VIC™ GAMES

PLUS BONUS PROGRAMS FOR GRAPHICS DISPLAYS,
SOUND AND MUSIC EFFECTS, AND MORE

NICK HAMPSHIRE

HAYDEN

VIC™ GAMES

NICK HAMPSHIRE

HAYDEN BOOK COMPANY, INC.
Rochelle Park, New Jersey

INTRODUCTION

This book is a mixed collection of games and useful programs demonstrating the capabilities of the VIC-20.

The novice home computer user will find great enjoyment in knowing that all of the programs in this book will work. The more experienced and adventurous user will find the listings of both the games and demonstration programs to be invaluable in his quest to produce his own programs.

In some cases the user will require expansion to his VIC-20 but most of these programs will run on a basic VIC-20.

I have enjoyed compiling this collection and hope that others will gain as much enjoyment using it.

NICK HAMPSHIRE

CONTENTS

Breakout 6
Find the Word.................................... 11
Space Pirates 17
VIC-VIC 22
Birds Demo 25
Rhino 30
Do-Ray-Me 39
Sound Effects 43
Arrow 47
Tank-V-UFO.................................... 52
Landmine 58
Spacewar 65
Joystick Test 72
Define Keys 74
U.S.A. Song 77
Digiclock 82
Leap Frog 85
Rubik's Cube.................................... 92
Boss Puzzle 97
Sketching 1 102
Sketching 2.................................... 106
Kaleidoscope 110
Hi-Res Demo 113
Bandit 116
Moon Lander 122
Circle Demo 127
Hi-Res Plot 130
Hangman 133
Gomoko 141
Supermind 147
Conquest.................................... 153
Hi-Res Aid.................................... 159
Tinymon 163
Racer 170
Car Race 176
Tape Search 180
Appendix to Special Print Characters.............. 183
Index 185

BREAKOUT

DESCRIPTION

This game is the computer version of the popular arcade game Breakout. The program includes some data at the end of the program which is entered into the memory. This machine code routine uses PEEK (197) to check if either of the cursor direction keys have been pressed and moves the bat along the bottom of the screen in the direction corresponding to the key pressed. This routine could be written in Basic but will slow the game to such an extent that it will become very boring.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

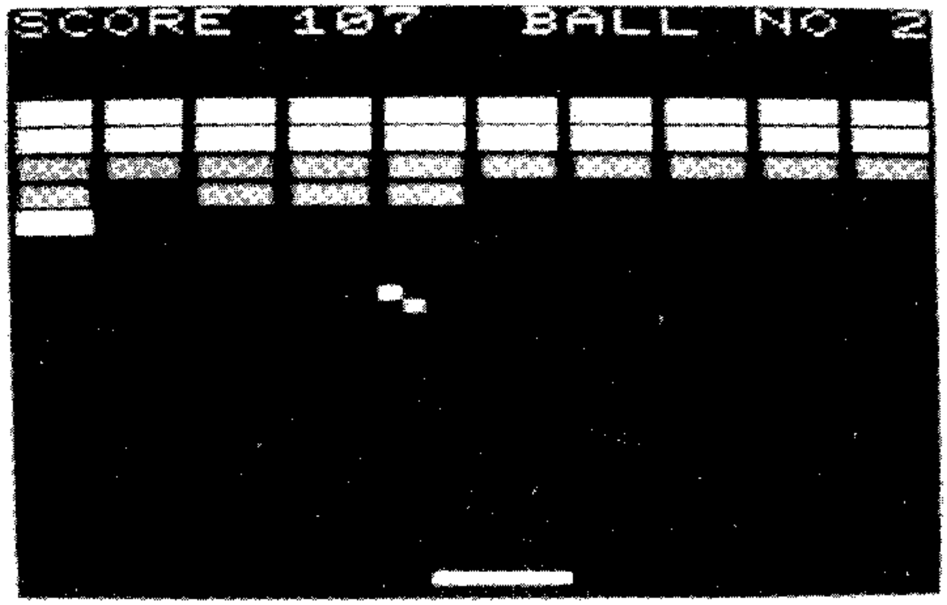
RUNNING THE PROGRAM

After typing run, the program will display a few instructions mainly explaining that the bat is moved using the two cursor keys. Hitting any key will start the game running. There are five balls to start with and the object of the game is to knock out as many of the bricks as possible. When all the bricks have been cleared from the screen, a new set appear and the game is played over again.

PROGRAM STRUCTURE

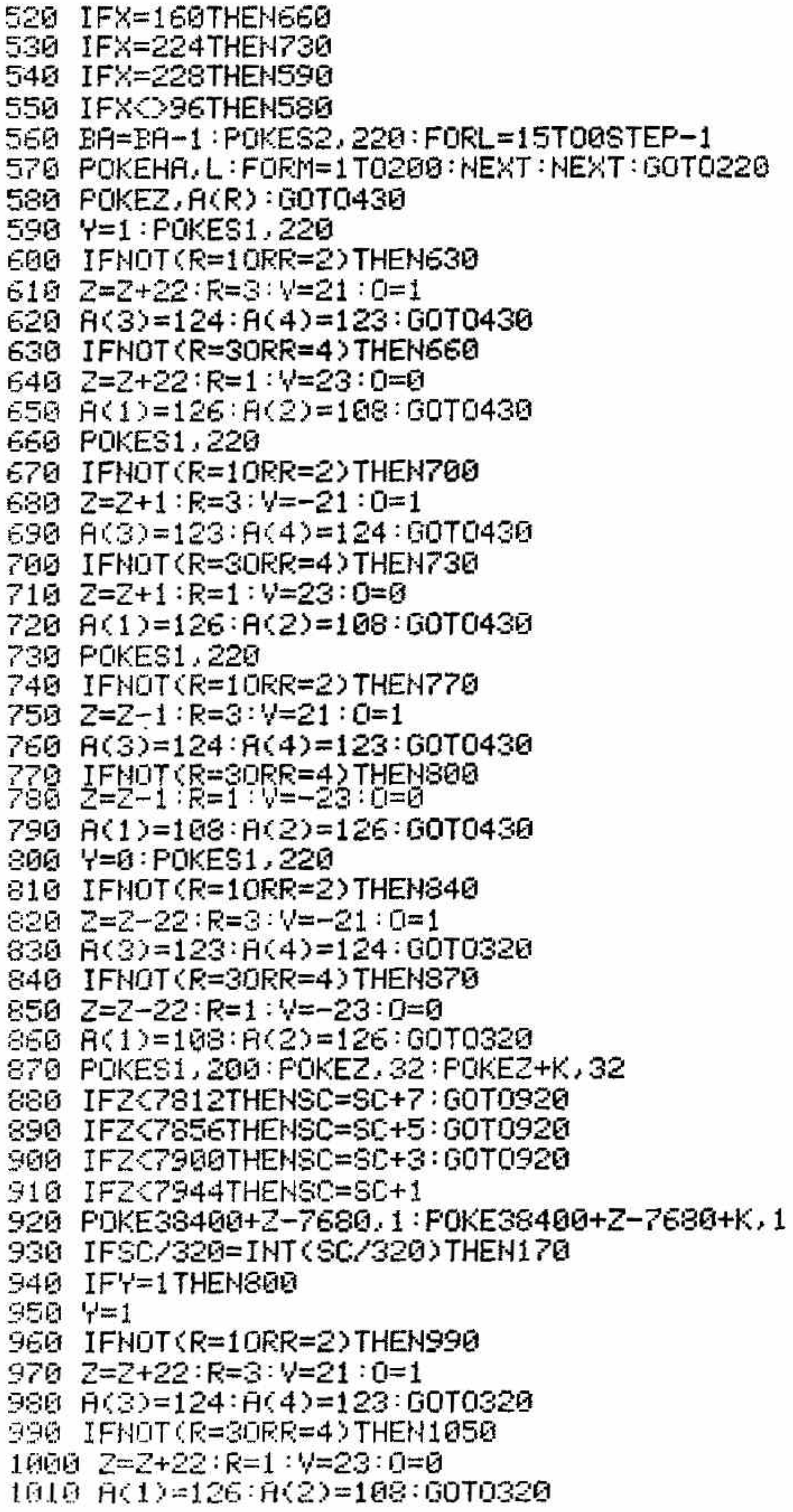
The lines of interest in this program are as follows.

120-230	Set up display.
280-310	Choose random ball start position and direction.
320-340	Display score on top of screen.
350-390	Input for another game.
430	Turn off sound and call M/C routine.
480-550	Check for ball hitting object.
880-910	Check if ball has hit blocks.
1050-1110	Print instructions.
1160-1320	POKE data for machine code into memory.

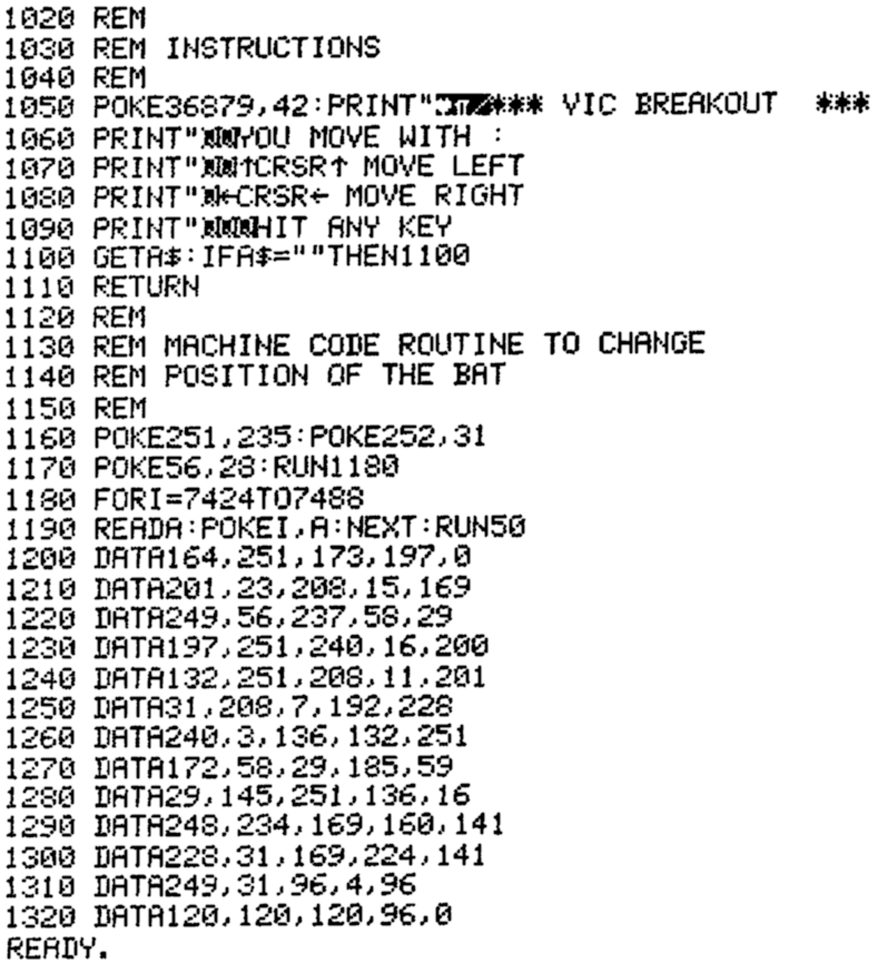


```
SCORE 107  BALL NO 2
```

```
10 REM BREAKOUT
20 REM ***************************************
30 REM
40 GOTO1160
50 HA=36878:S1=36876:S2=36877
60 POKEHA,15
70 GOSUB1050
80 POKE36879,12:P=8:BA=5
90 REM
100 REM DISPLAY
110 REM
120 PRINT"▒█▒ _______________________ ";
130 FORI=1TO21
140 PRINT"▒▒ ▀▒                         ▒▒ ▀▒████ "
150 NEXT
160 PRINT"▒▒ ▀▒                         ▒▒ ▀▒████ ▒"
170 PRINT"▒▒▒▒":A$="▒┬┬┬┬┬┬┬┬┬┬┐"
180 PRINT"▌▒"A$:PRINT"▌▌"A$
190 PRINT"▌▒"A$:PRINT"▌▌"A$
200 PRINT"▌▒"A$:PRINT"▌▌"A$
210 PRINT"▌▒"A$:PRINT"▌▌"A$
220 POKES2,0:POKEHA,15:A(1)=126:A(2)=108
230 A(3)=123:A(4)=124:R=0:O=0:V=23:Y=1
240 REM
250 REM RANDOM BALL START POSITION
260 REM AND DIRECTION
270 REM
280 Z=7945+RND(1)*20
290 G=INT(RND(1)*2)
300 IFG=1THENR=1
310 IFG=0THENR=3
320 POKES1,0:PRINT"▒▒▒▌SCORE"SC
330 PRINT"▒▒▒▌▌▌▌▌▌▌▌▌▌▌▌▌▌BALL NO"BA
340 IF BA<>0 THEN 430
350 PRINT"▒▒ANOTHER GAME ?";
360 GET A$:IF A$="" THEN 360
370 IF A$="Y" THEN RUN
380 IF A$<>"N" THEN 360
390 PRINT"▒▒":POKE 36879,27:END
400 REM
410 REM PLAY GAME AND MOVE BAT WITH SYS 7424
420 REM
430 POKES1,0:SYS7424
440 R=R+1
450 IFO=1THENIFR>4THENR=3:Z=Z+V:POKEZ-V,32:GOTO480
460 IFO=1THEN480
470 IFR>2THENR=1:Z=Z+V:POKEZ-V,32
480 X=PEEK(Z)
490 IFX=120THEN800
500 IFX=207THENK= 1:GOTO870
510 IFX=208THENK=-1:GOTO870
```



```
520 IFX=160THEN660
530 IFX=224THEN730
540 IFX=228THEN590
550 IFX<>96THEN580
560 BA=BA-1:POKES2,220:FORL=15TO0STEP-1
570 POKEHA,L:FORM=1TO200:NEXT:NEXT:GOTO220
580 POKEZ,A(R):GOTO430
590 Y=1:POKES1,220
600 IFNOT(R=1ORR=2)THEN630
610 Z=Z+22:R=3:V=21:O=1
620 A(3)=124:A(4)=123:GOTO430
630 IFNOT(R=3ORR=4)THEN660
640 Z=Z+22:R=1:V=23:O=0
650 A(1)=126:A(2)=108:GOTO430
660 POKES1,220
670 IFNOT(R=1ORR=2)THEN700
680 Z=Z+1:R=3:V=-21:O=1
690 A(3)=123:A(4)=124:GOTO430
700 IFNOT(R=3ORR=4)THEN730
710 Z=Z+1:R=1:V=23:O=0
720 A(1)=126:A(2)=108:GOTO430
730 POKES1,220
740 IFNOT(R=1ORR=2)THEN770
750 Z=Z-1:R=3:V=21:O=1
760 A(3)=124:A(4)=123:GOTO430
770 IFNOT(R=3ORR=4)THEN800
780 Z=Z-1:R=1:V=-23:O=0
790 A(1)=108:A(2)=126:GOTO430
800 Y=0:POKES1,220
810 IFNOT(R=1ORR=2)THEN840
820 Z=Z-22:R=3:V=-21:O=1
830 A(3)=123:A(4)=124:GOTO320
840 IFNOT(R=3ORR=4)THEN870
850 Z=Z-22:R=1:V=-23:O=0
860 A(1)=108:A(2)=126:GOTO320
870 POKES1,200:POKEZ,32:POKEZ+K,32
880 IFZ<7812THENSC=SC+7:GOTO920
890 IFZ<7856THENSC=SC+5:GOTO920
900 IFZ<7900THENSC=SC+3:GOTO920
910 IFZ<7944THENSC=SC+1
920 POKE38400+Z-7680,1:POKE38400+Z-7680+K,1
930 IFSC/320=INT(SC/320)THEN170
940 IFY=1THEN800
950 Y=1
960 IFNOT(R=1ORR=2)THEN990
970 Z=Z+22:R=3:V=21:O=1
980 A(3)=124:A(4)=123:GOTO320
990 IFNOT(R=3ORR=4)THEN1050
1000 Z=Z+22:R=1:V=23:O=0
1010 A(1)=126:A(2)=108:GOTO320
```



```
1020 REM
1030 REM INSTRUCTIONS
1040 REM
1050 POKE36879,42:PRINT"■■■*** VIC BREAKOUT  ***
1060 PRINT"■■YOU MOVE WITH :
1070 PRINT"■■↑CRSR↑ MOVE LEFT
1080 PRINT"■←CRSR← MOVE RIGHT
1090 PRINT"■■■HIT ANY KEY
1100 GETA$:IFA$=""THEN1100
1110 RETURN
1120 REM
1130 REM MACHINE CODE ROUTINE TO CHANGE
1140 REM POSITION OF THE BAT
1150 REM
1160 POKE251,235:POKE252,31
1170 POKE56,28:RUN1180
1180 FORI=7424TO7488
1190 READA:POKEI,A:NEXT:RUN50
1200 DATA164,251,173,197,0
1210 DATA201,23,208,15,169
1220 DATA249,56,237,58,29
1230 DATA197,251,240,16,200
1240 DATA132,251,208,11,201
1250 DATA31,208,7,192,228
1260 DATA240,3,136,132,251
1270 DATA172,58,29,185,59
1280 DATA29,145,251,136,16
1290 DATA248,234,169,160,141
1300 DATA228,31,169,224,141
1310 DATA249,31,96,4,96
1320 DATA120,120,120,96,0
READY.
```

FIND THE WORD

DESCRIPTION

Find The Word allows the user to input a word length, the computer will choose a word from the data statements of that length and put it in a matrix with a load of random letters. The user then has to find that word in the matrix. The word may be horizontal, vertical, or diagonal and may be backwards. If the word has not been found after three guesses, the computer will give clues by telling the user the first letter of the word, the second, etc until it can give no more clues without giving the whole word.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

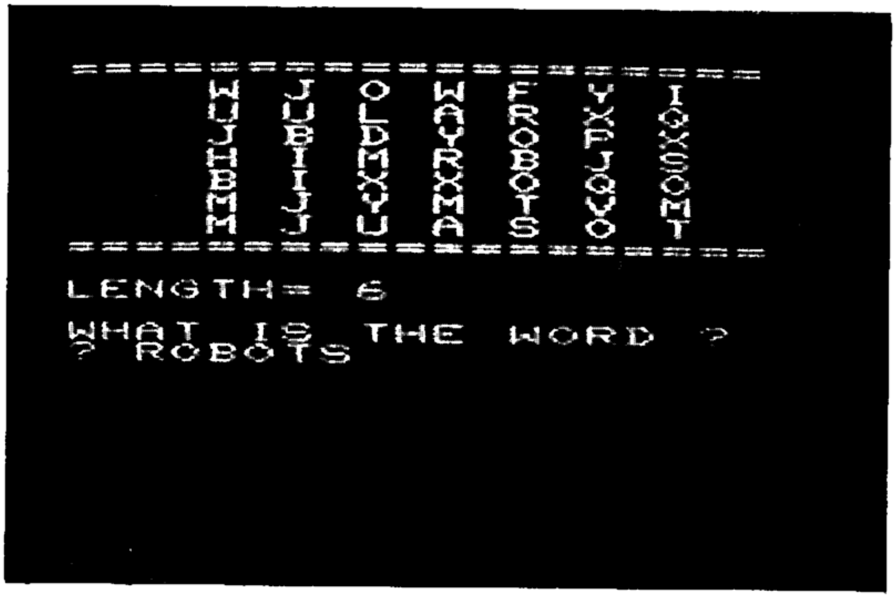
After typing run, the program will ask if instructions are required, if the answer is yes then a page of instructions will be printed on the screen. The game will then start by asking how many letters in the word. The variable M chooses whether the word is diagonal or not. Then the guessing of the word begins. When the word has been guessed or the computer has run out of clues the score for that word will be printed along with the running total.

