after a few seconds, no harm will have been done – do not despair, just press ESCAPE and the dreaded 'Bad Program' will reappear. Now type:-

A=PAGE+4:REP.A=A+1:U. ?A=13:P.~A-PAGE[RETURN]

If, for example, this returns 61 then type:-

!PAGE=&6101000D[RETURN]

and try the previous fix again. All will now be well, but make sure you fix any subtle errors which may have occurred in the listing before RUNning it.

BBC BAD PROGRAM FIX

The BBC Computer is prone to corrupt the first block of a program. This can be very annoying since you cannot even load the remainder of the program to try the 'BAD PROGRAM' fix detailed above. To overcome this use the following method to load the program, list it and insert the missing lines again to enable it to run.

Position the tape at the commencement of the faulty program and then take it out of the recorder. Place a new tape in the recorder and then type in a program of two lines, 10 and 20, each containing the maximum number of single characters, X is convenient. SAVE this program naming it with the name of the program you are trying to recover. Then reload the program, but press 'STOP' on the cassette player the instant the first block has been loaded – this must be judged by ear alone. Replace the tape with the original program tape and press 'PLAY' and allow the program to continue loading. This will not quite fool the computer however, and on completion of loading the 'Bad Program' message will appear. This can then be recovered with the

following modified version of the previous fix. No line numbers!
A=PAGE : REPEAT X=A?3: IF A?X=13 A=A+X UNTIL 0 ELSE UNTIL 1 : Z=0 : REPEAT Z=Z+1: UNTIL ?(A+Z)=13:?(A+3) =Z

Press [RETURN] and wait for the prompt to reappear, when with luck the program should be listable. Of course, the first few lines will only be 'X's, but at least it is now capable of repair.

BBC HEX INPUT

If you have ever wanted to set a hex number input into your BBC Computer for disassembly, etc. try the following:

```
10 INPUT A$
20 A%=EVAL ("&"+A$)
```

A% will now contain the value of the hexadecimal number specified in A$.

BEEB BREAK

The BREAK key on the BBC Computer can be defined just like the other user definable function keys. This is done by *KEY10 and can be very useful when used with *KEY10OLD**I**M**I**NLIST**I**M. See page 143 of the manual.

STRING YOUR BBC ALONG

At times it is useful to program one of the BBC function keys with a string which begins with a quotation mark. For instance, if you were entering a lot of PRINT statements it would be nice to program f0 to produce "**I**MPRINT". Then you could use AUTO and just press f0 at the end of each line allowing tasks like writing instructions for games to be performed easily.

The problem comes when you try to program the key with *KEY0"**I**MPRINT".

The BBC will assume the quotes are the delimiters of the string **I**MPRINT which you wish to place on the key. We could use the double quotes inside quotes method, to give a string which contains the required quotation marks i.e. *KEY0""""**I**MPRINT"""" but this is generally confusing. A much neater solution relies on the fact that the BBC only recognises the quote as a delimiter if it occupies the first position, so we can insert a harmless control

I is obtained from the upper symbol on the key to the left of the ← key. It looks like ll in MODE 7.

code before the required string thus:-
*KEY0**I**@"**I**MPRINT"
and achieve the desired result without counting up strings of quotes.

KEY-IN

The user-defined function keys on the BBC can be used for many purposes but it is often too much effort to program them directly each time you sit down.
You can use the following short program to initialise them for you:-

```
10 *KEY 0 OLD**I**MMODE 7**I**M**I**NLIST**I**M
20 *KEY 1 CLS**I**MRUN**I**M
30 *KEY 2 MODE7**I**MLIST**I**M
```

Obviously this can be changed and extended to suit any particular user's needs but all the keys must be loaded on separate BASIC lines since the computer doesn't respond to colons when sending commands to the operating system.

CLOCKING-ON TIME FOR THE BBC

Here's a quick way of getting the BBC to display a real time clock. Define function keys 0 and 1 as shown below.

```
*KEY0 CLS:I. "Hours,mins,secs",H%,M%,S%:TI.=100*(S%+M%*
60+H%*3600) **I**M
*KEY1 CLS:TI.=TI.  MOD8640000:P.TI.  DIV360000":";
(TI.  DIV6000)MOD60":";(TI.  DIV100)MOD60**I**M
```

Press f0 and type in the time. From now on pressing f1 will tell you the time.