PROGRAM STRUCTURE

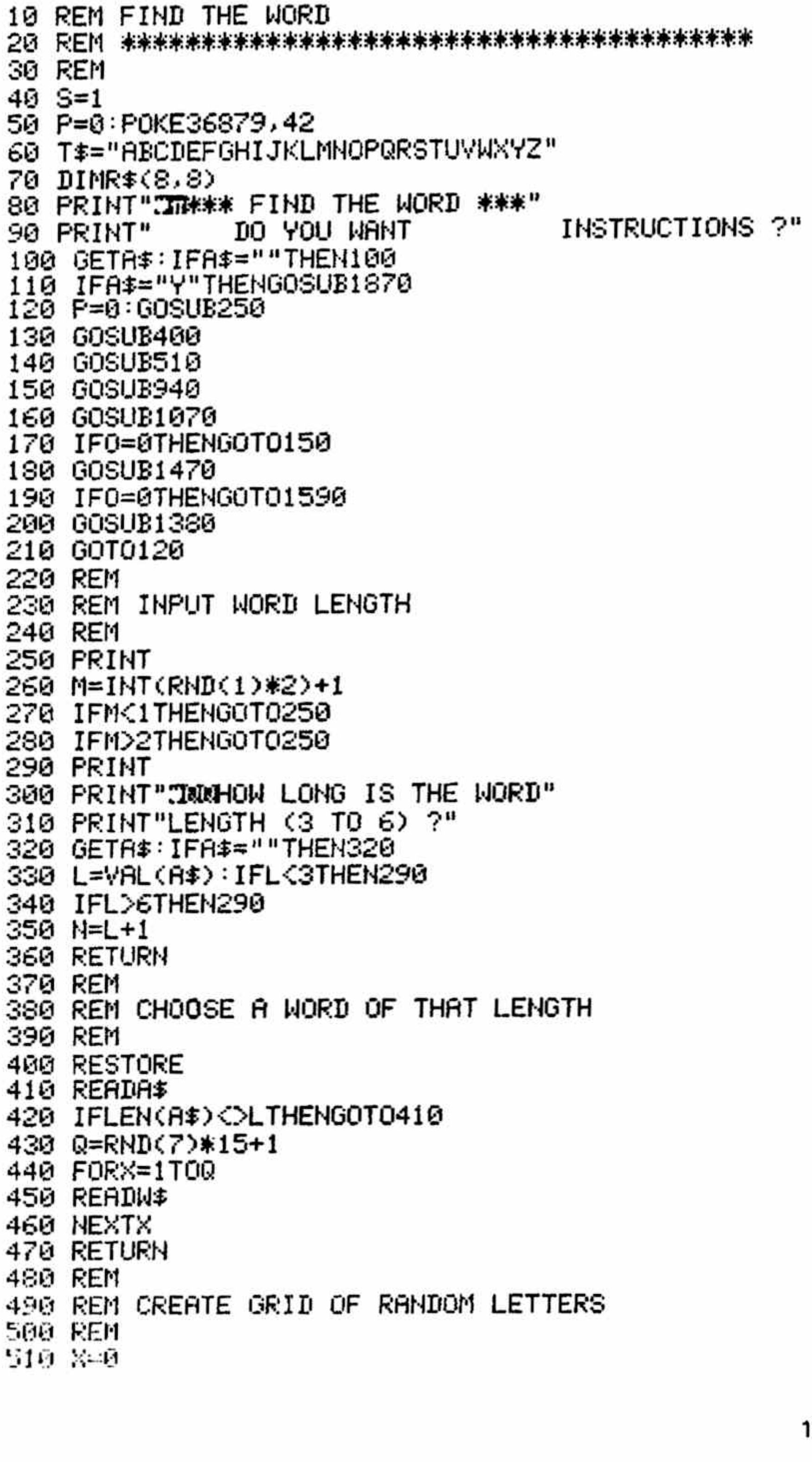
The lines of interest in this program are as follows.

250-350 Input length of word.
400-470 Choose a word of that length.
510-900 Create grid and put the word somewhere in the grid.
940-1030 Print the grid on the screen.
1070-1420 Guess at word and check to see if the guess is correct.
1470-1590 Guessed the word or no more clues available.
1630-1640 Data for three letter words.

1680-1700 Data for four letter words.
1740-1760 Data for five letter words.
1800-1830 Data for six letter words.
1870-1940 Print instructions.



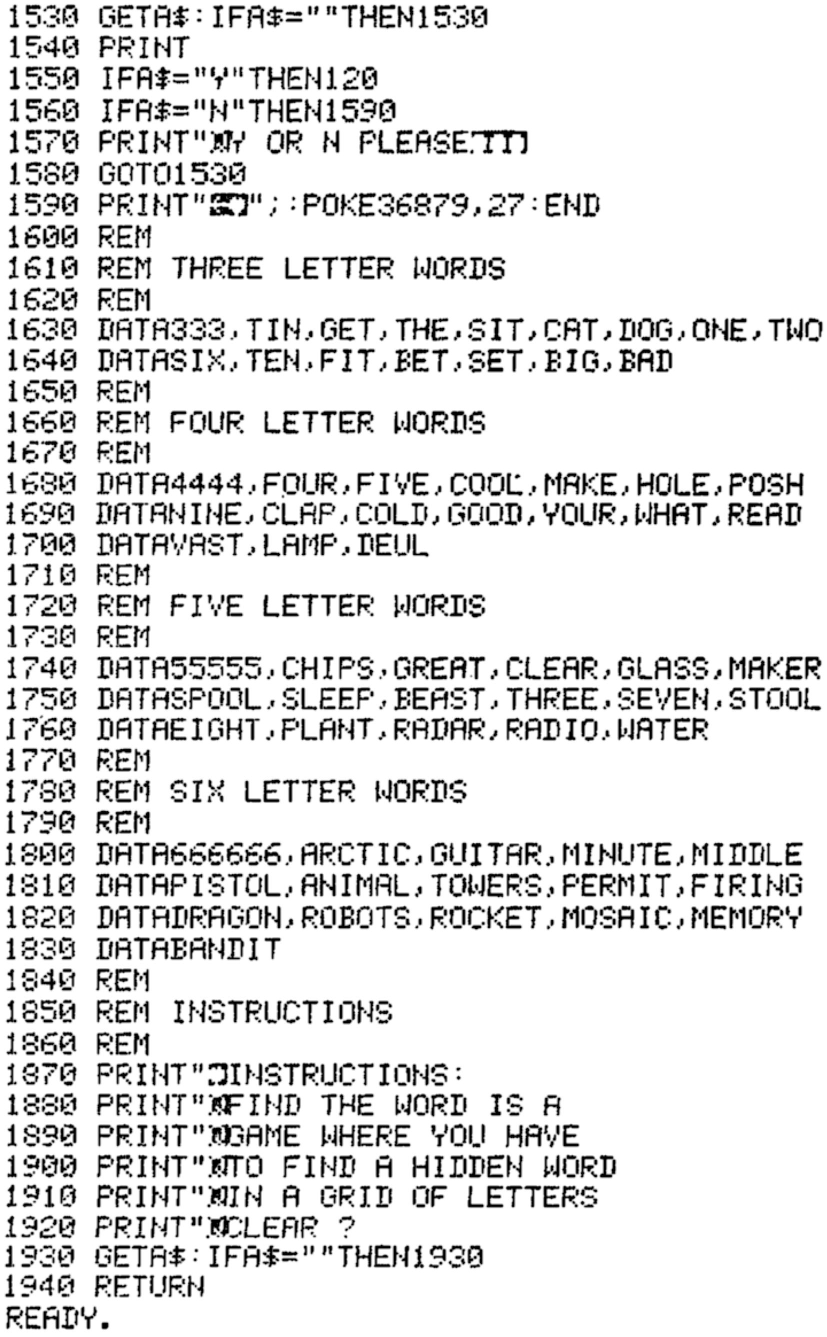
```
===================
     W J O W F Y I
     U U L A R X Q
     J B D Y O F X
     H I M R B J S
     B I X X O Q O
     M J Y M T V M
     M J U A S O T
===================
LENGTH= 6
WHAT IS THE WORD ?
? ROBOTS
```



```
10 REM FIND THE WORD
20 REM **************************************
30 REM
40 S=1
50 P=0:POKE36879,42
60 T$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
70 DIMR$(8,8)
80 PRINT"♥Q*** FIND THE WORD ***"
90 PRINT"     DO YOU WANT         INSTRUCTIONS ?"
100 GETA$:IFA$=""THEN100
110 IFA$="Y"THENGOSUB1870
120 P=0:GOSUB250
130 GOSUB400
140 GOSUB510
150 GOSUB940
160 GOSUB1070
170 IFO=0THENGOTO150
180 GOSUB1470
190 IFO=0THENGOTO1590
200 GOSUB1380
210 GOTO120
220 REM
230 REM INPUT WORD LENGTH
240 REM
250 PRINT
260 M=INT(RND(1)*2)+1
270 IFM<1THENGOTO250
280 IFM>2THENGOTO250
290 PRINT
300 PRINT"♥QQHOW LONG IS THE WORD"
310 PRINT"LENGTH (3 TO 6) ?"
320 GETA$:IFA$=""THEN320
330 L=VAL(A$):IFL<3THEN290
340 IFL>6THEN290
350 N=L+1
360 RETURN
370 REM
380 REM CHOOSE A WORD OF THAT LENGTH
390 REM
400 RESTORE
410 READA$
420 IFLEN(A$)<>LTHENGOTO410
430 Q=RND(7)*15+1
440 FORX=1TOQ
450 READW$
460 NEXTX
470 RETURN
480 REM
490 REM CREATE GRID OF RANDOM LETTERS
500 REM
510 X=0
```

```
520 R=INT(RND(1)*N+1)
530 K=INT(RND(1)*N+1)
540 IFR<=2ORR>=N-1THENGOTO580
550 IFK<=2ORK>=N-1THENGOTO580
560 X=X+1
570 GOTO520
580 FORX=1TON
590 FORY=1TON
600 Q=INT(RND(1)*26+1)
610 Q$=MID$(T$,Q,1)
620 R$(X,Y)=Q$
630 NEXTY
640 NEXTX
650 IFM=2THENGOTO810
660 IFR<=2ORR>=N-1THENGOTO740
670 IFK<=2THENKR=1
680 IFK>=N-1THENKR=-1
690 FORX=1TOLEN(W$)
700 R$(K,R)=MID$(W$,X,1)
710 K=K+KR
720 NEXTX
730 RETURN
740 IFR<=2THENRR=1
750 IFR>=N-1THENRR=-1
760 FORX=1TOLEN(W$)
770 R$(K,R)=MID$(W$,X,1)
780 R=R+RR
790 NEXTX
800 RETURN
810 IFR<=2THENRR=1
820 IFR>=N-1THENRR=-1
830 IFK<=2THENKR=1
840 IFK>=N-1THENKR=-1
850 FORX=1TOLEN(W$)
860 R$(K,R)=MID$(W$,X,1)
870 K=K+KR
880 R=R+RR
890 NEXTX
900 RETURN
910 REM
920 REM PRINT GRID ON SCREEN
930 REM
940 PRINT"⊐╪=====================
950 FORX=1TON:PRINT"     ";
960 FORY=1TON
970 PRINTR$(X,Y);" ";
980 NEXTY
990 PRINT
1000 NEXTX
1010 PRINT"=====================π
1020 PRINT"▪LENGTH=";L
```

```
1030 RETURN
1040 REM
1050 REM GUESS AT WORD
1060 REM
1070 PRINT"▒WHAT IS THE WORD ?
1080 INPUTA$
1090 P=P+1
1100 IFA$=W$THENGOTO1300
1110 PRINT"▒SORRY ▒▒WRONG▒▒"
1120 IF(P-2)=LTHENGOTO1240
1130 IFP>2THENGOTO1170
1140 PRINT"▒NOT VERY GOOD"
1150 O=0:FORKK=1TO3000:NEXT
1160 RETURN
1170 H$=LEFT$(W$,P-2)
1180 PRINT"▒YOU NEED HELP."
1190 IFP>3THEN1220
1200 PRINT"▒THE FIRST LETTER IS":PRINTH$
1210 GOTO1150
1220 PRINT"▒THE FIRST";P-2;"LETTERS":PRINT"ARE  ";H$
1230 GOTO1150
1240 PRINT"▒WRONG, YOU DID":PRINT"NOT GET THE WORD
1250 PRINT"RIGHT.
1260 PRINT"▒IT WAS ▒▒";W$"▒
1270 FORI=1TO3000:NEXT
1280 O=-1
1290 RETURN
1300 PRINT
1310 PRINT"▒▒THAT IS IT
1320 PRINT"▒YOU GOT IT IN"P
1330 PRINT"TRY"
1340 IFP>1THENPRINT"]TRIES"
1350 FORI=1TO3000:NEXT
1360 O=1
1370 RETURN
1380 S=S+1
1390 RR=0
1400 KR=0
1410 P=0
1420 RETURN
1430 REM
1440 REM GUESSED THE WORD OR NO MORE
1450 REM CLUES AVAILABLE
1460 REM
1470 PRINT
1480 T=T+(L/P)*(L/P)*M
1485 Z=INT((L/P)*(L/P)*M)
1490 PRINT"]THE SCORE FOR THIS":PRINT"WORD:";Z
1500 PRINT"▒(MAXIMUM SCORE":PRINT"WAS";INT(L*L*M);"▌)"
1510 PRINT"▒TOTAL SCORE"INT(T/S)
1520 PRINT"▒▒ANOTHER GO ?
```



```
1530 GETA$:IFA$=""THEN1530
1540 PRINT
1550 IFA$="Y"THEN120
1560 IFA$="N"THEN1590
1570 PRINT"▒Y OR N PLEASE.TT]
1580 GOTO1530
1590 PRINT"▒]";:POKE36879,27:END
1600 REM
1610 REM THREE LETTER WORDS
1620 REM
1630 DATA333,TIN,GET,THE,SIT,CAT,DOG,ONE,TWO
1640 DATASIX,TEN,FIT,BET,SET,BIG,BAD
1650 REM
1660 REM FOUR LETTER WORDS
1670 REM
1680 DATA4444,FOUR,FIVE,COOL,MAKE,HOLE,POSH
1690 DATANINE,CLAP,COLD,GOOD,YOUR,WHAT,READ
1700 DATAVAST,LAMP,DEUL
1710 REM
1720 REM FIVE LETTER WORDS
1730 REM
1740 DATA55555,CHIPS,GREAT,CLEAR,GLASS,MAKER
1750 DATASPOOL,SLEEP,BEAST,THREE,SEVEN,STOOL
1760 DATAEIGHT,PLANT,RADAR,RADIO,WATER
1770 REM
1780 REM SIX LETTER WORDS
1790 REM
1800 DATA666666,ARCTIC,GUITAR,MINUTE,MIDDLE
1810 DATAPISTOL,ANIMAL,TOWERS,PERMIT,FIRING
1820 DATADRAGON,ROBOTS,ROCKET,MOSAIC,MEMORY
1830 DATABANDIT
1840 REM
1850 REM INSTRUCTIONS
1860 REM
1870 PRINT"]INSTRUCTIONS:
1880 PRINT"▒FIND THE WORD IS A
1890 PRINT"▒GAME WHERE YOU HAVE
1900 PRINT"▒TO FIND A HIDDEN WORD
1910 PRINT"▒IN A GRID OF LETTERS
1920 PRINT"▒CLEAR ?
1930 GETA$:IFA$=""THEN1930
1940 RETURN
READY.
```

SPACE PIRATES

DESCRIPTION

In this game, the player is in control of three lazer gun bases at the bottom of the screen. The three bases are fired with the use of the keys A, S, D respectively. The space pirates go across the screen in groups of five and the player has to shoot them out using the lazer guns. If the space pirates get past the bases, they start to destroy the cells on the right hand side of the screen. When they have passed through the cells they appear on the left side again but lower down the screen. When the space pirates have all been annihilated, another set of pirates appear but in a different form. The game is over when all of the cells have been destroyed.
This program makes good use of sound and user definable characters.

EQUIPMENT REQUIRED

This program works on an unexpanded VIC-20 but will not work with expansion.

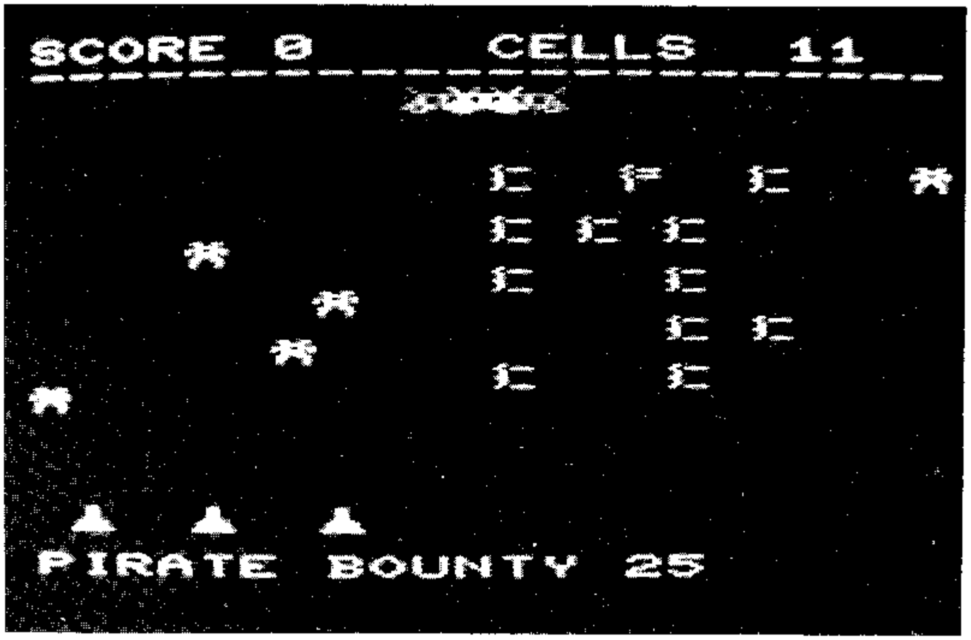
RUNNING THE PROGRAM

After typing run, the program will display which bases are fired by which keys and will pause while the character generator is being moved. The game will then start.

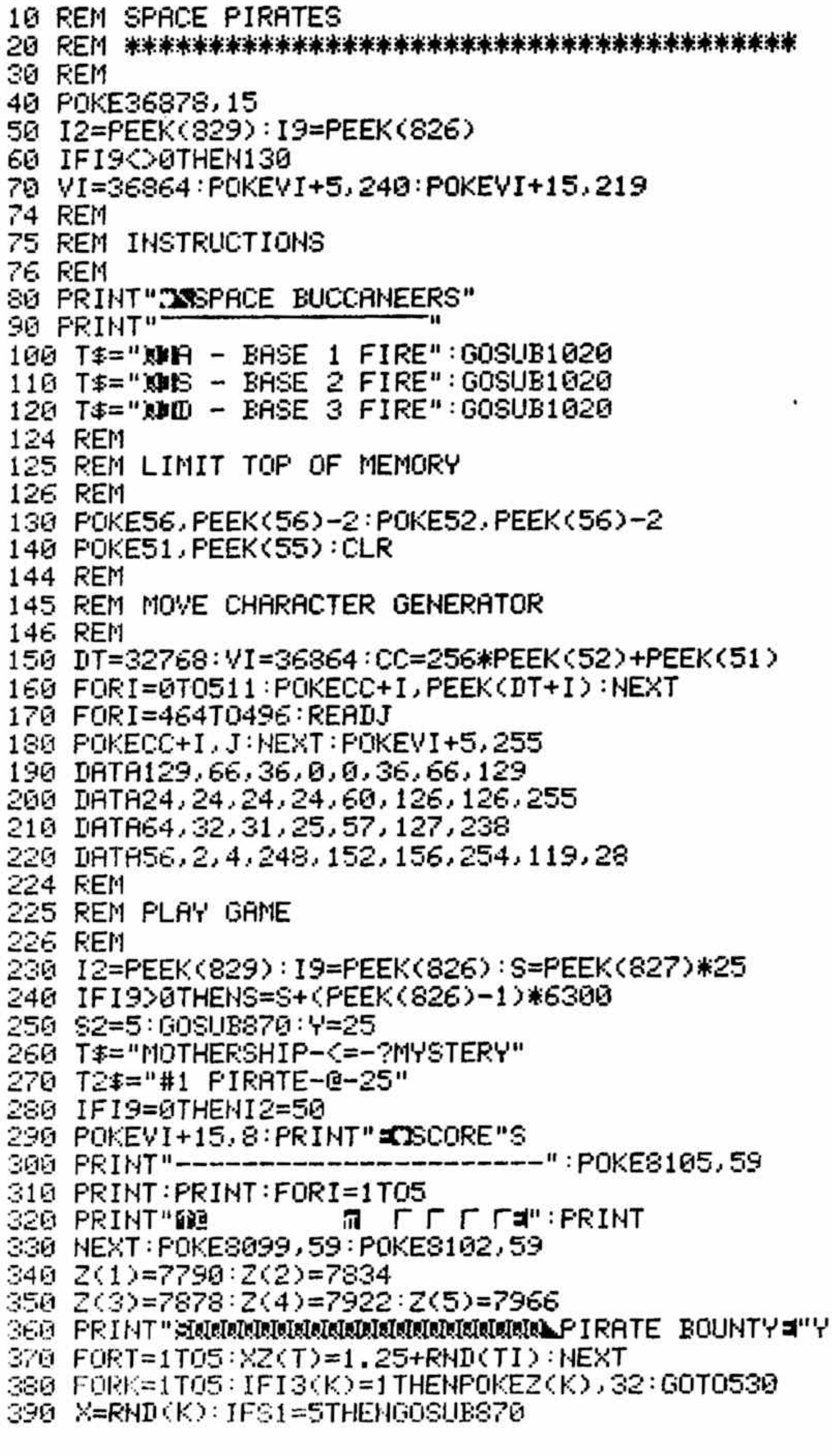
PROGRAM STRUCTURE

Lines of interest in this program are as follows.

80-120 Display instructions.
130-140 Limit top of memory.
150-220 Move character generator and set up other characters.
230-570 Play the game.
580-620 Input from keyboard.
880-960 Change the character for the space pirates.
1040-1180 Game over, play again ?

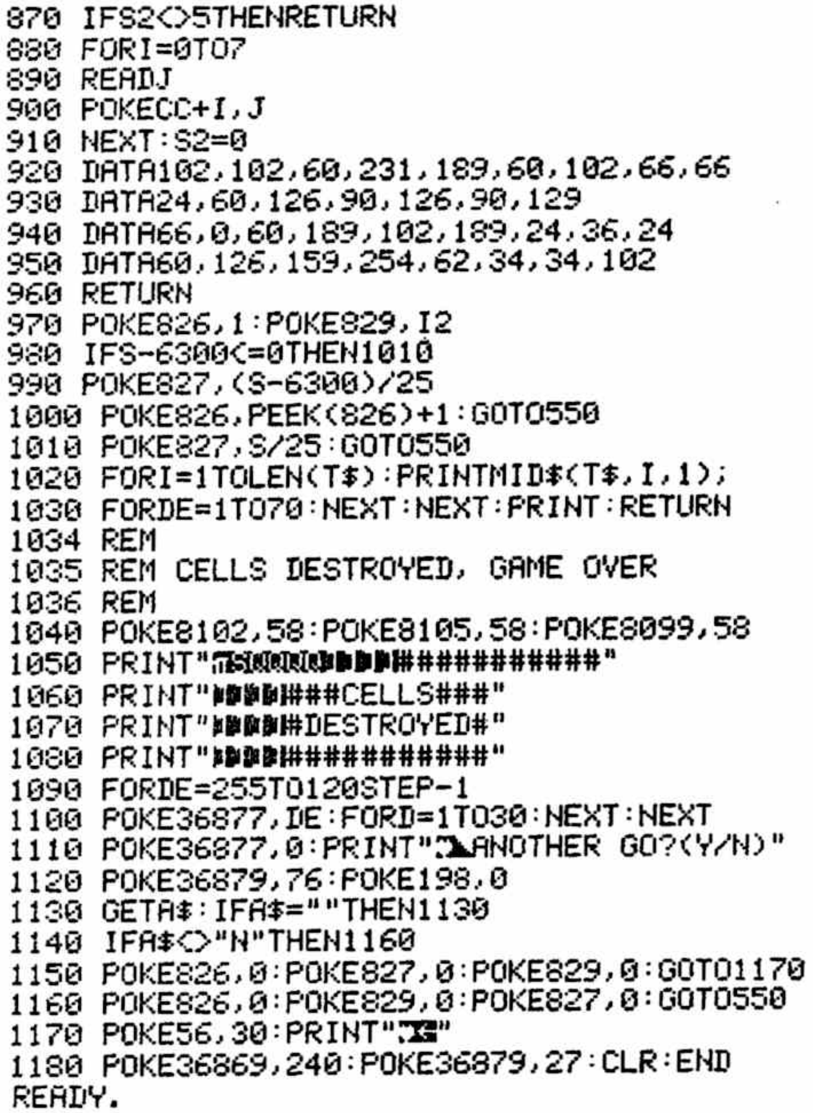


SCORE 0
CELLS 11
PIRATE BOUNTY 25



```
10 REM SPACE PIRATES
20 REM ****************************************
30 REM
40 POKE36878,15
50 I2=PEEK(829):I9=PEEK(826)
60 IFI9<>0THEN130
70 VI=36864:POKEVI+5,240:POKEVI+15,219
74 REM
75 REM INSTRUCTIONS
76 REM
80 PRINT"▓▓SPACE BUCCANEERS"
90 PRINT"▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔"
100 T$="▓▓A - BASE 1 FIRE":GOSUB1020
110 T$="▓▓S - BASE 2 FIRE":GOSUB1020
120 T$="▓▓D - BASE 3 FIRE":GOSUB1020
124 REM
125 REM LIMIT TOP OF MEMORY
126 REM
130 POKE56,PEEK(56)-2:POKE52,PEEK(56)-2
140 POKE51,PEEK(55):CLR
144 REM
145 REM MOVE CHARACTER GENERATOR
146 REM
150 DT=32768:VI=36864:CC=256*PEEK(52)+PEEK(51)
160 FORI=0TO511:POKECC+I,PEEK(DT+I):NEXT
170 FORI=464TO496:READJ
180 POKECC+I,J:NEXT:POKEVI+5,255
190 DATA129,66,36,0,0,36,66,129
200 DATA24,24,24,24,60,126,126,255
210 DATA64,32,31,25,57,127,238
220 DATA56,2,4,248,152,156,254,119,28
224 REM
225 REM PLAY GAME
226 REM
230 I2=PEEK(829):I9=PEEK(826):S=PEEK(827)*25
240 IFI9>0THENS=S+(PEEK(826)-1)*6300
250 S2=5:GOSUB870:Y=25
260 T$="MOTHERSHIP-<=-?MYSTERY"
270 T2$="#1 PIRATE-@-25"
280 IFI9=0THENI2=50
290 POKEVI+15,8:PRINT"▓▓SCORE"S
300 PRINT"----------------------":POKE8105,59
310 PRINT:PRINT:FORI=1TO5
320 PRINT"▓▓           ▓  ┌ ┌ ┌ ┌▓":PRINT
330 NEXT:POKE8099,59:POKE8102,59
340 Z(1)=7790:Z(2)=7834
350 Z(3)=7878:Z(4)=7922:Z(5)=7966
360 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓PIRATE BOUNTY▓"Y
370 FORT=1TO5:XZ(T)=1.25+RND(TI):NEXT
380 FORK=1TO5:IFI3(K)=1THENPOKEZ(K),32:GOTO530
390 X=RND(K):IFS1=5THENGOSUB870
```

```
400 IFI4=0ANDRND(4)<.02THENI4=1:BU=7724
410 IFI4<>1THEN430
420 POKEBU,32:BU=BU+1:POKEBU+1,61:POKEBU,60
430 IFNOT(I4=1ANDBU>=7743)THEN450
440 I4=0:POKEBU,32:POKEBU+1,32
450 IFX<=.5THEN480
460 POKEZ(K),32:Z(K)=Z(K)+XZ(K)
470 POKEZ(K),0:POKEZ(K)+30720,5
480 GOSUB580
490 IFPEEK(Z(K)+1)<>79THEN510
500 I2=I2-1:PRINT"S▮▮▮▮▮▮▮▮▮▮▮▮CELLS "I2"▮ "
510 IFI2<=0THEN1040
520 IFZ(K)>=8098THEN1040
530 NEXT:GETA$:IFA$="Q"THEN1170
540 GOTO380
550 POKE36869,240:POKE56,30:POKE52,30
560 CLR:PRINT"JQ▮SCANNING SECTOR"
570 PRINT"Q▮STARDATE "TI$:RUN
574 REM
575 REM INPUT FROM KEYBOARD
576 REM
580 Z=PEEK(197)
590 IFZ=17THENC=8077:GOSUB630
600 IFZ=41THENC=8080:GOSUB630
610 IFZ=18THENC=8083:GOSUB630
620 RETURN
630 MM=32:FORI=C-22TOC-330STEP-22
640 POKEI+22,MM:IFPEEK(I-22)=0THEN710
650 MM=PEEK(I-22):IFMM=61ORMM=60THENGOSUB820
660 POKEI,33
670 IFBU=I-22THENPOKEI,32:I=C-330:GOTO700
680 FORDE=253TO250STEP-1
690 POKE36875,DE:NEXT:POKE36875,0
700 NEXT:POKEI+22,32:RETURN
710 POKE36875,0:POKEI,32
720 POKEI-22,58:POKEI+30698,2
730 FORDE=200TO255:POKE36877,DE
740 NEXT:POKE36877,0:POKEI-22,32
750 S=S+Y:S1=S1+1:PRINT"SSCORE"S:I=C-352
760 I3(K)=1:S2=S2+1:IFS>=12000THENS=S-1200
770 IFS1/20=INT(S1/20)ANDS1<>0THEN970
780 IFS2<>5THEN810
790 GOSUB870:Y=Y+25:FORJ=1TO5
800 I3(J)=0:NEXT:GOTO290
810 GOTO680
820 V=INT(RND(7)*4+1)*50:POKEBU+1,32:POKEBU,58
830 FORDE=150TO180:POKE36877,DE
840 NEXT:POKE36877,0:I4=0:POKEBU,32
850 S=S+V:PRINT"SSCORE"S:POKEI,32:RETURN
860 IFS1/20=INT(S1/20)ANDS1<>0THEN970
```



```
870 IFS2<>5THENRETURN
880 FORI=0TO7
890 READJ
900 POKECC+I,J
910 NEXT:S2=0
920 DATA102,102,60,231,189,60,102,66,66
930 DATA24,60,126,90,126,90,129
940 DATA66,0,60,189,102,189,24,36,24
950 DATA60,126,159,254,62,34,34,102
960 RETURN
970 POKE826,1:POKE829,I2
980 IFS-6300<=0THEN1010
990 POKE827,(S-6300)/25
1000 POKE826,PEEK(826)+1:GOTO550
1010 POKE827,S/25:GOTO550
1020 FORI=1TOLEN(T$):PRINTMID$(T$,I,1);
1030 FORDE=1TO70:NEXT:NEXT:PRINT:RETURN
1034 REM
1035 REM CELLS DESTROYED, GAME OVER
1036 REM
1040 POKE8102,58:POKE8105,58:POKE8099,58
1050 PRINT"▮▮▮▮▮▮▮▮▮############"
1060 PRINT"▮▮▮▮###CELLS###"
1070 PRINT"▮▮▮▮#DESTROYED#"
1080 PRINT"▮▮▮▮############"
1090 FORDE=255TO120STEP-1
1100 POKE36877,DE:FORD=1TO30:NEXT:NEXT
1110 POKE36877,0:PRINT"▮▮ANOTHER GO?(Y/N)"
1120 POKE36879,76:POKE198,0
1130 GETA$:IFA$=""THEN1130
1140 IFA$<>"N"THEN1160
1150 POKE826,0:POKE827,0:POKE829,0:GOTO1170
1160 POKE826,0:POKE829,0:POKE827,0:GOTO550
1170 POKE56,30:PRINT"▮▮"
1180 POKE36869,240:POKE36879,27:CLR:END
READY.
```

VIC VIC

DESCRIPTION

VIC VIC program demonstrates how the display can be manipulated to create interesting effects by changing some of the control registers. First printing a grid of colours and then by manipulating the control registers the display is 'pulled' into the centre of the screen and then back out. A coloured VIC appears on the screen and the same thing happens. This will continue until stop/reset is hit.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

Type run and watch the effects produced by the program. If the program is stopped, the screen will stay as it was when stopped until stop/reset is pressed.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

40-150	Print coloured grid on the screen.
160-330	Print large 'VIC' on the screen.
340	Go back and repeat.
360-430	Manipulate the registers for the coloured grid.
440-510	Manipulate the registers for the large 'VIC'.

VIC

```
10 REM VIC VIC
20 REM ****************************************
30 REM
40 PRINT"♥";:FORI=1TO5
50 PRINT"▚▆     ▐    ◩    ◣    ▒     ";:NEXTI
60 FORI=1TO4
70 PRINT"▚▒     ▛    ◪    ▥    ▪     ";:NEXTI
80 FORI=1TO5
90 PRINT"▚▆     ▐    ◩    ◣    ▒     ";:NEXTI
100 FORI=1TO4
110 PRINT"▚▒     ▛    ◪    ▥    ▪     ";:NEXTI
120 FORI=1TO4
130 PRINT"▚▆     ▐    ◩    ◣    ▒     ";:NEXTI
140 PRINT"▚▆     ▐    ◩    ◣    ▒    ▒";:GOSUB360
150 PRINT"♥▚▐                      ";
160 FORI=0TO4
170 PRINT"▚▐                      ";:NEXTI
180 PRINT"▚▚  ▐    ◩  ▐  ◪  ▐    ▥    ▐  ";
190 PRINT"▚▚  ▐    ◩  ▐  ◪  ▐   ▥       ▐ ";
200 PRINT"▚▚  ▐    ◩  ▐  ◪  ▐  ▥  ▐    ▥  ";
210 PRINT"▚▚  ▐    ◩  ▐  ◪  ▐  ▥  ▐    ▥  ";
220 PRINT"▚▐ ▚  ▐  ◩  ▐   ◪  ▐  ▥  ▐      ";
230 PRINT"▚▐ ▚  ▐  ◩  ▐   ◪  ▐  ▥  ▐      ";
240 PRINT"▚▐ ▚  ▐  ◩  ▐   ◪  ▐  ▥  ▐      ";
250 PRINT"▚▐ ▚  ▐  ◩  ▐   ◪  ▐  ▥  ▐      ";
260 PRINT"▚▐  ▚    ▐    ◪  ▐  ▥  ▐    ▥  ";
270 PRINT"▚▐  ▚    ▐    ◪  ▐  ▥  ▐    ▥  ";
280 PRINT"▚▐  ▚    ▐    ◪  ▐   ▥      ▐ ";
290 PRINT"▚▐   ▚  ▐     ◪  ▐    ▥    ▐  ";
300 FORI=0TO3
310 PRINT"▚▐                      ";:NEXTI
320 PRINT"▚▐                     ▒";
330 GOSUB440
340 GOTO40
350 END
360 FORI=1TO22:POKE36866,128+I:POKE36867,2+2*I
370 POKE36864,34-I:POKE36865,82-2*I:NEXTI
380 FORI=0TO255:NEXTI:R=INT(RND(1)*7)
390 POKE36879,25+R:FORI=0TO255:NEXTI
400 FORI=22TO1STEP-1:POKE36866,128+I
410 POKE36867,2+2*I
420 POKE36864,34-I:POKE36865,82-2*I
430 NEXTI:RETURN
440 FORI=1TO22:POKE36866,128+I:POKE36867,2+2*I
450 POKE36864,34-I:POKE36865,82-2*I:NEXTI
460 FORI=0TO999:NEXTI:R=INT(RND(1)*7)
470 POKE36879,25+R:FORI=0TO999:NEXTI
480 FORI=22TO1STEP-1:POKE36866,128+I
490 POKE36867,2+2*I
500 POKE36864,34-I:POKE36865,82-2*I
510 NEXTI:RETURN
READY.
```

BIRDS DEMO

DESCRIPTION

This program is a demonstration of user defined characters and real time animation. Displayed on the screen is a sea view—a cloud in the top right hand corner and sea with waves on the lower half of the screen. Attractively coloured birds are soaring over the sea.
As the display stops moving and a tune begins to play, a missile is fired from under the sea and shoots down a bird before beginning all over again.

EQUIPMENT REQUIRED

Basic VIC-20.

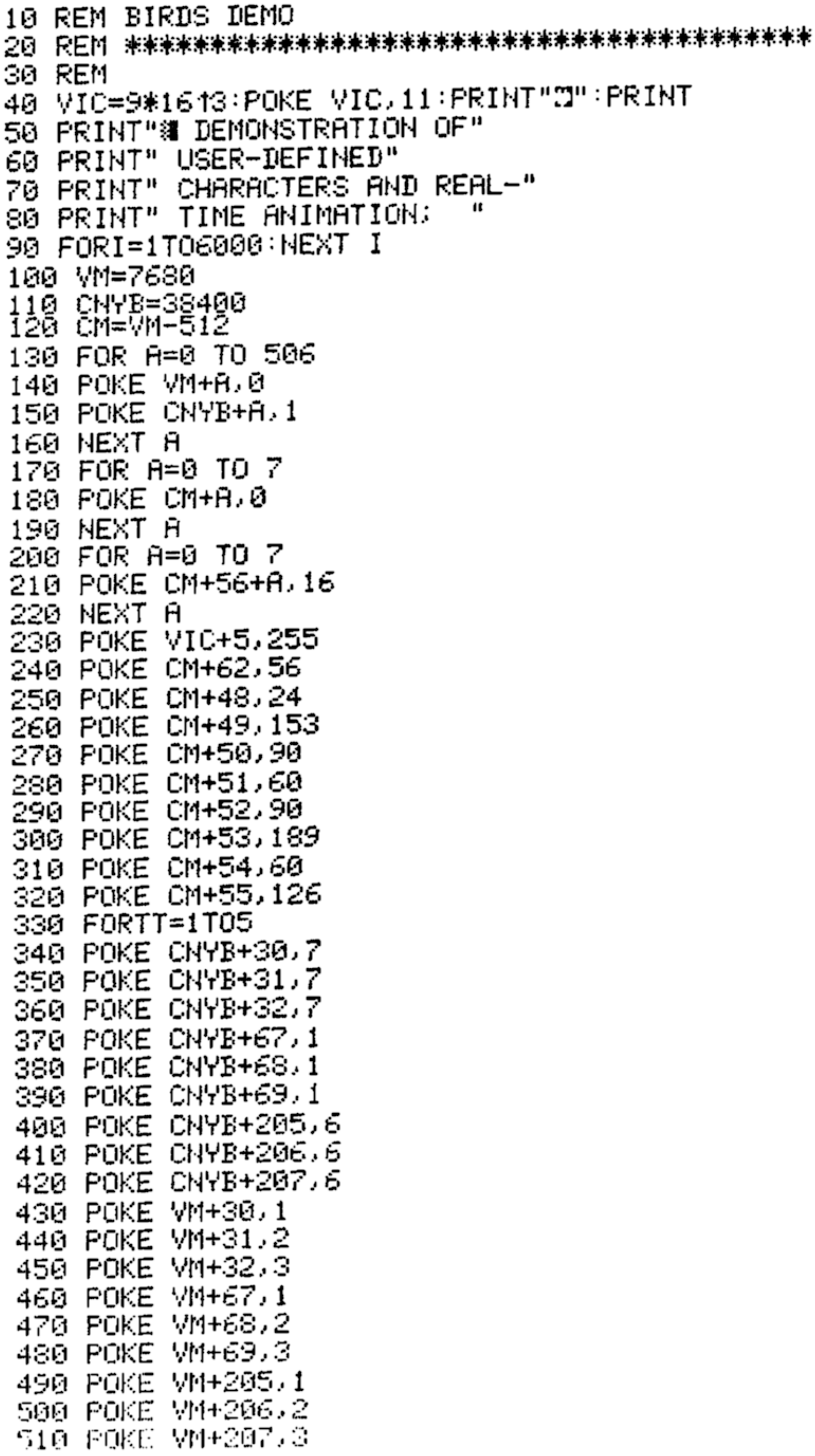
RUNNING THE PROGRAM

When loaded type RUN and watch the display appear on the screen.