BEEB STRING SEARCH

Here is another extremely useful program that will fit on one function key of your BBC Computer. It will search through a BASIC program looking for a specific string (you INPUT it) and will report the line numbers at which the string occurs. This is very useful for debugging long programs (for example adventures) where you have lost track of where certain procedures and variables are mentioned. Note that you can't use it to look for BASIC keywords as these are stored as token values.

```
*KEY9CLS:I."Enter string",N$:P=PAGE+1:REPEAT:N=256*P?0
+P?1:P=P+2:L=P?0:NL=P+L-2:P=P+1:IF INSTR($P,N$)<>0
THENP.;N:P=NL:U.?P=&FF:END:ELSEP=NL:U.?P=&FF:END**I**M
```

To execute a search press f9 and enter the string you wish to find when prompted. The program will reply by listing all the line numbers at which the string occurs.

While entering this program you may come across one of the annoying problems with function keys. The function keys can only hold 255 characters altogether (a good reason to use abbreviations for keywords). If the number of characters on the keys already plus the number you are trying to program exceeds 255, you set a 'Bad Key' error. However, this doesn't allow for those characters which are already on the key and will be erased by the new definition. The best way to avoid this is to use *KEYn[RETURN] before defining a key (where n is the number of the key you wish to use). Of course, you can simply remove all the definitions at once with *FX18.

KEY SAVER

Having defined your favourite set of function key programs a simple way of storing them so that they can be recalled at a later date is to save the soft key buffer itself. This can be done with the command:

*SAVE "FILE" B00 BFF

This will save page &B of memory where the definitions are stored. This can be reloaded at any time using *LOAD and will not affect any programs currently in memory. Since the file is only one block long it doesn't take very long to load either.

BEEB VANISHING TRICK

It is quite often useful to have programs that vanish after running. On the BBC micro an example would be a !BOOT program which erases itself after printing a message and resetting the function keys.

Unfortunately you cannot just stick NEW at the end of the program since this is a direct command only. But you can call the language initialisation routines with CALL &8000. For systems with the new BASIC you will have to set A%=1 before the call. OLD can of course still be used to set your program back.

BEEB'S SECRET COLOURS

The following is a simple program to display some of the numerous shades of colour which can be obtained using the BBC micro.

Colours are mixed by drawing parallel lines close together and the distance between lines is determined by the step size in line 50.

All the effects are obtained using a two colour mode. MODE 1 or 5 would allow four colours to be mixed at once.

```
10 MODE 0
20 FOR C=7 TO 0 STEP -1
30 FOR K=0 TO 7
40 VDU 19,0,C,0,0,0,:VDU 19,1,K,0,0,0
50 FOR X%=0 TO 1280 STEP 8
60 MOVE X%,0
70 DRAW X%,1023
80 NEXT X%
90 NEXT K
100 NEXT C
```

MAKING SHIFT WITH BBC FUNCTIONS

For games you can use INKEY to read the keyboard but this can lead to problems when the auto-repeat speed isn't keeping up with the INKEY function. You can get round this by resetting the repeat speed to be very fast with *FX12,1. Remember to set it back to normal with *FX12,0 before leaving the program. This method will not, however, allow the detection of some keys like 'shift' or 'caps lock'. A better way round the problem is to use INKEY with a negative number. This doesn't wait for a set time but immediately checks whether one particular key is being held down or not. For instance INKEY(-1) is TRUE if 'shift' is held down and FALSE if it isn't. You can test any key at any time in this way regardless of what other keys are held down. Further information starts on page 273 of the User Guide.

BBC INKEY

The machine code equivalent to INKEY on the BBC is described as an OSBYTE call with A=&81. If Y contains a positive number (in two's complement notation) the OSBYTE call is equivalent to INKEY with a positive parameter. The User Guide then mentions that if Y is negative the call can be used to test for specific key closure, (i.e. INKEY with a negative parameter). However, it doesn't make it clear that Y can contain any negative number and it is X which holds the number for the key to be tested. For

example, to test the 'shift' key you would need to use the call with A=&81, Y=&80 and X=(256-1) (where -1 is the figure given for the 'shift' key in the INKEY section of the User Guide).

POKE SUBSTITUTE

For anyone mourning the lack of a standard PEEK and POKE on the BBC Computer here are a couple of declarations they could try in their programs.

```
DEF PROCPOKE(A,B):?A=B:ENDPROC
DEF FNPEEK(A):=?A
```

You will now be able to POKE by saying PROCPOKE(address, data) and PEEK by saying data=FNPEEK(address).

At least these look vaguely like the familiar forms. Naturally getting the hang of the !,$ and ? operators is better in the long term and more flexible.

BBC WRAP AROUND

The routine below can be used instead of the PRINT command to output messages for games, programs, etc. without splitting words across two lines. The routine as it stands will accept lines of any length and cut them down into forty character lines. The 40 can, of course, be changed to 20 or 80 for other modes, or anything else to print inside windows or other restrictions.

```
2000 DEF PROCno  split(A$)
2010 LOCAL L%
2020 L%=LEN(A$)
2030 IF L%+COUNT<40 THEN PRINT A$ : ENDPROC
2040 REPEAT
2050 REPEAT
2060 REPEAT
2070 L%=L%-1
2080 UNTIL L%<1 OR MID$(A$,L%,1)=" "
2090 UNTIL L%+COUNT<=40
2100 PRINT LEFT$(A$,L%-1)
2110 A$=RIGHT$(A$,LEN(A$)-L%)
2120 L%=LEN(A$)
2130 UNTIL L%<40
2140 PRINT RIGHT$(A$,L%);
2150 ENDPROC
```

CAPITALISE ON YOUR BBC MICRO

Here are two functions which take a string and return it with all its letters in upper case. The first is straight-forward and requires no setting up. The second executes much faster but requires that DIM Z% 255 has been executed at the start of the program:-

```
10 DEF FNUPRC(A$) : IF A$="" THEN =""
20 LOCAL A%,B% : FOR A%=1 TO LEN A$ : B%=ASC(MID$
(A$,A%))
30 IF B%>96 AND B%<123 THEN A$=LEFT$(A$,A%-1)+CHR$
(B%-32)+MID$(A$,A%+1)
40 NEXT : =A$
```

Alternatively after DIM Z% 255 you could use:-

```
10 DEF FNUPRC(A$) : IF A$="" THEN =""
20 LOCAL A%,B% : $Z%=A$ : FOR A%=0 TO LEN A$-1 : B%=Z%
?A%
30 B%=B%+32*(B%>96 AND B%<123): Z%?A%=B%
40 NEXT : =$Z%
```

These functions can be used to make keyboard entry checking easy e.g.:-

```
KEY$=FNUPRC(GET$) : IF KEY$="Y" THEN END
INPUT A$ : A$=FNUPRC(A$) : IF A$="YES" THEN END
```

Another handy tip is that all lines starting DEF are ignored by the computer. So you can have different names and different entry points for the same procedure. You could even make it a procedure and a function:-

```
1000 DEF PROCYIPPEE(A$,W%) : LOCAL F% : F%=TRUE
1010 DEF FNYIPPEE(A$,W%) : LOCAL F% : F%=FALSE
''
''
''
1050 IF F% THEN ENDPROC ELSE =A$
```

You could use this to let the user use procedure calls, e.g.:-

```
100 ON ERROR PRINT "Bad procedure call"
110 INPUT A$ : A=EVAL("FN"+A$)
120 ON ERROR OFF
```

On error allows you to use direct commands as long as they immediately follow the ON ERROR statement. Unfortunately ON ERROR forgets about procedures and loops, so it is only useful for error trapping in the main program.

BEEB KEYBOARD BY-PASSED

If you still have operating system 0.1 and you want to put characters into the keyboard buffer, here is a routine to allow you to simulate *FX 138 (available from version 1.0 onwards). This puts characters directly into the keyboard buffer between locations &3E0 and &3FF. The present position in the buffer is pointed to by &234 with &23C pointing to the last character entered. The differences between these two are characters waiting to be processed.

To use the procedure enter PROCstring("HEY, LOOK!") or PROCchar(13). As an example, suppose an INPUT wants the answer YES or NO and NO is the expected answer, you could use:-

```
100 PROCstring("NO")
110 INPUT "Exit program ? (YES/NO)",A$
```

You might also find it handy to clear the buffer directly by using ?&23C=?&234

```
100 DEF PROCstring(A$)
110 LOCAL X,C
120 FOR X=1 TO LEN A$ : C=ASC(MID$(A$,X))
130 PROCchar(C) : NEXT : ENDPROC
140 DEF PROCchar(C)
150 ?(&300+?&23C)=C : ?&23C=?&23C+1 : IF ?&23C=0 THEN
?&23C=&E0
160 ENDPROC
```