PROGRAM STRUCTURE

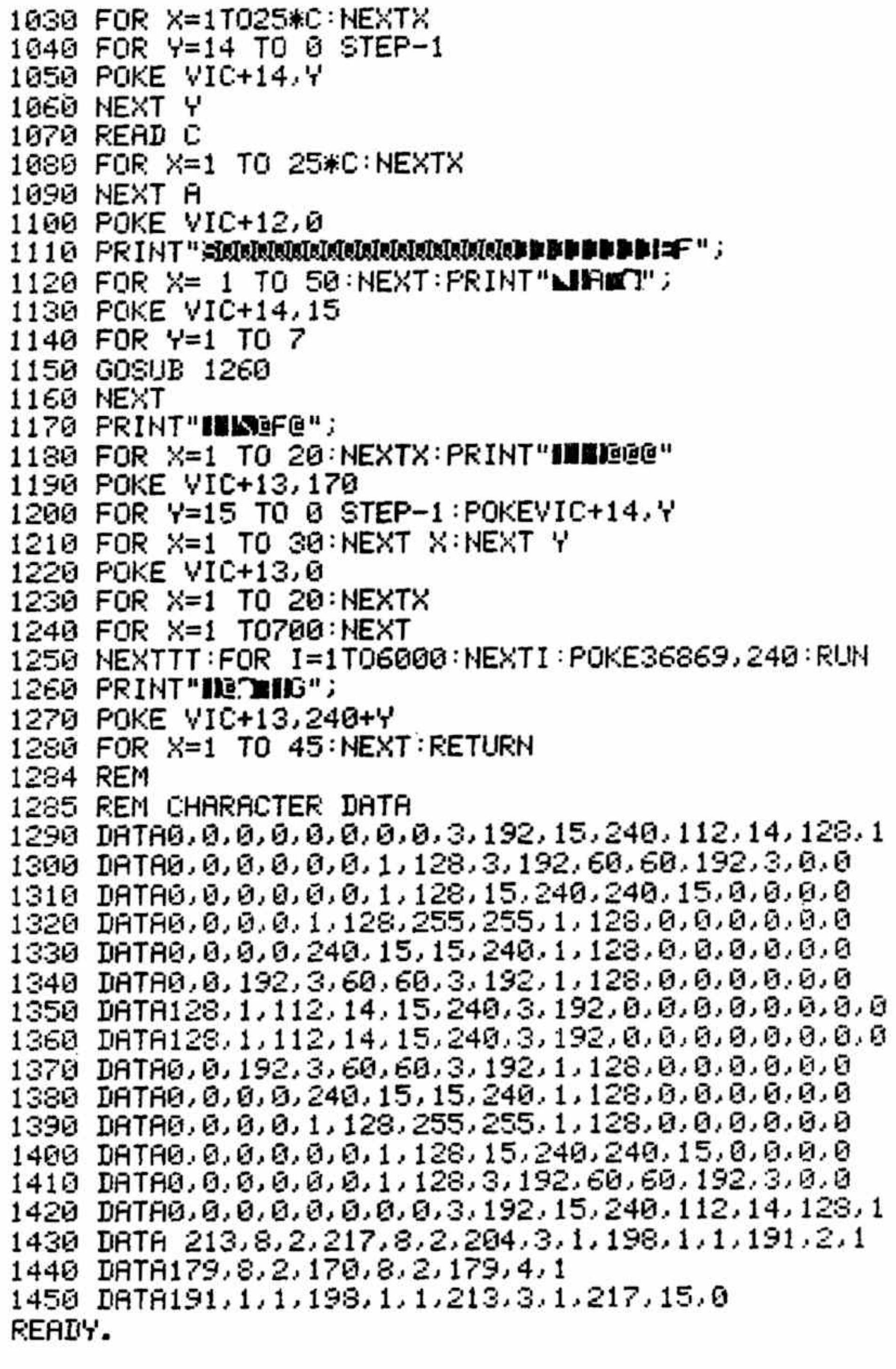
The lines of most interest in this program are as follows:

50-80 Display introduction to program.
130-160 Put character 0 white in every position on the screen.
170-190 Set character 0 to a space.
240-320 Change character 6.
340-420 Put colours yellow, white and blue for three birds on the screen.
430-510 Put birds on the screen.
520-590 Change character 2.
610-700 Display sea, birds and cloud.
740-790 Change character 5.
800-980 Change characters 1-7 using data at end of program.
1290-1450 Character data.



```
10 REM BIRDS DEMO
20 REM ****************************************
30 REM
40 VIC=9*16↑3:POKE VIC,11:PRINT"♥":PRINT
50 PRINT"▓ DEMONSTRATION OF"
60 PRINT" USER-DEFINED"
70 PRINT" CHARACTERS AND REAL-"
80 PRINT" TIME ANIMATION:  "
90 FORI=1TO6000:NEXT I
100 VM=7680
110 CNYB=38400
120 CM=VM-512
130 FOR A=0 TO 506
140 POKE VM+A,0
150 POKE CNYB+A,1
160 NEXT A
170 FOR A=0 TO 7
180 POKE CM+A,0
190 NEXT A
200 FOR A=0 TO 7
210 POKE CM+56+A,16
220 NEXT A
230 POKE VIC+5,255
240 POKE CM+62,56
250 POKE CM+48,24
260 POKE CM+49,153
270 POKE CM+50,90
280 POKE CM+51,60
290 POKE CM+52,90
300 POKE CM+53,189
310 POKE CM+54,60
320 POKE CM+55,126
330 FORTT=1TO5
340 POKE CNYB+30,7
350 POKE CNYB+31,7
360 POKE CNYB+32,7
370 POKE CNYB+67,1
380 POKE CNYB+68,1
390 POKE CNYB+69,1
400 POKE CNYB+205,6
410 POKE CNYB+206,6
420 POKE CNYB+207,6
430 POKE VM+30,1
440 POKE VM+31,2
450 POKE VM+32,3
460 POKE VM+67,1
470 POKE VM+68,2
480 POKE VM+69,3
490 POKE VM+205,1
500 POKE VM+206,2
510 POKE VM+207,3
```

```
520 POKE CM+16,126
530 POKE CM+17,60
540 POKE CM+18,153
550 POKE CM+19,255
560 POKE CM+20,255
570 POKE CM+21,126
580 POKE CM+22,24
590 POKE CM+23,24
600 POKE VIC+15,239
610 PRINT"█████████████████";
620 PRINT"◣ACACACACACACACACACAC";
630 PRINT"█@ACACA@@@CACA@@CAC@AC";
640 PRINT"◣@@AC@@@CACACA@ACAC@@@";
650 PRINT"█@@@ACACAC@@ACAC@CAC@@@";
660 PRINT"◣CAC@@CAC@ACACA@AC@ACA@";
670 PRINT"█@@ACACA@@CAC@ACA@@@AC";
680 PRINT"███████████ABC██████ABC██████ABC"
690 PRINT"████████████████EDDDDDD███████",
700 PRINT"EDDDDD██████EDDDD███EDD██E"
710 FOR A=0 TO 9
720 POKE CM+32+A,255
730 NEXT A
740 POKE CM+42,127
750 POKE CM+43,127
760 POKE CM+44,63
770 POKE CM+45,31
780 POKE CM+46,15
790 POKE CM+47,3
800 FOR Z=1 TO 5
810 RESTORE
820 FOR A=1 TO 7
830 FOR B=0 TO 7
840 READ C
850 POKE CM+8+B,C
860 READ C
870 POKE CM+24+B,C
880 NEXT B
890 NEXT A
900 FOR A=1 TO 7
910 FOR B=0 TO 7
920 READ C
930 POKE CM+8+B,C
940 READ C
950 POKE CM+24+B,C
960 NEXT B
970 NEXT A
980 NEXT Z
990 FOR A=1 TO 12
1000 READB:READC
1010 POKE VIC+12,B
1020 POKE VIC+14,15
```



```
1030 FOR X=1TO25*C:NEXTX
1040 FOR Y=14 TO 0 STEP-1
1050 POKE VIC+14,Y
1060 NEXT Y
1070 READ C
1080 FOR X=1 TO 25*C:NEXTX
1090 NEXT A
1100 POKE VIC+12,0
1110 PRINT"SQQQQQQQQQQQQQQQQQ█████████▐=F";
1120 FOR X= 1 TO 50:NEXT:PRINT"◣▌A■⌐]";
1130 POKE VIC+14,15
1140 FOR Y=1 TO 7
1150 GOSUB 1260
1160 NEXT
1170 PRINT"██◣@F@";
1180 FOR X=1 TO 20:NEXTX:PRINT"███@@@"
1190 POKE VIC+13,170
1200 FOR Y=15 TO 0 STEP-1:POKEVIC+14,Y
1210 FOR X=1 TO 30:NEXT X:NEXT Y
1220 POKE VIC+13,0
1230 FOR X=1 TO 20:NEXTX
1240 FOR X=1 TO700:NEXT
1250 NEXTTT:FOR I=1TO6000:NEXTI:POKE36869,240:RUN
1260 PRINT"▌@⌐█▌G";
1270 POKE VIC+13,240+Y
1280 FOR X=1 TO 45:NEXT:RETURN
1284 REM
1285 REM CHARACTER DATA
1290 DATA0,0,0,0,0,0,0,0,3,192,15,240,112,14,128,1
1300 DATA0,0,0,0,0,1,128,3,192,60,60,192,3,0,0
1310 DATA0,0,0,0,0,1,128,15,240,240,15,0,0,0,0
1320 DATA0,0,0,0,1,128,255,255,1,128,0,0,0,0,0
1330 DATA0,0,0,0,240,15,15,240,1,128,0,0,0,0,0
1340 DATA0,0,192,3,60,60,3,192,1,128,0,0,0,0,0
1350 DATA128,1,112,14,15,240,3,192,0,0,0,0,0,0,0
1360 DATA128,1,112,14,15,240,3,192,0,0,0,0,0,0,0
1370 DATA0,0,192,3,60,60,3,192,1,128,0,0,0,0,0
1380 DATA0,0,0,0,240,15,15,240,1,128,0,0,0,0,0
1390 DATA0,0,0,0,1,128,255,255,1,128,0,0,0,0,0
1400 DATA0,0,0,0,0,1,128,15,240,240,15,0,0,0,0
1410 DATA0,0,0,0,0,1,128,3,192,60,60,192,3,0,0
1420 DATA0,0,0,0,0,0,0,3,192,15,240,112,14,128,1
1430 DATA 213,8,2,217,8,2,204,3,1,198,1,1,191,2,1
1440 DATA179,8,2,170,8,2,179,4,1
1450 DATA191,1,1,198,1,1,213,3,1,217,15,0
READY.
```

RHINO

DESCRIPTION

The aim of the game is to make your way through the jungle while avoiding wild and dangerous rhinos. A map of the jungle with the position of the trees, the house and the players current position are displayed attractively on the screen.

The opponents — the player and the rhinos — are hidden from each other by dense jungle. Once the rhino has seen you, it's position is/are (depending on how far into the game the player has ventured) displayed and then the race is on. The rhino will charge, moving one step to each of the players. It is up to the player to escape amongst the trees; seemingly an easy thing to do, but rhinos are very smart critters. If a rhino catches up with the player, its wham: and the game is lost. Each time the player outwits the rhinos more come to their aid, until the player has to avoid a maximum of 20 such beasts.

Without giving too much away a tip for the player is to try to trap the rhino in a downward facing U-shaped group of trees. The player will be able to move diagonally out of the area but the rhinos remain feeding on their frustration.

EQUIPMENT REQUIRED

Basic VIC-20 with 3K expansion.

RUNNING THE PROGRAM

This program will only work with 3K expansion if all the REM statements are omitted.

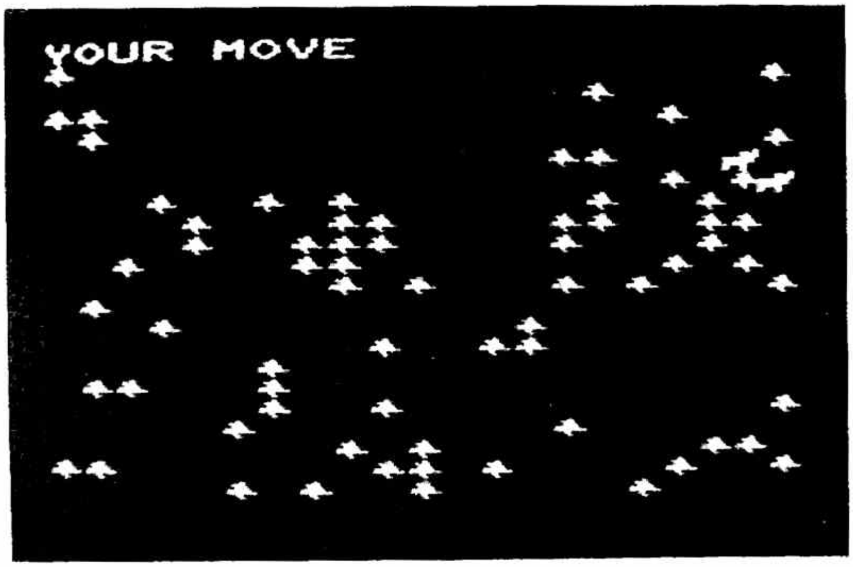
After loading type run and follow the instructions.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

120-130	Limit the topof memory and dimension arrays.
200-3170	Play the game.
770-1090	Set up display.

1130-1400 Move the man in the required direction.
3210-3270 Set up graphics characters.
3310-3390 Character data.