BBC DUMP

Here is a short program to dump all the programs from a cassette to a printer. As the LIST command cannot be used in a program it is placed on the function key f0. This contains the program to load and list the program, then jump to line 80 where the code for f0 is placed directly into the keyboard buffer. Users with OS 0.1 will need to use the routine given above instead of the *FX 138 call used here. This technique can be used to execute other direct commands from inside programs.

```
10 MODE 7
20 *FX6,0
30 *KEY0P%=PAGE:PAGE=&2000|M*LOAD""2000|MLISTO7
|MIB L.|MIL|C PAGE =P%:GOTO80|M
40 CLS : PRINT TAB(0,10)"BATCH PRINTER" ' "Load cassette
and set up printer" '
```

```
"to skip to next program" ' "press [ESCAPE] then f0. " ' "Press
space bar when ready"
50 *FX137,1
60 WAIT=GET : CLS
70 *FX137,0
80 *FX138,0,144
90 END
$
```

GRAPHIC MEMORY

It is difficult to fault the BBC micro, but one criticism which is justifiably levelled against it is the shortage of RAM available when full use is made of the memory mapped graphics. This note describes the steps required to locate a program at a lower memory address than usual, effectively giving additional program/data space.

The technique is based on assigning an appropriate value to the variable PAGE before LOADing. CHAINing or typing in a program; for example:

```
> PRINT PAGE 3584 :REM The default value is &E00 = decimal
3584.
> PAGE=3328 :REM The new value must be on a page boundary
(3328 = &D00).
> NEW :REM This is only necessary if the program is to be typed
in,
> LOAD "Long-prog" :REM it is optional before a LOAD or
CHAIN operation.
```

The new page boundary chosen depends on the facilities employed. In particular, pages &0C and &0D may be used if there are no user supplied routines or characters, page &0B if the programmable keys are not employed, pages &09 and &0A if the serial port is unused and lastly page &08 when the program is free of sound commands. Needless to say, the ***lowest*** value of PAGE is determined by the ***highest*** unused page, since the Basic program must be stored in contiguous locations. However, you can always store machine code subroutines below PAGE: for example, if you require special characters (page &0C) you could put machine code between &09 and &0C.

Fairly straight-forward - but there is a catch. Typing "OLD" after pressing the BREAK key no longer restores a program, since

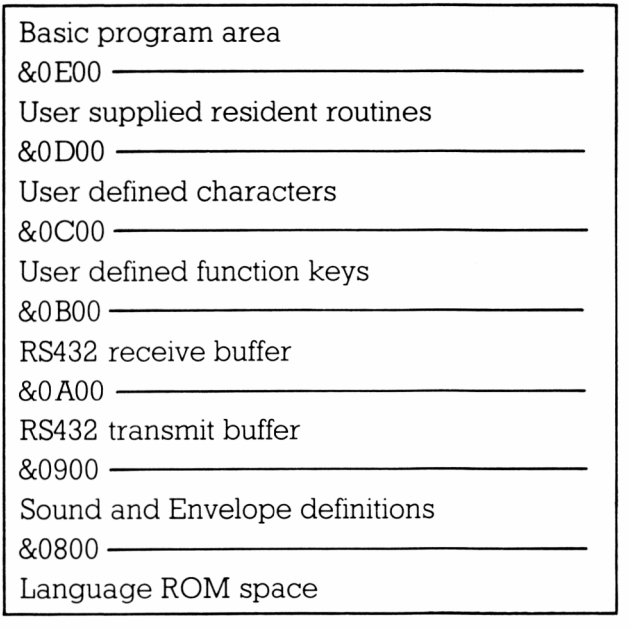
PAGE is reset to its default value and two special characters are written to &0E00 and &0E01, corrupting any code in these locations. If you feel that you need this facility you will have to store the contents of &0E00 and &0E01 in an integer system variable, say G%. Thus, one of the first statements in the program would be G%=!&3584. Then, after pressing BREAK, the following sequence would restore the program:

```
> PAGE=3328 : REM Reset PAGE to the value chosen.
> OLD
>!3584=G% :REM Write the two corrupted bytes back to memory.
> END :REM Sort out the internal Basic pointers.
```

If desired, the BREAK key may be redefined to incorporate these statements:

```
>*KEY 10 PAGE=3328 |M OLD |M !3584=G% |M END |M
```

but then page &0B00 cannot be used for Basic and the maximum saving is only one third of the potential 1½ kbytes.