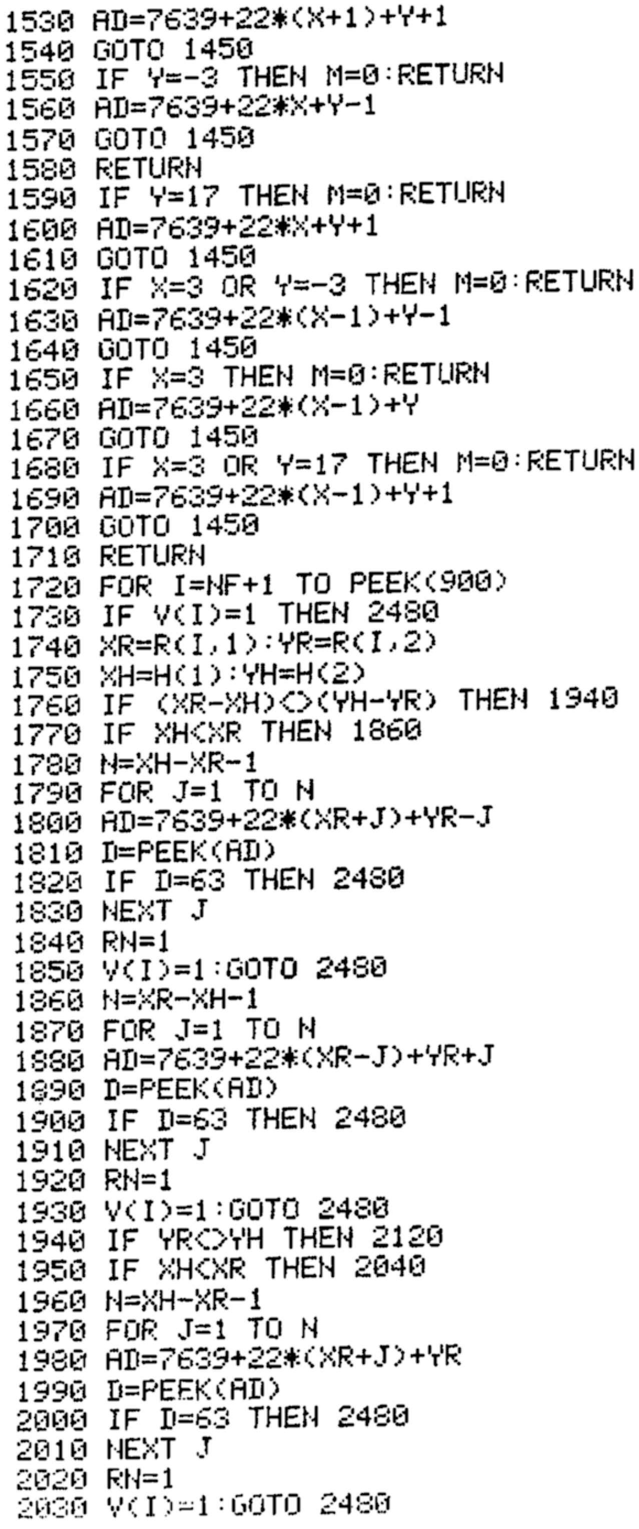


YOUR MOVE

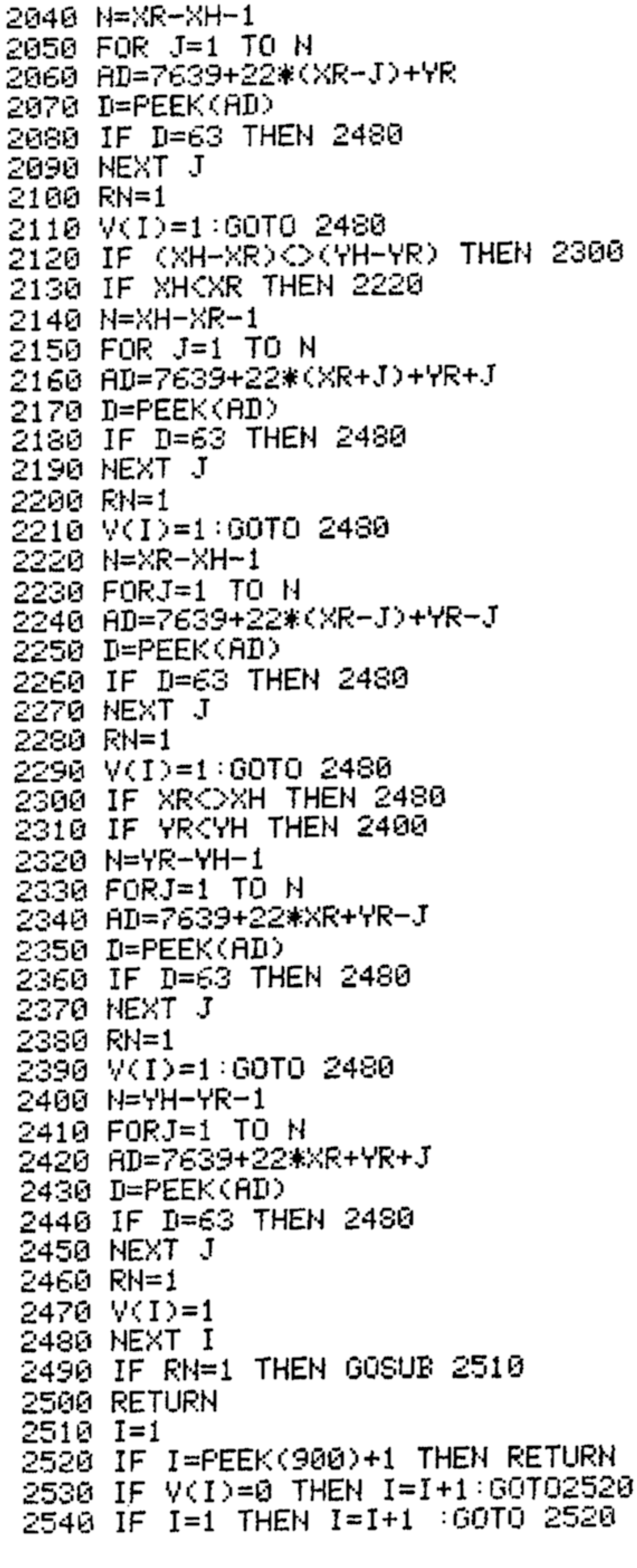
```
10 REM RHINO
20 REM ****************************************
30 REM
40 REM THIS PROGRAM WILL WORK WITH 3K
50 REM EXPANSION, PROVIDED THE REM
60 REM STATEMENTS ARE LEFT OUT
70 REM
80 POKE 900,1:REM # OF RHINOS
90 REM
100 REM LIMIT TOP OF MEMORY
110 REM
120 POKE 56,PEEK(56)-1:CLR:POKE 56,PEEK(56)+1
130 DIM H(2),R(20,2),V(20):POKE 56,PEEK(56)-1
140 GOSUB 3210:POKE 36869,255
150 PRINT"▣":POKE 36879,42
160 GOSUB 730
170 REM
180 REM PLAY GAME
190 REM
200 NF=0
210 X=H(1):Y=H(2)
220 A$="<"
230 PRINT"■■■YOUR MOVE            ■"
240 GET M$:IF M$="" THEN 240
250 PRINT"■■                     ■"
260 M=VAL(M$)
270 IF M=0 THEN 450
280 GOSUB 1410
290 IF WIN>0 THEN 690
300 GOSUB 1060
310 H(1)=X:H(2)=Y
320 RN=0
330 IF NF<PEEK(900)THEN GOSUB 1720
340 IF RN=0 THEN GOSUB 2600
350 IF WIN>0 THEN520
360 IF RN=0 THEN 210
370 FOR I=1 TO 20
380 IF V(I)=0 THEN 210
390 AD=7639+22*R(I,1)+R(I,2)
400 POKE AD,62:POKE AD+30720,1
410 NF=NF+1
420 NEXT I
430 GOTO 210
440 END
450 POKE 36869,240:PRINT"▣■ YOU RESIGN "
460 GOTO 470
470 POKE 56,PEEK(56)+1:POKE 36869,240
480 INPUT"■DO YOU WANT ANOTHER GO";R$
490 IFRIGHT$(R$,1)<>"N"ANDRIGHT$(R$,2)<>"NO"THEN510
500 PRINT"■":POKE 36879,27:END
510 CLR:GOTO 120
```

```
520 PRINT"▒";
530 FOR II=1 TO H(1)-2
540 PRINT"▒";
550 NEXT II
560 FOR II=1 TO H(2)+3
570 PRINT"▒";
580 NEXT II
590 PRINT"▒";
600 FOR I=1 TO 4
610 PRINT"▒▒"CHR$(60-I);
620 FOR J=1 TO 500:NEXTJ
630 NEXT I
640 T=TI
650 IF TI-T<120 THEN 650
660 PRINT"▒";
670 WIN=0
680 GOTO 470
690 PRINT"▒▒ YOU WIN"
700 IF PEEK(900)<20 THEN POKE 900,PEEK(900)+1
710 WIN=0
720 GOTO 470
730 PRINT"▒▒▒THERE ARE "PEEK(900)" RHINOS▒"
740 REM
750 REM SET UP DISPLAY
760 REM
770 FOR I=1 TO 21
780 A$=""
790 FOR J=1 TO 21
800 P=INT(RND(1)+.2)
810 IFP=1 THEN A$=A$+"▒?▒":GOTO 830
820 A$=A$+"▒ ▒"
830 NEXT J
840 PRINTA$
850 NEXT I
860 PRINTTAB(7);"▒▒▒▒       ▒"
870 PRINTTAB(7);"▒      ▒"
880 PRINTTAB(7);"▒  ▒▒=▒▒   ▒"
890 PRINTTAB(7);"▒      ▒"
900 PRINTTAB(7);"▒      ▒"
910 PRINTTAB(14);"▒▒▒▒▒▒▒▒▒▒▒▒"
920 PRINTTAB(13);"▒    ▒"
930 PRINTTAB(13);"▒ ▒▒<▒▒ ▒▒";
940 H(1)=22:H(2)=11
950 FOR I=1 TO PEEK(900)
960 Y=INT(RND(1)*30)+5
970 X=INT(RND(1)*8)+8
980 R(I,1)=X:R(I,2)=Y
990 D=PEEK(7639+22*X+Y)
1000 IF D=63 THEN 960
1010 NEXT I
```

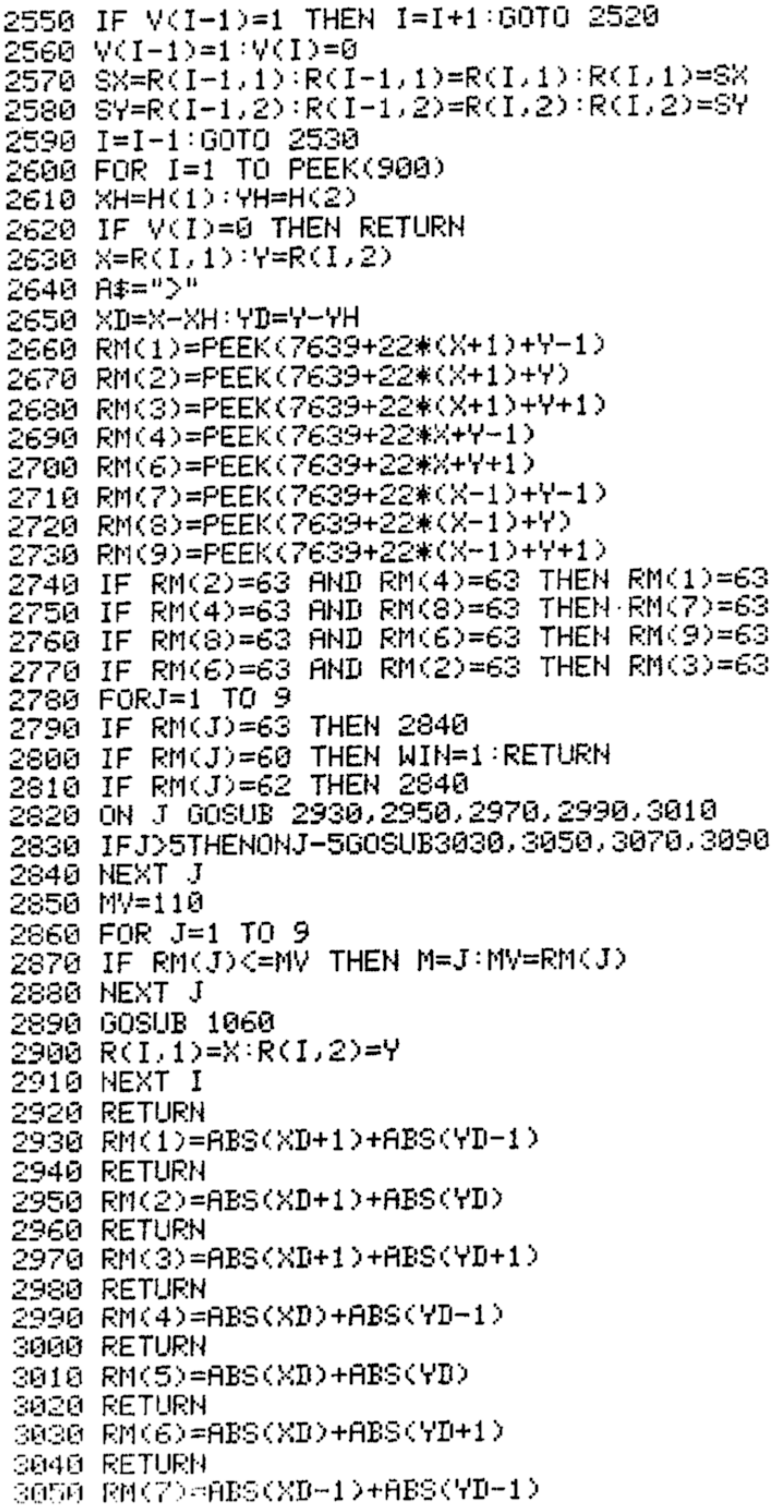
```
1020 FOR I=1 TO 20
1030 V(I)=0
1040 NEXT I
1050 RETURN
1060 IF ABS(M-5)>4 THENM=5
1070 C$=" "
1080 GOSUB 3110
1090 C$=A$
1100 REM
1110 REM MOVE MAN
1120 REM
1130 ON M GOTO 1150,1180,1210,1240,1270
1140 IFM>5THEN ON M-5 GOTO 1290,1320,1350,1380
1150 X=X+1:Y=Y-1
1160 GOSUB 3110
1170 RETURN
1180 X=X+1
1190 GOSUB 3110
1200 RETURN
1210 X=X+1:Y=Y+1
1220 GOSUB 3110
1230 RETURN
1240 Y=Y-1
1250 GOSUB 3110
1260 RETURN
1270 GOSUB 3110
1280 RETURN
1290 Y=Y+1
1300 GOSUB 3110
1310 RETURN
1320 X=X-1:Y=Y-1
1330 GOSUB 3110
1340 RETURN
1350 X=X-1
1360 GOSUB 3110
1370 RETURN
1380 X=X-1:Y=Y+1
1390 GOSUB 3110
1400 RETURN
1410 ON M GOTO 1430,1490,1520,1550,1580
1420 IFM>5THEN ON M-5 GOTO 1590,1620,1650,1680
1430 IF X=23 OR Y=-3 THEN M=0:RETURN
1440 AD=7639+22*(X+1)+(Y-1)
1450 T=PEEK(AD)
1460 IF T=63 THEN M=0
1470 IF T=61 THEN WIN=2 :M=0
1480 RETURN
1490 IF X=23 THEN M=0:RETURN
1500 AD=7639+22*(X+1)+Y
1510 GOTO 1450
1520 IF X=23 OR Y=17 THEN M=0:RETURN
```



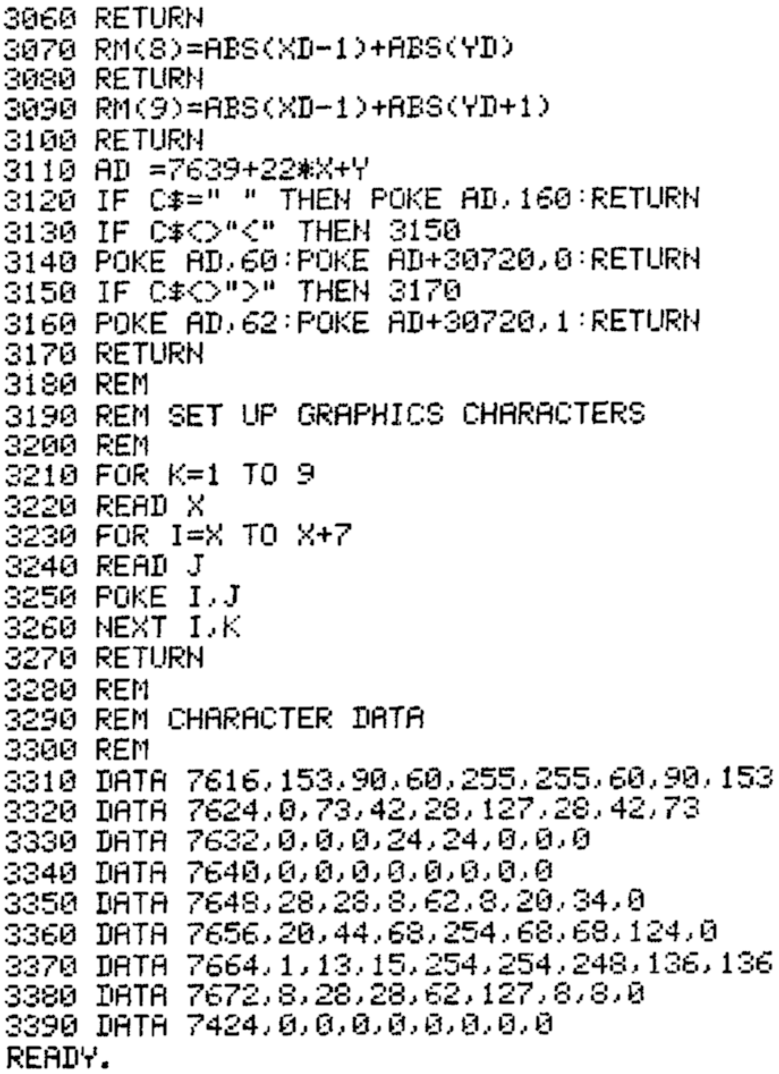
```
1530 AD=7639+22*(X+1)+Y+1
1540 GOTO 1450
1550 IF Y=-3 THEN M=0:RETURN
1560 AD=7639+22*X+Y-1
1570 GOTO 1450
1580 RETURN
1590 IF Y=17 THEN M=0:RETURN
1600 AD=7639+22*X+Y+1
1610 GOTO 1450
1620 IF X=3 OR Y=-3 THEN M=0:RETURN
1630 AD=7639+22*(X-1)+Y-1
1640 GOTO 1450
1650 IF X=3 THEN M=0:RETURN
1660 AD=7639+22*(X-1)+Y
1670 GOTO 1450
1680 IF X=3 OR Y=17 THEN M=0:RETURN
1690 AD=7639+22*(X-1)+Y+1
1700 GOTO 1450
1710 RETURN
1720 FOR I=NF+1 TO PEEK(900)
1730 IF V(I)=1 THEN 2480
1740 XR=R(I,1):YR=R(I,2)
1750 XH=H(1):YH=H(2)
1760 IF (XR-XH)<>(YH-YR) THEN 1940
1770 IF XH<XR THEN 1860
1780 N=XH-XR-1
1790 FOR J=1 TO N
1800 AD=7639+22*(XR+J)+YR-J
1810 D=PEEK(AD)
1820 IF D=63 THEN 2480
1830 NEXT J
1840 RN=1
1850 V(I)=1:GOTO 2480
1860 N=XR-XH-1
1870 FOR J=1 TO N
1880 AD=7639+22*(XR-J)+YR+J
1890 D=PEEK(AD)
1900 IF D=63 THEN 2480
1910 NEXT J
1920 RN=1
1930 V(I)=1:GOTO 2480
1940 IF YR<>YH THEN 2120
1950 IF XH<XR THEN 2040
1960 N=XH-XR-1
1970 FOR J=1 TO N
1980 AD=7639+22*(XR+J)+YR
1990 D=PEEK(AD)
2000 IF D=63 THEN 2480
2010 NEXT J
2020 RN=1
2030 V(I)=1:GOTO 2480
```



```
2040 N=XR-XH-1
2050 FOR J=1 TO N
2060 AD=7639+22*(XR-J)+YR
2070 D=PEEK(AD)
2080 IF D=63 THEN 2480
2090 NEXT J
2100 RN=1
2110 V(I)=1:GOTO 2480
2120 IF (XH-XR)<>(YH-YR) THEN 2300
2130 IF XH<XR THEN 2220
2140 N=XH-XR-1
2150 FOR J=1 TO N
2160 AD=7639+22*(XR+J)+YR+J
2170 D=PEEK(AD)
2180 IF D=63 THEN 2480
2190 NEXT J
2200 RN=1
2210 V(I)=1:GOTO 2480
2220 N=XR-XH-1
2230 FORJ=1 TO N
2240 AD=7639+22*(XR-J)+YR-J
2250 D=PEEK(AD)
2260 IF D=63 THEN 2480
2270 NEXT J
2280 RN=1
2290 V(I)=1:GOTO 2480
2300 IF XR<>XH THEN 2480
2310 IF YR<YH THEN 2400
2320 N=YR-YH-1
2330 FORJ=1 TO N
2340 AD=7639+22*XR+YR-J
2350 D=PEEK(AD)
2360 IF D=63 THEN 2480
2370 NEXT J
2380 RN=1
2390 V(I)=1:GOTO 2480
2400 N=YH-YR-1
2410 FORJ=1 TO N
2420 AD=7639+22*XR+YR+J
2430 D=PEEK(AD)
2440 IF D=63 THEN 2480
2450 NEXT J
2460 RN=1
2470 V(I)=1
2480 NEXT I
2490 IF RN=1 THEN GOSUB 2510
2500 RETURN
2510 I=1
2520 IF I=PEEK(900)+1 THEN RETURN
2530 IF V(I)=0 THEN I=I+1:GOTO2520
2540 IF I=1 THEN I=I+1  :GOTO 2520
```



```
2550 IF V(I-1)=1 THEN I=I+1:GOTO 2520
2560 V(I-1)=1:V(I)=0
2570 SX=R(I-1,1):R(I-1,1)=R(I,1):R(I,1)=SX
2580 SY=R(I-1,2):R(I-1,2)=R(I,2):R(I,2)=SY
2590 I=I-1:GOTO 2530
2600 FOR I=1 TO PEEK(900)
2610 XH=H(1):YH=H(2)
2620 IF V(I)=0 THEN RETURN
2630 X=R(I,1):Y=R(I,2)
2640 A$=">"
2650 XD=X-XH:YD=Y-YH
2660 RM(1)=PEEK(7639+22*(X+1)+Y-1)
2670 RM(2)=PEEK(7639+22*(X+1)+Y)
2680 RM(3)=PEEK(7639+22*(X+1)+Y+1)
2690 RM(4)=PEEK(7639+22*X+Y-1)
2700 RM(6)=PEEK(7639+22*X+Y+1)
2710 RM(7)=PEEK(7639+22*(X-1)+Y-1)
2720 RM(8)=PEEK(7639+22*(X-1)+Y)
2730 RM(9)=PEEK(7639+22*(X-1)+Y+1)
2740 IF RM(2)=63 AND RM(4)=63 THEN RM(1)=63
2750 IF RM(4)=63 AND RM(8)=63 THEN RM(7)=63
2760 IF RM(8)=63 AND RM(6)=63 THEN RM(9)=63
2770 IF RM(6)=63 AND RM(2)=63 THEN RM(3)=63
2780 FORJ=1 TO 9
2790 IF RM(J)=63 THEN 2840
2800 IF RM(J)=60 THEN WIN=1:RETURN
2810 IF RM(J)=62 THEN 2840
2820 ON J GOSUB 2930,2950,2970,2990,3010
2830 IFJ>5THENONJ-5GOSUB3030,3050,3070,3090
2840 NEXT J
2850 MV=110
2860 FOR J=1 TO 9
2870 IF RM(J)<=MV THEN M=J:MV=RM(J)
2880 NEXT J
2890 GOSUB 1060
2900 R(I,1)=X:R(I,2)=Y
2910 NEXT I
2920 RETURN
2930 RM(1)=ABS(XD+1)+ABS(YD-1)
2940 RETURN
2950 RM(2)=ABS(XD+1)+ABS(YD)
2960 RETURN
2970 RM(3)=ABS(XD+1)+ABS(YD+1)
2980 RETURN
2990 RM(4)=ABS(XD)+ABS(YD-1)
3000 RETURN
3010 RM(5)=ABS(XD)+ABS(YD)
3020 RETURN
3030 RM(6)=ABS(XD)+ABS(YD+1)
3040 RETURN
3050 RM(7)=ABS(XD-1)+ABS(YD-1)
```



```
3060 RETURN
3070 RM(8)=ABS(XD-1)+ABS(YD)
3080 RETURN
3090 RM(9)=ABS(XD-1)+ABS(YD+1)
3100 RETURN
3110 AD =7639+22*X+Y
3120 IF C$=" " THEN POKE AD,160:RETURN
3130 IF C$<>"<" THEN 3150
3140 POKE AD,60:POKE AD+30720,0:RETURN
3150 IF C$<>">" THEN 3170
3160 POKE AD,62:POKE AD+30720,1:RETURN
3170 RETURN
3180 REM
3190 REM SET UP GRAPHICS CHARACTERS
3200 REM
3210 FOR K=1 TO 9
3220 READ X
3230 FOR I=X TO X+7
3240 READ J
3250 POKE I,J
3260 NEXT I,K
3270 RETURN
3280 REM
3290 REM CHARACTER DATA
3300 REM
3310 DATA 7616,153,90,60,255,255,60,90,153
3320 DATA 7624,0,73,42,28,127,28,42,73
3330 DATA 7632,0,0,0,24,24,0,0,0
3340 DATA 7640,0,0,0,0,0,0,0,0
3350 DATA 7648,28,28,8,62,8,20,34,0
3360 DATA 7656,20,44,68,254,68,68,124,0
3370 DATA 7664,1,13,15,254,254,248,136,136
3380 DATA 7672,8,28,28,62,127,8,8,0
3390 DATA 7424,0,0,0,0,0,0,0,0
READY.
```

DO-RAY-ME

DESCRIPTION

This program allows the VIC keyboard to be used as a piano keyboard. The program has a range of three octaves and sharps produced by using the shift key on the top two octaves. The keys being used are : Z-, Bottom octave (no sharps), A-K Middle octave (shifted for sharp), Q-I Top octave (shifted for sharp). 0 turns off the note.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion required and connected to a television set for sound.

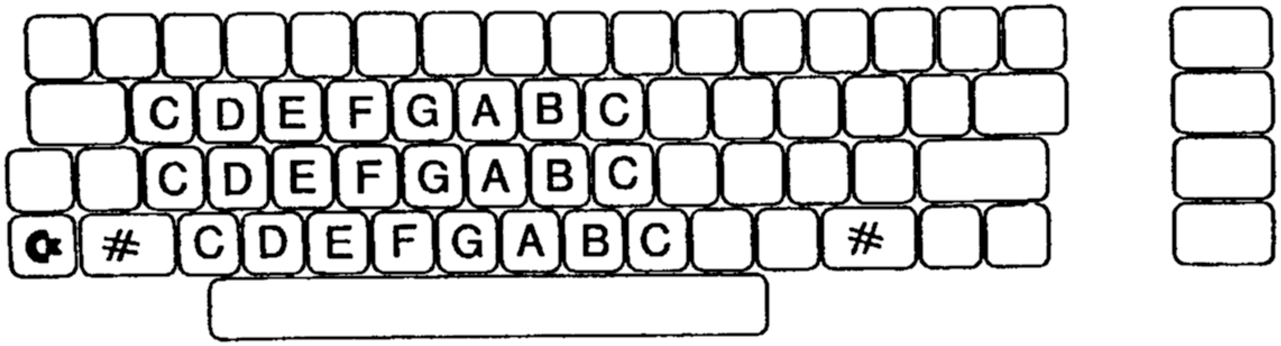
RUNNING THE PROGRAM

After typing run, use the VIC keyboard as a piano and play some music.

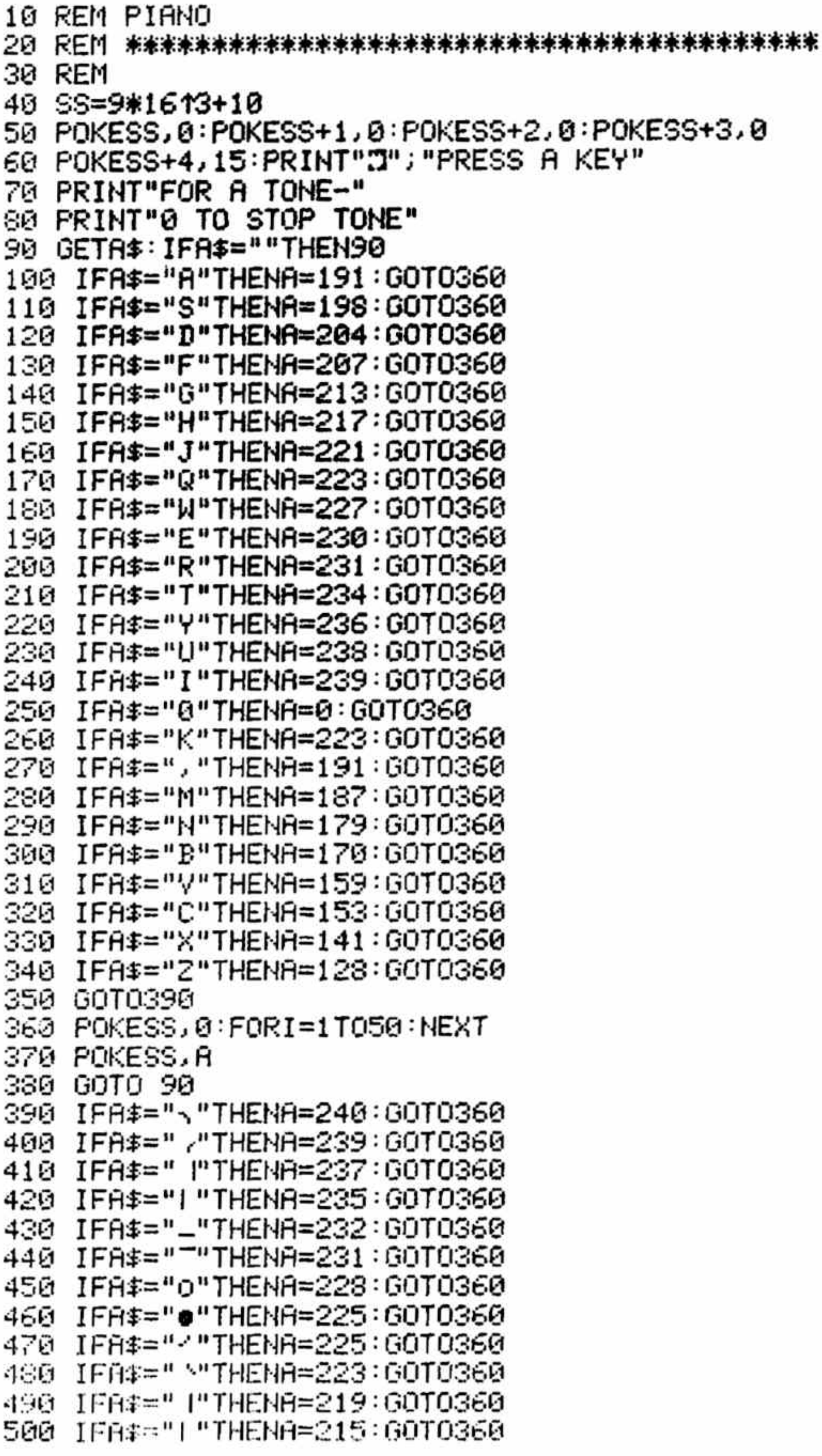
PROGRAM STRUCTURE

The lines of interest in this program are as follows.

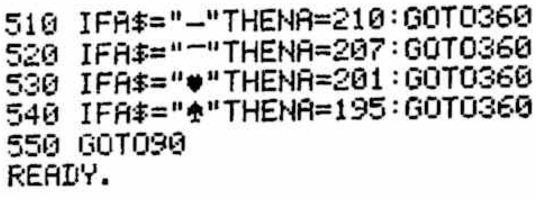
40	SS holds the address of sound register 0.
50	Set four sound registers to zero.
60	Set volume register to 15.
90	Input key for note.
100-350	Check for unshifted key.
360-550	Play note.
390-550	Check for shifted key.



Layout of the VIC keyboard



```
10 REM PIANO
20 REM ******************************************
30 REM
40 SS=9*16↑3+10
50 POKESS,0:POKESS+1,0:POKESS+2,0:POKESS+3,0
60 POKESS+4,15:PRINT"⊐";"PRESS A KEY"
70 PRINT"FOR A TONE-"
80 PRINT"0 TO STOP TONE"
90 GETA$:IFA$=""THEN90
100 IFA$="A"THENA=191:GOTO360
110 IFA$="S"THENA=198:GOTO360
120 IFA$="D"THENA=204:GOTO360
130 IFA$="F"THENA=207:GOTO360
140 IFA$="G"THENA=213:GOTO360
150 IFA$="H"THENA=217:GOTO360
160 IFA$="J"THENA=221:GOTO360
170 IFA$="Q"THENA=223:GOTO360
180 IFA$="W"THENA=227:GOTO360
190 IFA$="E"THENA=230:GOTO360
200 IFA$="R"THENA=231:GOTO360
210 IFA$="T"THENA=234:GOTO360
220 IFA$="Y"THENA=236:GOTO360
230 IFA$="U"THENA=238:GOTO360
240 IFA$="I"THENA=239:GOTO360
250 IFA$="0"THENA=0:GOTO360
260 IFA$="K"THENA=223:GOTO360
270 IFA$=","THENA=191:GOTO360
280 IFA$="M"THENA=187:GOTO360
290 IFA$="N"THENA=179:GOTO360
300 IFA$="B"THENA=170:GOTO360
310 IFA$="V"THENA=159:GOTO360
320 IFA$="C"THENA=153:GOTO360
330 IFA$="X"THENA=141:GOTO360
340 IFA$="Z"THENA=128:GOTO360
350 GOTO390
360 POKESS,0:FORI=1TO50:NEXT
370 POKESS,A
380 GOTO 90
390 IFA$="⸜"THENA=240:GOTO360
400 IFA$=" ⸝"THENA=239:GOTO360
410 IFA$=" |"THENA=237:GOTO360
420 IFA$="| "THENA=235:GOTO360
430 IFA$="_"THENA=232:GOTO360
440 IFA$="‾"THENA=231:GOTO360
450 IFA$="o"THENA=228:GOTO360
460 IFA$="●"THENA=225:GOTO360
470 IFA$="⸝"THENA=225:GOTO360
480 IFA$=" ⸜"THENA=223:GOTO360
490 IFA$=" |"THENA=219:GOTO360
500 IFA$="| "THENA=215:GOTO360
```



```
510 IFA$="_"THENA=210:GOTO360
520 IFA$="‾"THENA=207:GOTO360
530 IFA$="♥"THENA=201:GOTO360
540 IFA$="♠"THENA=195:GOTO360
550 GOTO90
READY.
```

SOUND EFFECTS

DESCRIPTION

This program has a selection of noises generated by the computer, with which the user can play around to produce some interesting sound effects.
For example, when the display appears on the screen the user is offered ten choices by mixing sound number three, the bombardment with sound number two, the explosion; the effect is that of a bomb dropping and its resulting explosive sound.
The actual display for this program is a very simple one at first glance, but a great deal of work has gone into the writing. The border surrounding the ten choices changes colour in a random manner.
Should the user decide to write his own games program he will find this program to be of great use when deciding which sound effects to add or spruce up the end result of his work.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

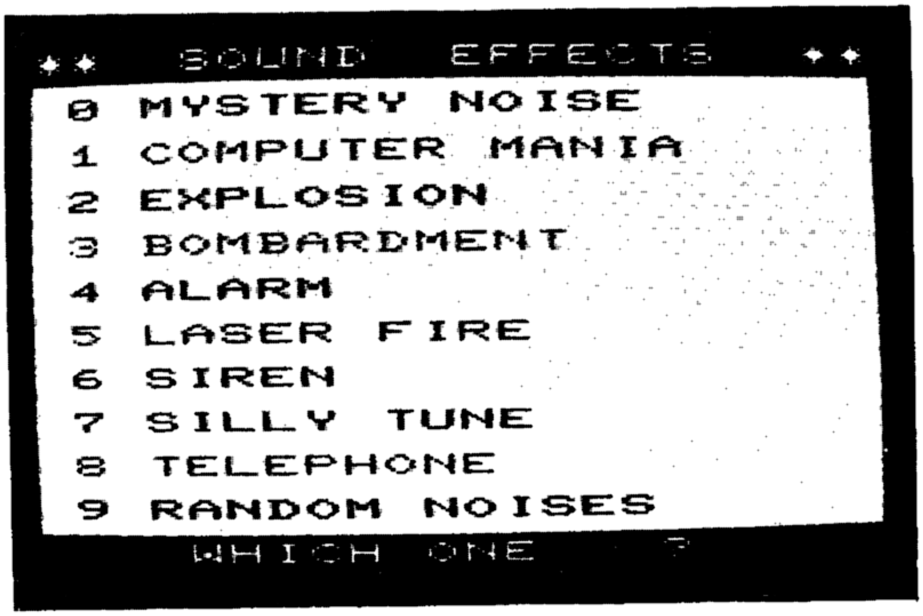
After loading the program and typing run the attractive display will invite the user to make his choice of sound. By hitting the appropriate key i.e. 0, the user will hear a mystery noise. To retire from the program the user must hit the space bar.

PROGRAM STRUCTURE

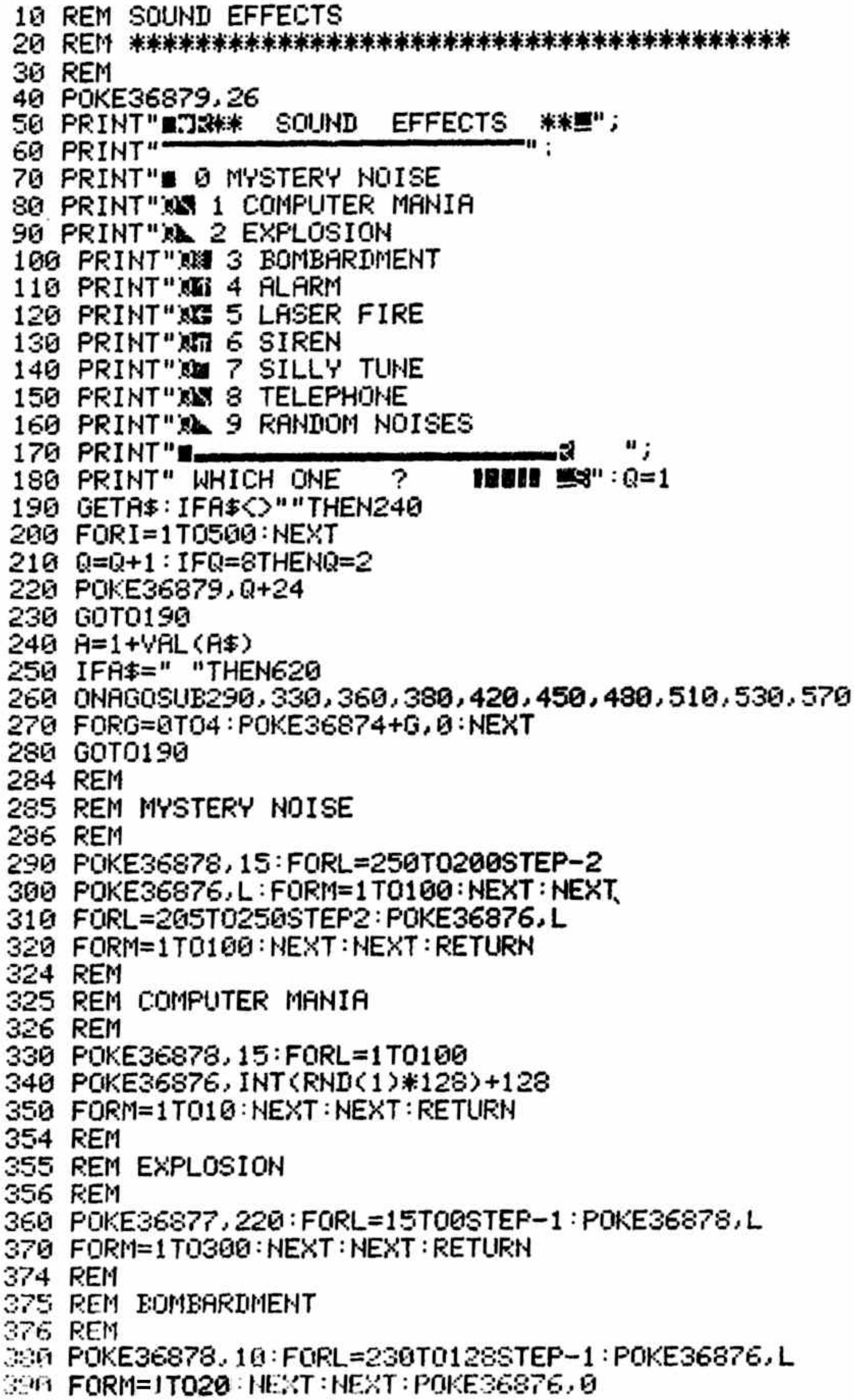
The lines to note in this program are as follows.

50-190	Instructions and the input for 'Which Choice.
290-320	Subroutine to generate the mystery noise.
330-350	Subroutine to generate the computer mania sound.
360-370	Subroutine to generate the explosion sound.
380-410	Subroutine to generate the bombardment sound.

420-440 Subroutine to generate the alarm noise.
450-470 Subroutine to generate the laser fire sound.
480-500 Subroutine to generate the siren.
510-520 Subroutine to generate the silly tune.
530-560 Subroutine to generate the telephone ringing tone.
570-610 Subroutine to generate the random noises.
620 End of program.



```
♦♦    SOUND  EFFECTS    ♦♦
0 MYSTERY NOISE
1 COMPUTER MANIA
2 EXPLOSION
3 BOMBARDMENT
4 ALARM
5 LASER FIRE
6 SIREN
7 SILLY TUNE
8 TELEPHONE
9 RANDOM NOISES
      WHICH ONE   ?
```



```
10 REM SOUND EFFECTS
20 REM ****************************************
30 REM
40 POKE36879,26
50 PRINT"■▯▩**  SOUND  EFFECTS  **▤";
60 PRINT"▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬";
70 PRINT"■ 0 MYSTERY NOISE
80 PRINT"▩◣ 1 COMPUTER MANIA
90 PRINT"▩◣ 2 EXPLOSION
100 PRINT"▩▩ 3 BOMBARDMENT
110 PRINT"▩▩ 4 ALARM
120 PRINT"▩▩ 5 LASER FIRE
130 PRINT"▩▩ 6 SIREN
140 PRINT"▩■ 7 SILLY TUNE
150 PRINT"▩◣ 8 TELEPHONE
160 PRINT"▩◣ 9 RANDOM NOISES
170 PRINT"■▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▩   ";
180 PRINT" WHICH ONE   ?    ▮▮▮▮▮ ▤▩":Q=1
190 GETA$:IFA$<>""THEN240
200 FORI=1TO500:NEXT
210 Q=Q+1:IFQ=8THENQ=2
220 POKE36879,Q+24
230 GOTO190
240 A=1+VAL(A$)
250 IFA$=" "THEN620
260 ONAGOSUB290,330,360,380,420,450,480,510,530,570
270 FORG=0TO4:POKE36874+G,0:NEXT
280 GOTO190
284 REM
285 REM MYSTERY NOISE
286 REM
290 POKE36878,15:FORL=250TO200STEP-2
300 POKE36876,L:FORM=1TO100:NEXT:NEXT
310 FORL=205TO250STEP2:POKE36876,L
320 FORM=1TO100:NEXT:NEXT:RETURN
324 REM
325 REM COMPUTER MANIA
326 REM
330 POKE36878,15:FORL=1TO100
340 POKE36876,INT(RND(1)*128)+128
350 FORM=1TO10:NEXT:NEXT:RETURN
354 REM
355 REM EXPLOSION
356 REM
360 POKE36877,220:FORL=15TO0STEP-1:POKE36878,L
370 FORM=1TO300:NEXT:NEXT:RETURN
374 REM
375 REM BOMBARDMENT
376 REM
380 POKE36878,10:FORL=230TO128STEP-1:POKE36876,L
390 FORM=1TO20:NEXT:NEXT:POKE36876,0
```

```
400 POKE36878,200:FORL=15TO0STEP-.5
410 POKE36878,L:NEXT:RETURN
414 REM
415 REM ALARM
416 REM
420 POKE36878,15:FORL=1TO10:FORM=180TO235STEP2
430 POKE36876,M:FORN=1TO10:NEXT:NEXT
440 POKE36876,0:FORM=1TO100:NEXT:NEXT:RETURN
444 REM
445 REM LASER FIRE
446 REM
450 POKE36878,15:FORL=1TO30:FORM=250TO240STEP-1
460 POKE36876,M:NEXT:FORM=240TO250
470 POKE36876,M:NEXT:POKE36876,0:NEXT:RETURN
474 REM
475 REM SIREN
476 REM
480 POKE36878,15:FORL=1TO10:POKE36875,200
490 FORM=1TO500:NEXT:POKE36875,0:POKE36876,200
500 FORM=1TO500:NEXT:POKE36876,0:NEXT:RETURN
504 REM
505 REM SILLY TUNE
506 REM
510 POKE36878,15:FORL=1TO15:POKE36876,160+L
520 FORM=1TO400:NEXT:NEXT:RETURN
524 REM
525 REM TELEPHONE
526 REM
530 POKE36878,15:FORL=1TO5:FORK=1TO50
540 POKE36876,220:FORN=1TO5:NEXT:POKE36876,0
550 NEXT
560 FORM=1TO3000:NEXT:NEXT:RETURN
564 REM
565 REM RANDOM NOISES
566 REM
570 POKE36878,25:FORL=1TO20
580 FORM=254TO240+INT(RND(1)*10)STEP-1
590 POKE36876,M:NEXT
600 POKE36876,0:FORM=0TOINT(RND(1)*100)+120
610 NEXT:NEXT:RETURN
620 :PRINT"▓▓";:POKE36879,27
READY.
```

ARROW

DESCRIPTION

The aim of this game is to score the highest number of points before the time limit of sixty seconds runs out. In this game the user becomes an arrow which, as the game progresses, grows little blobs on the tail. These blobs increase the risk factor of travelling around within the game area. But travel the player must. To score points the arrow head must pierce randomly appearing squares which contain different point values.