BEEB DEBUGGING AID

This error-handling routine is for the BBC Micro, and is intended as an aid to debugging (and encourage the user's idleness!). When control is transferred to this routine, it will report the error, then list the line that gave rise to it. The routine has high line

numbers for convenient *SPOOL -*EXEC use, and, since it fills the keyboard buffer using call 138, is suitable only for machines with O.S 1.0 or above. The routine first places the string 'L' into the keyboard buffer (32320, 32330), then converts the two bytes at zero page locations 8 and 9 which, it turns out, contain the value for the function 'ERL', into a string of ASCII numerals which are placed sequentially in the buffer (32340 to 32370). Finally, the code for carriage return is appended (32380), causing the command to be performed.

```
   10 ON ERROR GOTO 32300
   20 PRINT "THIS LINE OK"
   30 *FX0:REM M.O.S. doesn't like remarks
   40 PRINT "...no trailing
   50 A$=SIN(12*(54/1234)
   60 .....etc
   70
   80
   90
32300 REPORT: PRINT"in this:-"
32310 OSBYTE=&FFF4
32320 A%=138:X%=0:Y%=76:CALL OSBYTE
32330 Y%=46:CALL OSBYTE
32340 E$=STR$(!(8) AND&FFFF)
32350 FOR I%=1 TO LEN(E$)
32360   Y%=ASC(MID$(E$,I%,1)):CALL OSBYTE
32370   NEXT
32380 Y%=13:CALL OSBYTE
>
>
>RUN
THIS LINE OK
Bad command in this:-
>L.30
   30 *FX0:REM M.O.S.  doesn't like remarks
>30
>RUN
THIS LINE OK
Missing " in this:-
>L.40
```

```
    40 PRINT "...no trailing
>40
>RUN
THIS LINE OK
Missing ) in this:-
>L.50
    50 A$=SIN(12*(64/1234)
>50
>RUN
THIS LINE OK
Syntax error in this:-
>L.60
    60 .....etc
>
>
```

Available in Century's Science & Technology series

Dictionary of New Information Technology
A. J. MEADOWS, M. GORDON and A. SINGLETON
What to Buy for Business: A Handbook of New Office Technology
JOHN DERRICK and PHILLIP OPPENHEIM
The Electronic Mail Handbook
STEPHEN CONNELL and IAN A. GALBRAITH
The Way the New Technology Works
KEN MARSH
Mind
DAVID A. TAYLOR
Computers and Your Child
RAY HAMMOND
International Dictionary of Graphic Symbols
JOEL ARNSTEIN
In association with 'Personal Computer World'
Microcomputing for Business: A User's Guide
edited by DICK OLNEY
The Microcomputer Handbook: A Buyer's Guide
edited by DICK OLNEY
The Spectrum Handbook
TIM LANGDELL
35 Programs for the Dragon 32
TIM LANGDELL
The Intimate Machine
NEIL FRUDE
35 Educational Programs for the BBC Micro
IAN MURRAY
Educational Programs for the Dragon 32
IAN MURRAY
Educational Programs for the Spectrum
IAN MURRAY
Century Computer Programming Course
PETER MORSE and IAN ADAMSON
Information Technology Yearbook
edited by PHILIP HILLS
Computer Gamesmanship
DAVID LEVY
Best of PCW Software for the Dragon
Best of PCW Software for the Spectrum
The Database Primer
ROSE DEAKIN

CENTURY SOFTWARE VOUCHER

The programs in this book are available on cassette from the publishers.

Please cut out this voucher (do not copy the voucher) and send it together with a cheque/postal order for £5 to:

Dept PCW (BBC),
Century Communications Ltd,
Portland House,
12-13 Greek Street,
London W1.

VOUCHER

Please send me one cassette of the BEST OF PCW SOFTWARE FOR THE BBC MICRO at the special price of £5 including postage and packing.

NAME .....

ADDRESS .....

.....

.....

Please allow 28 days for delivery.

For the past five years, *Personal Computer World* has led the market in microcomputer magazines and has developed a reputation for publishing the very best software for all the most popular micros. Now the cream of these programs has been gathered together in a single volume, enhanced and converted for the BBC Micro.

Along with the forgotten gems from the archives available for the first time for the BBC Micro, this *Best of PCW* series also contains the finest programs written especially for this popular micro, plus a wealth of hints, tips and utilities which will prove essential reading for all serious programmers.

ISBN 0 7126 0266 6
CENTURY PUBLISHING CO. LTD

£5.95