The danger from ever increasing blobs comes from unwary directing of the arrow head in the user's eagerness to spear the squares. The user may (and does) spear himself in the tail. Hitting the surrounding border of the game will also lose the user the match because the effect of hitting the wall turns the arrow head back upon its self and into it's tail.

EQUIPMENT REQUIRED

Basic unexpanded VIC-20.

RUNNING THE PROGRAM

After loading the program and typing run the game appears on your screen with the timing gauge ticking away in the top left hand corner. In the centre of the top border line the scoring gauge tried valiantly to keep up with the timer. To move the arrow head the following letters have been designated:

```
  T
F   H
  B
```

T = Up.

F = Left.

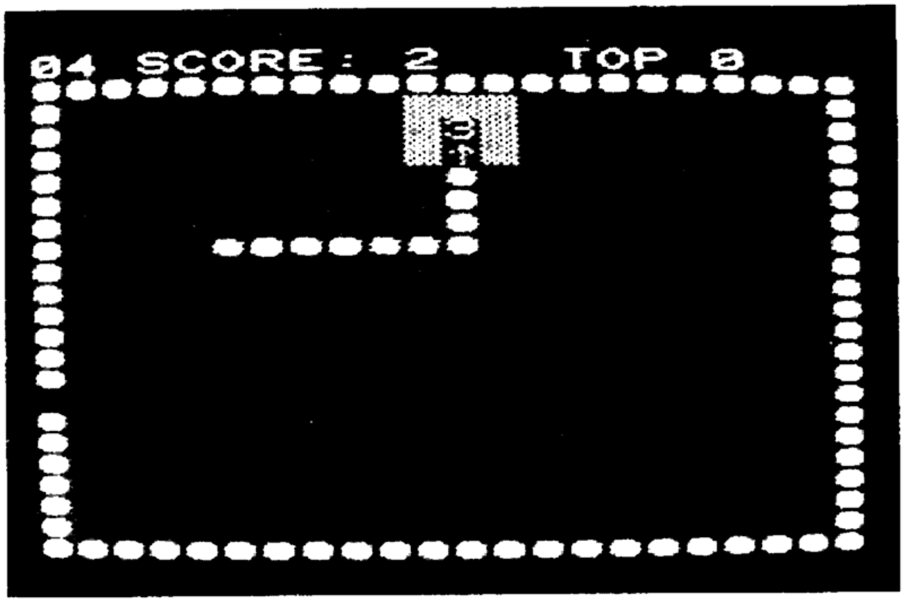
H = Right.

B = Down.

PROGRAM STRUCTURE

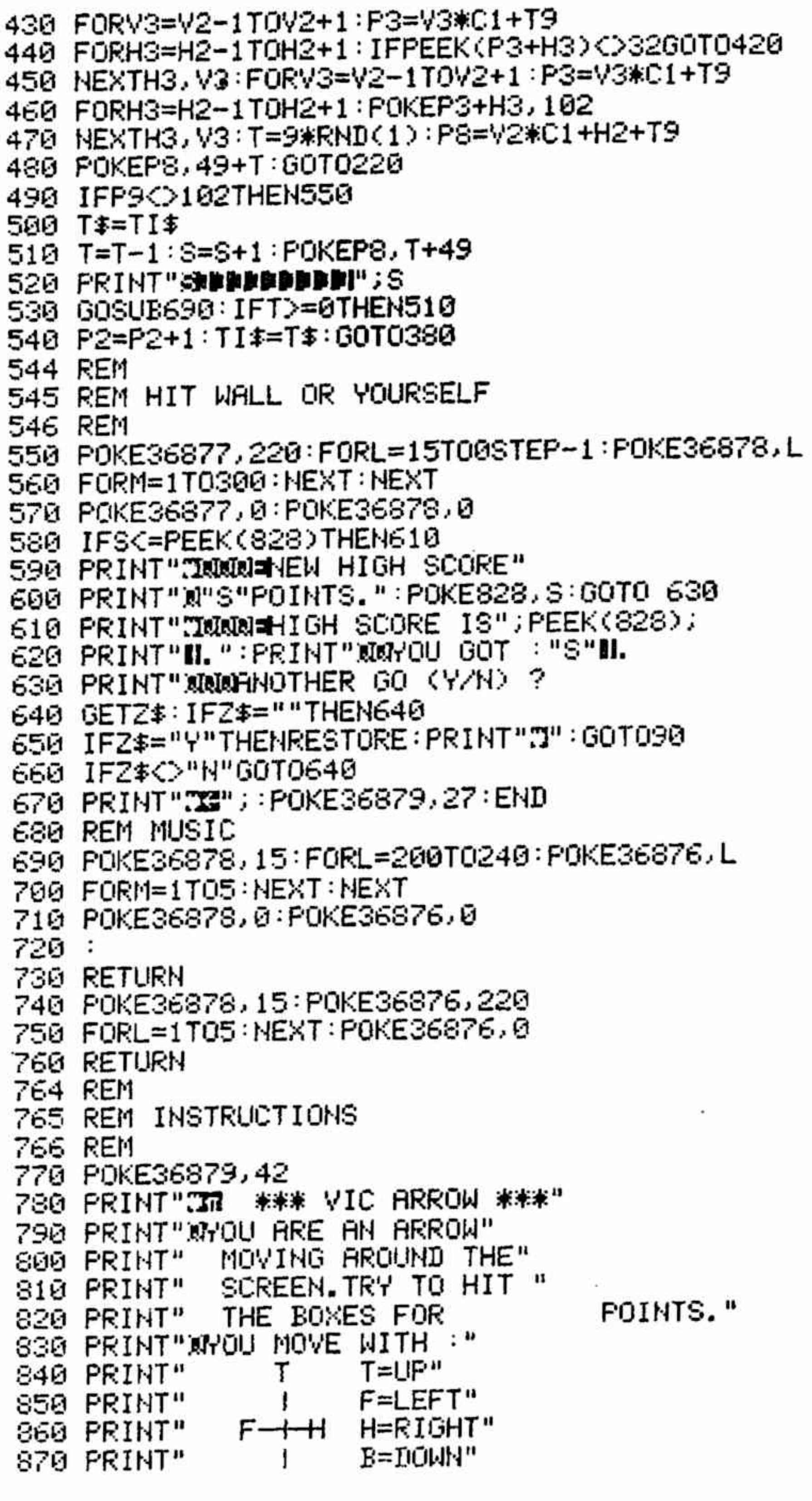
The lines of interest in this program are as follows.

40-50 Find start of screen and colour memory maps.
150-190 Displays border.
200-540 Plays the game.
550-580 Indicates player hitting the wall or his tail.
570-620 Informs player of new high score.
630-640 Input for another go.
690-710 Sounds when scoring.
740-750 Sound when arrow is moving.
770-960 Print instructions.



```
04 SCORE: 2     TOP 0
```

```
10 REM ARROW
20 REM ************************************
30 REM
34 REM
35 REM FIND START OF SCREEN AND COLOUR MEMORY MAPS
36 REM
40 VR=PEEK(648)*256
50 KR=38400:IFVR<>7680THENKR=37888
60 GOSUB770
70 POKE828,0
80 DIMP(80)
90 KL(1)=1:KL(2)=3:KL(3)=5:KL(4)=7:S=0
100 FORI=1TO4:READA$(I):NEXT:DATAB,F,H,T
110 D(0)=22:D(1)=60:D(2)=62:D(3)=30:T6=3599
120 T9=VR
130 CS=VR
140 C1=22    :REM SCREEN WIDTH
144 REM
145 REM DISPLAY BORDER AROUND SCREEN
146 REM
150 PRINT"≡◘   SCORE: 0   TOP"PEEK(828);
160 PRINT"▥▧◙●●●●●●●●●●●●●●●●●●●●●●";
170 FORI=1TO20
180 PRINT"●                    ●";:NEXT
190 PRINT"●●●●●●●●●●●●●●●●●●●●●▮▮▮●S▮
194 REM
195 REM SET TIME TO ZERO AND PLAY GAME
196 REM
200 V=15:H=10:V1=0:H1=-1
210 P2=10:D1=1:TI$="000000"
220 PRINT"▧";RIGHT$(TI$,2):IFTI>T6GOTO580
230 GETZ$:IFZ$=""THEN280
240 Z=50:FORY=1TO4:IFZ$=A$(Y)THENZ=Y-1:Y=5
250 NEXT:IFZ<>INT(Z)ORZ<0ORZ>3GOTO280
260 D1=Z:D=Z-1.5:V1=INT(ABS(D))*SGN(D)
270 H1=SGN(D)-V1
280 V=V-V1:H=H+H1:GOSUB740
290 P=CS+V*C1+H:KL=INT(RND(1)*4+1):KL=KL(KL)
300 POKEKR+V*C1+H,KL
310 P9=PEEK(P)
320 R6=R7:R7=R7+1:IFR7>P2THENR7=0
330 P1=P(R7):P(R7)=P
340 IFP1<>0THENPOKEP1,32:POKEP1-CS+KR,7
350 POKEP,D(D1):P1=P(R6):IFP1<>0THENPOKEP1,81
360 IFP9<>32GOTO490
370 IFRND(1)>.05THEN220
380 FORV3=V2-1TOV2+1:P3=V3*C1+T9
390 FORH3=H2-1TOH2+1:IFPEEK(P3+H3)<>102GOTO410
400 POKEP3+H3,32
410 NEXTH3,V3:T=0:POKEP8,32
420 V2=INT(RND(1)*18)+3:H2=INT(RND(1)*19)+2
```



```
430 FORV3=V2-1TOV2+1:P3=V3*C1+T9
440 FORH3=H2-1TOH2+1:IFPEEK(P3+H3)<>32GOTO420
450 NEXTH3,V3:FORV3=V2-1TOV2+1:P3=V3*C1+T9
460 FORH3=H2-1TOH2+1:POKEP3+H3,102
470 NEXTH3,V3:T=9*RND(1):P8=V2*C1+H2+T9
480 POKEP8,49+T:GOTO220
490 IFP9<>102THEN550
500 T$=TI$
510 T=T-1:S=S+1:POKEP8,T+49
520 PRINT"S█████████";S
530 GOSUB690:IFT>=0THEN510
540 P2=P2+1:TI$=T$:GOTO380
544 REM
545 REM HIT WALL OR YOURSELF
546 REM
550 POKE36877,220:FORL=15TO0STEP-1:POKE36878,L
560 FORM=1TO300:NEXT:NEXT
570 POKE36877,0:POKE36878,0
580 IFS<=PEEK(828)THEN610
590 PRINT"]███≡NEW HIGH SCORE"
600 PRINT"█"S"POINTS.":POKE828,S:GOTO 630
610 PRINT"]███≡HIGH SCORE IS";PEEK(828);
620 PRINT"█.":PRINT"██YOU GOT :"S"█.
630 PRINT"███ANOTHER GO (Y/N) ?
640 GETZ$:IFZ$=""THEN640
650 IFZ$="Y"THENRESTORE:PRINT"]":GOTO90
660 IFZ$<>"N"GOTO640
670 PRINT"]█";:POKE36879,27:END
680 REM MUSIC
690 POKE36878,15:FORL=200TO240:POKE36876,L
700 FORM=1TO5:NEXT:NEXT
710 POKE36878,0:POKE36876,0
720 :
730 RETURN
740 POKE36878,15:POKE36876,220
750 FORL=1TO5:NEXT:POKE36876,0
760 RETURN
764 REM
765 REM INSTRUCTIONS
766 REM
770 POKE36879,42
780 PRINT"]█  *** VIC ARROW ***"
790 PRINT"█YOU ARE AN ARROW"
800 PRINT"  MOVING AROUND THE"
810 PRINT"  SCREEN.TRY TO HIT "
820 PRINT"  THE BOXES FOR          POINTS."
830 PRINT"█YOU MOVE WITH :"
840 PRINT"       T      T=UP"
850 PRINT"       !      F=LEFT"
860 PRINT"     F-+-H    H=RIGHT"
870 PRINT"       !      B=DOWN"
```

```
880 PRINT"      B
890 PRINT"▓YOU MUST AVOID THE"
900 PRINT"  WALLS AND YOURSELF."
910 PRINT"  AS TIME GOES ON THE"
920 PRINT"  ARROW GETS LONGER."
930 PRINT"▓YOU HAVE 60 SECONDS."
940 PRINT"▓HIT ANY KEY TO START■"
950 GETA$:IFA$=""THEN950
960 PRINT"⊐":RETURN
READY.
```

TANK-V-UFO

DESCRIPTION

This game is a test of timing and skill. The aim of the game is to destroy the invading UFO which floats across the screen at varying heights while dropping bombs on the red tank below. The tank meanwhile transverses the ground firing at the UFO. Very attractive little explosion characters appear but are not dangerous.

When hit, the UFO explodes spectacularly but the tank flashes red and black.

EQUIPMENT REQUIRED

Basic VIC-20.

RUNNING THE PROGRAM

In order to run this program on an unexpanded VIC remove all REM statements.

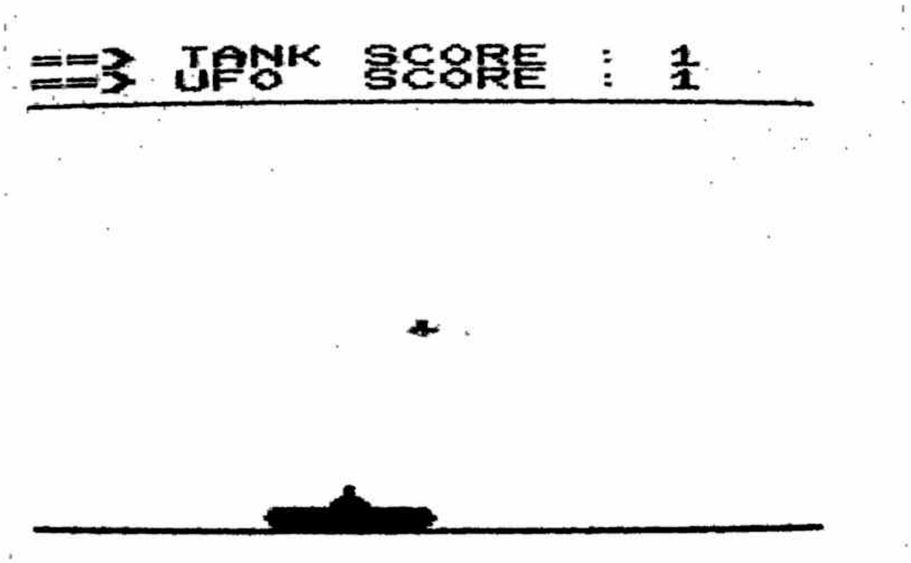
After loading the program and typing run the screen will display a simple line along the base of the screen on which rests the tank. At the top of the screen the first two lines display the score, the number of tanks hit and the number of UFO's wiped out.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

40-50	Limits the top of memory.
70-80	Find start of video and colour memory maps.
130-200	Sets up display.
220-540	Plays the game.
550-620	Motivates the tank.
630-650	Explodes the bombs on the ground.
670-700	Clears the bomb explosions.
710-770	Explodes the UFO's.
780-840	Clears exploded UFO's of the screen.
850-1000	Explodes the tanks.
1070-1170	Print instructions.

1180-1220 Displays score at the top of the screen.
1230-1240 Clears keyboard buffer and ends program.
1260-1320 Sets up characters.
1330-1410 Character data.



```
==> TANK SCORE : 1
==> UFO  SCORE : 1
```

```
10 REM TANK-V-UFO
20 REM ****************************************
30 REM
34 REM
35 REM LIMIT TOP OF MEMORY
36 REM
40 POKE 56,PEEK(56)-2:POKE 52,PEEK(56)-2
50 POKE 51,PEEK(55):CLR:POKE36879,42
60 GOSUB1070:PRINT"□"
64 REM
65 REM FIND START OF VIDEO AND COLOUR MEMORY MAPS
66 REM
70 VR=PEEK(648)*256
80 KR=38400:IFVR<>7680THENKR=37888
90 OF=KR-VR
100 VA=9*16↑3+14:VN=9*16↑3+13
110 POKEVA+1,25
119 REM
120 REM SET UP DISPLAY
121 REM
130 PRINT"■SQQQQQQQQQQQQQQQQQQQQQ";
140 PRINT"S──────────────────────■ΠΠΠΠΠ "
150 FORA=KRTOKR+22*23:POKEA,0:NEXT:A=0
160 PRINT
170 PRINT"■  8 "
180 PRINT"9:::; "
190 PRINT"ΠΠΠ";
200 GOSUB1180
209 REM
210 REM READ KEYBOARD
211 REM
220 IFPEEK(197)=34THENGOSUB590:A=A+1:GOTO530
230 IF PEEK(197)<>48 THEN 260
240 POKEVA,0:POKEVN,0
250 PRINT"■□";:GOTO1230
260 IFPEEK(197)=33THENGOSUB590:A=A-1:GOTO530
270 IFNOT(PEEK(197)=35ANDB=0)THEN300
280 B=1:C=8078+A:L=1:P=15:POKEVN,158
290 POKEC,63:POKEC+OF,2:GOTO370
300 IFL=0THEN320
310 POKEVA,P:P=P-1:IFP=-1THENL=0:POKEVN,0
320 IFB=0THENPOKE36878,0:GOTO 370
330 POKEC,32:C=C-22
340 IFPEEK(C)=60ORPEEK(C)=61THEN710
350 IFC<7746THENB=0:GOTO370
360 POKEC,63
370 IFD=0THEN1030
380 IFD=0THEN430
390 POKEE,32:POKEE-1,32:POKEE-2,32:K=K+1
400 IFE=ITHEND=0:GOTO430
410 E=E+J:POKEE,61:POKEE-1,60
```

```
420 IFJ=1THEN460
430 IFNOT(INT((8186-E)/22)=22-K-AANDF=0)THEN450
440 F=1:G=E+21:M=21
450 GOTO480
460 IFNOT(INT((8098-E)/22)=A-KANDF<>0)THEN480
470 F=1:G=E+23:M=23
480 IFF=0THEN220
490 POKEG,32:G=G+M
500 IFPEEK(G)<>32THEN850
510 IFG>VR+22*21THENF=0:GOTO630
520 POKEG,62:GOTO220
530 IFA<0THENA=0
540 IFA>16THENA=16
544 REM
545 REM MOVE TANK
546 REM
550 PRINT
560 PRINTTAB(A)"  8"
570 PRINTTAB(A)"9:::;"
580 PRINT"TTT";:GOTO270
590 PRINT
600 PRINTTAB(A)"     "
610 PRINTTAB(A)"      "
620 PRINT"TTT";:RETURN
624 REM
625 REM EXPLODE BOMB ON GROUND
626 REM
630 POKEG,194:POKEG+1,206:POKEG-1,205
640 POKEG-20,174:POKEG-21,174:POKEG-22,174
650 POKEG-23,174:POKEG-24,174
656 REM
660 FORAA=1TO100:NEXT
664 REM
665 REM CLEAR BOMB EXPLOSION
670 POKEG,32:POKEG+1,32:POKEG-1,32
680 POKEG-20,32:POKEG-21,32:POKEG-22,32
690 POKEG-23,32:POKEG-24,32
700 GOTO220
704 REM
705 REM EXPLODE UFO
706 REM
710 POKE 30720+C,5
720 POKE 30720+C+1,5:POKE 30720+C-1,5
730 L=0
740 POKEVN,228
750 FORGG=15TO0STEP-1:POKEVA,GG+128
760 FORGH=1TO70:NEXT:NEXT
770 B=0:D=0
774 REM
775 REM CLEAR UFO EXPLOSION
776 REM
```

```
780 POKE 30720+C,0:POKE C,32
790 POKE 30720+C+1,0:POKE C+1,32
800 POKE 30720+C-1,0:POKE C-1,32
810 GOTO820
820 PRINT"▓▓▓";
830 DU=DU+1:GOSUB1180
840 GOTO220
844 REM
845 REM EXPLODE TANK
846 REM
850 POKEVN,128:L=0
860 A=A:FORKL=1TO200STEP3:POKEVA,15-INT(KL/13)+128
870 PRINTTAB(A)"
880 PRINTTAB(A)"▓  8"
890 PRINTTAB(A)"9:::;"
900 PRINT"TT";
910 FOR I=1 TO 50:NEXT I
920 PRINTTAB(A)"▓  8
930 PRINTTAB(A)"9:::;"
940 PRINT"TTT";
950 FOR I=1 TO 50:NEXT I
960 NEXT
970 PRINTTAB(A)"    "
980 PRINTTAB(A)"    "
990 PRINTTAB(A)"      ";
1000 PRINT"TTT";
1010 DT=DT+1:GOSUB1180
1020 F=0:A=0:GOTO160
1030 D=1:E=7702+INT(RND(1)*14)*22+88:I=E-20:K=0
1040 J=-1:IFRND(1)>.5THENE=E-21:I=E+20:J=1
1050 GOTO380
1059 REM
1060 REM PRINT INSTRUCTIONS
1061 REM
1070 PRINT"▓▓*** TANK VERSUS UFO***
1080 PRINT"▓YOU ARE THE DRIVER OF
1090 PRINT"  A TANK.
1100 PRINT"▓SHOOT THE UFO'S WITH.
1110 PRINT"▓▓ Z = ▓LEFT▓ MOVEMENT
1120 PRINT"▓ C = ▓RIGHT▓ MOVEMENT
1130 PRINT"▓ B = ▓FIRE▓
1140 PRINT"▓ Q = ▓QUIT▓
1150 PRINT"▓▓▓▓START WITH ANY KEY
1160 GETA$:IFA$=""THEN1160
1170 GOSUB 1260:POKE36869,255:RETURN
1180 PRINT"▓▓▓==> TANK SCORE :"DU
1190 PRINT"▓==> UFO  SCORE :"DT
1200 PRINT"▓▓──────────────────────
1210 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓
1220 RETURN
```

```
1230 FORI=1TO50:GETA$:NEXT:POKE36879,27
1240 POKE 36869,240:POKE 56,PEEK(56)+2:END
1249 REM
1250 REM SET UP GRAPHIC CHARACTERS
1251 REM
1260 FOR I=1 TO 9
1270 READ X
1280 FOR J=0 TO 7
1290 READ K
1300 POKE X+J,K
1310 NEXT J,I
1320 RETURN
1324 REM
1325 REM CHARACTER DATA
1326 REM
1330 DATA 7424,0,0,0,0,0,0,0,0
1340 DATA 7648,64,32,31,25,57,127,238,56
1350 DATA 7656,2,4,248,152,156,254,119,28
1360 DATA 7664,68,108,124,56,56,56,16,0
1370 DATA 7672,16,56,56,56,124,108,68,0
1380 DATA 7616,24,24,24,24,60,126,255,255
1390 DATA 7624,15,31,31,63,63,31,31,15
1400 DATA 7632,255,255,255,255,255,255,255,255
1410 DATA 7640,240,248,248,252,252,248,248,240
READY.
```

LANDMINE

DESCRIPTION

This is a game of increasing difficulty. The object being to get your man home safely through the minefield. The computer offers help by warning you where the mines are up to two moves away from the man in any direction. The warning given consists of a high pitched alarm should there be a mine one move away from the man and a lower pitched alarm if the mine is two moves away.
Once the player has made it safely to his home or has been blown up the player must hit the 'E' to end the game. Any other key and the game is reset for a new go. Each time the player elects to play again two more mines are added to the minefield to increase the difficulty of play.

EQUIPMENT REQUIRED

A basic VIC-20 with 3K of expansion memory.

RUNNING THE PROGRAM

To move the man in the required direction use the number keys, thus:

7	8	9
4		6
1	2	3

Remember to listen for the landmine warnings!

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

40	Sets up number of mines at 10 initially.
50	Limits top of memory.
60	Array for storing grass and mines.
110-150	Sets up grass in all positions in the array.
170-240	Chooses random positions for mines.
250-260	Sets grass in top left, house in bottom right position.
290-420	Displays minefield.

470-500 Inputs moves.

520-670 Increases or decreases X and Y positions according to move input.

680-700 Check for validification of move.

710-720 Displays white grass path taken by man.

730-810 Check for mines near man.

860-890 Displays mine field successfully crossed.

900-980 Displays mine stepped on.

990-1020 Input for new go, E for end of the game.

1030-1040 Increases the number of mines by 2 and sets up screen to replay game.

1090-1190 Displays routine.

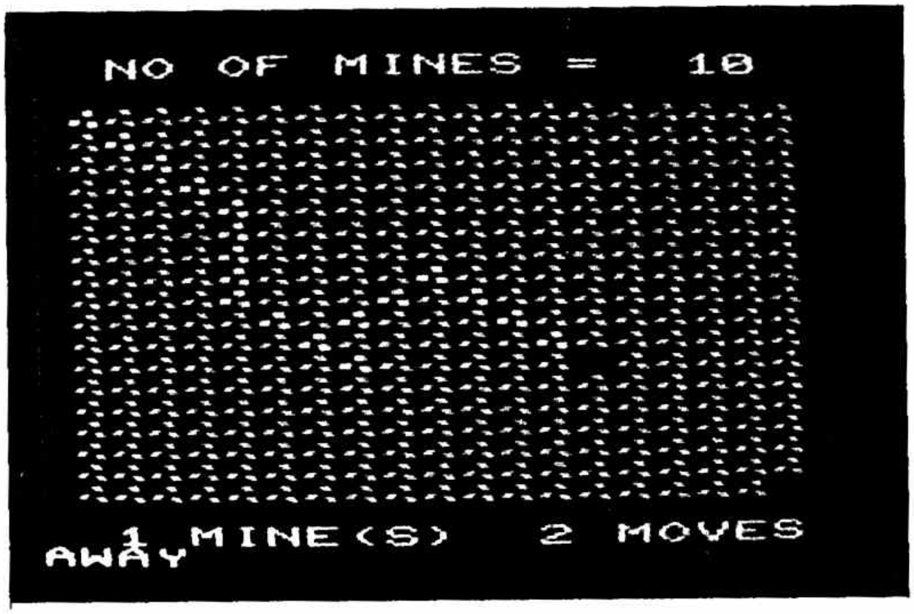
1220-1360 Authorises warning to man that mines are in the vicinity.

1435-1500 Sets up graphics characters.

1510-1590 Data for characters.

1610-1740 Displays explosion.

1760-1940 Input move and sounds alarm if mines are close.



```
NO OF MINES = 10

1 MINE(S)  2 MOVES
AWAY
```

```
10 REM LANDMINE
20 REM ****************************************
30 REM
40 POKE 900,10:REM NO MINES = 10 INITIALLY
50 POKE 56,PEEK(56)-1:CLR:POKE 56,PEEK(56)+1
60 DIM A$(19,18)
70 POKE 36879,42:PRINT"■"
80 GOSUB 1440
90 POKE 56,PEEK(56)-1
100 POKE 36869,255
110 FOR I=1 TO 19
120 FOR J=1 TO 18
130 A$(I,J)="■="
140 NEXT J
150 NEXT I
159 REM
160 REM SET UP MINES AT RANDOM
161 REM
170 FOR I=1 TO PEEK(900)
180 X%=RND(1)*19+1
190 Y%=RND(1)*18+1
200 IF A$(X%,Y%)="■>" THEN 180
210 IF X%=1 AND Y%=1 THEN 180
220 IF X%=19 AND Y%=18 THEN 180
230 A$(X%,Y%)="■>"
240 NEXT I
250 A$(1,1)="■="
260 A$(19,18)="■8"
270 PRINT"■■■  NO OF MINES = ";PEEK(900)
279 REM
280 REM PRINT MINEFIELD ON SCREEN
281 REM
290 CL=2:NO=19:Z$="■=":T$="H"
300 FOR RW=2 TO 19
310 GOSUB 1050
320 NEXT RW
330 CL=0:NO=21:RW=0:Z$="■■▒■":T$="H"
340 GOSUB 1050
350 RW=20
360 GOSUB 1050
370 CL=0:NO=19:RW=1:Z$="■■▒■":T$="V"
380 GOSUB 1050
390 CL=21
400 GOSUB 1050
410 X%=1:Y%=1
420 CL=20:RW=19:T$="H":NO=1:Z$="■8":GOSUB 1050
430 PRINT"■"
440 GOTO 740
450 GOSUB 1370
460 PRINT"■?";
470 GOSUB 1750
```

```
480 GG=VAL(GG$)
490 IF GG<1 OR GG>9 THEN 450
500 AA%=X%:BB%=Y%
510 ON GG GOTO 520,540,560,580,600,610,630,650,670
520 XX%=X%-1:YY%=Y%+1:CR$="▌▐◙"
530 GOTO 680
540 XX%=X%:YY%=Y%+1:CR$="◙"
550 GOTO 680
560 XX%=X%+1:YY%=Y%+1:CR$="▐◙"
570 GOTO 680
580 XX%=X%-1:YY%=Y%:CR$="▌▐"
590 GOTO 680
600 GOTO 450
610 XX%=X%+1:YY%=Y%:CR$="▐▌"
620 GOTO 680
630 XX%=X%-1:YY%=Y%-1:CR$="▜▌▐"
640 GOTO 680
650 XX%=X%:YY%=Y%-1:CR$="▜"
660 GOTO 680
670 XX%=X%+1:YY%=Y%-1:CR$="▜▐▌"
680 IFXX%<1 OR XX%>19 THEN 450
690 IFYY%<1 OR YY%>18 THEN 450
700 X%=XX%:Y%=YY%
710 CL=1+AA%:RW=1+BB%:T$="H":Z$="≢◙"
720 GOSUB 1050:A$(AA%,BB%)="≢"
729 REM
730 REM NOW CHECK FOR MINES IN VICINITY
731 REM
740 IF A$(X%,Y%)="▥D" THEN 900
750 A$(X%,Y%)="■?"
760 IF X%=19 AND Y%=18 THEN 860
769 REM
770 REM CHECK FOR 1 MOVE AWAY
771 REM
780 MS%=1:GOSUB 1200
790 IF MN<>0 THEN 820
799 REM
800 REM CHECK FOR 2 MOVES AWAY
801 REM
810 MS%=2:GOSUB 1200
819 REM
820 REM RESET CURSOR WITHIN MINEFIELD
821 REM
830 GOSUB 1370
840 GOTO 450
849 REM
850 REM MINEFIELD CROSSED SUCCESSFULLY
851 REM
860 CL=0:NO=1:RW=21:T$="H"
870 Z$="◪≢MINEFIELD SUCCESSFULLY CROSSED  ▀"
880 GOSUB 1050
```

```
890 A$(19,18)="█8":GOTO 930
899 REM
900 REM LANDMINE EXPLODES
901 REM
910 GOSUB 1600
920 A$(X%,Y%)="▛9"
930 FOR Y=1 TO 18
940 X%=1:Y%=Y:GOSUB 1370
950 FOR X=1 TO 19
960 PRINT A$(X,Y);
970 NEXT X
980 NEXT Y
990 GET GG$:IF GG$="" THEN 990
1000 IF GG$<>"E" THEN 1030
1010 POKE 56,PEEK(56)+1:POKE 36869,240
1020 POKE 36879,27:PRINT"▚▛":END
1030 POKE 900,PEEK(900)+2
1040 POKE 56,PEEK(56)+1:GOTO 50
1049 REM
1050 REM SUBROUTINE TO DRAW LINE ON SCREEN
1060 REM CL=START COLUMN,RW=START ROW
1070 REM NO=NO OF CHARS,T$= V(VERT),H(HORIZ)
1080 REM Z$=TEXT TO PRINT
1081 REM
1090 PRINT"▚"
1100 IF CL=0 THEN 1120
1110 FOR I=1 TO CL-1:PRINT"▐";:NEXT I
1120 IF RW=0 THEN 1140
1130 FOR I=1 TO RW-1:PRINT"▚";:NEXT I
1140 IF T$="H" THEN I$=""
1150 IF T$="V" THEN I$="▚█▌"
1160 FOR I=1 TO NO
1170 PRINTZ$;I$;
1180 NEXT I
1190 RETURN
1199 REM
1200 REM CHECK FOR MINES IN VICINITY
1210 REM MS%=1 CLOSE SEARCH,MS%=2 WIDE SEARCH
1211 REM
1220 MN=0
1230 FOR I=X%-MS% TO X%+MS%
1240 FOR J=Y%-MS% TO Y%+MS%
1250 IF I<1 OR J<1 THEN 1280
1260 IF I>19 OR J>18 THEN 1280
1270 IF A$(I,J)="▛D" THEN MN=MN+1
1280 NEXT J
1290 NEXT I
1300 Z$="▚                           █"
1310 IF MN<=0 THEN 1340
1320 Z$="▚▜"+STR$(MN)+" MINE(S) "
1330 Z$=Z$+STR$(MS%)+" MOVES  AWAY█"
```

```
1340 CL=1:NO=1:RW=21:T$="H"
1350 GOSUB 1050
1360 RETURN
1369 REM
1370 REM CURSOR ADDRESS TO WITHIN MINEFIELD
1380 REM X%=X AXIS, Y%=Y AXIS
1381 REM
1390 PRINT"■";
1400 FOR I=1 TO X%+1:PRINT"■";:NEXT I
1410 FOR I=1 TO Y%+1:PRINT"■";:NEXT I
1420 PRINT"■";
1430 RETURN
1434 REM
1435 REM SET UP GRAPHICS CHARACTERS
1436 REM
1440 FOR I=1 TO 9
1450 READ X
1460 FOR J=0 TO 7
1470 READ K
1480 POKE X+J,K
1490 NEXT J,I
1500 RETURN
1504 REM
1505 REM CHARACTER DATA
1506 REM
1510 DATA 7424,0,0,0,0,0,0,0,0
1520 DATA 7656,0,12,12,0,96,102,6,0
1530 DATA 7664,60,66,153,165,165,153,66,60
1540 DATA 7672,28,28,8,62,8,20,34,0
1550 DATA 7624,153,90,60,255,255,60,90,153
1560 DATA 7632,146,84,56,254,56,84,146,0
1570 DATA 7640,0,0,0,24,24,0,0,0
1580 DATA 7648,0,0,0,0,0,0,0,0
1590 DATA 7616,20,44,68,254,68,68,124,0
1599 REM
1600 REM EXPLOSION
1601 REM
1610 POKE 36877,220
1620 FOR L= 15 TO 0 STEP -1
1630 POKE 36878,L
1640 IF L=15 THEN Z$="■K"
1650 IF L=11 THEN Z$="■;"
1660 IF L=7 THEN Z$="■:"
1670 IF L=3 THEN Z$="■9"
1680 CL=1+X%:RW=1+Y%:NO=1
1690 GOSUB 1050
1700 FOR M=1 TO 50:NEXT M
1710 NEXT L
1720 POKE 36877,0
1730 POKE 36878,0
1740 RETURN
```

```
1749 REM
1750 REM GET MOVE AND ALARM
1751 REM
1760 IF MN<>0 THEN 1790
1770 GET GG$:IF GG$="" THEN 1770
1780 RETURN
1790 IF MS%=2 THEN SH=180:SL=128:GOTO 1810
1800 SL=180:SH=235
1810 POKE 36878,15
1820 FOR M=SL TO SH STEP 2
1830 POKE 36876,M
1840 GET GG$:IF GG$<>"" THEN 1920
1850 FOR N=1 TO 10
1860 NEXT N
1870 NEXT M
1880 POKE 36876,0
1890 FOR M=1 TO 100
1900 NEXT M
1910 GOTO 1820
1920 POKE 36878,0
1930 POKE 36876,0
1940 RETURN
READY.
```

SPACEWAR

DESCRIPTION

The game is in two parts. The first sets up the parameters and prints the instructions. The second plays the actual game. The first part must therefore be loaded and run before the second part. When part one is run, it will display two pages of instructions separated by 'hit any key'. The next page will input skill level from 1-9 and then either the option of sun's gravity, black hole, or no gravity. The second part plays the game and keeps count of how many of the users spacecraft and the computers have been destroyed. The game is over when five spacecraft of either side have been destroyed. A new game will start after a reasonable pause. Written in Basic, the game is rather slow.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.
Joystick.

RUNNING THE PROGRAM

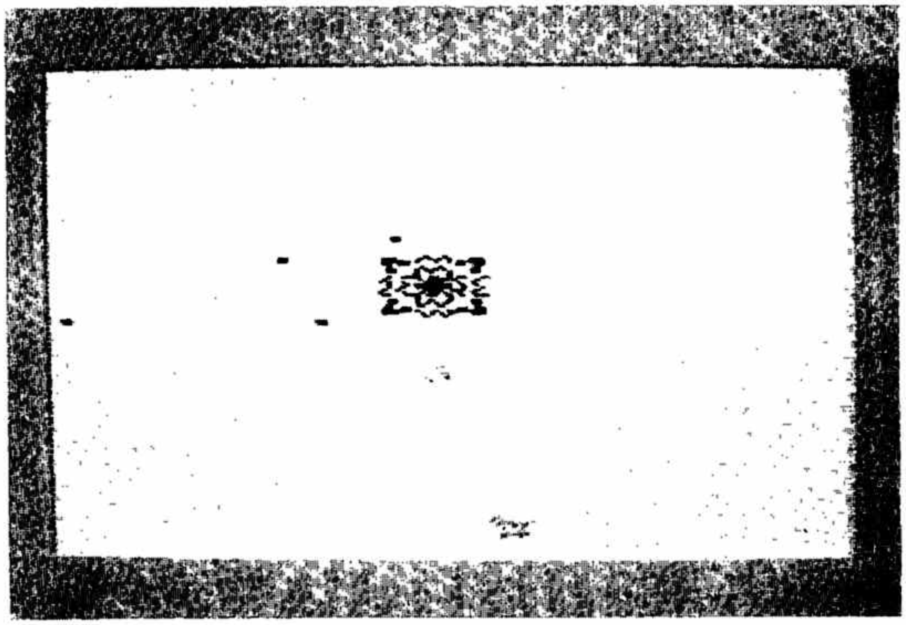
Load in part one and run. After a slight pause the instructions are displayed. When the instructions have been displayed and the two parameters have been input, the program will end. Load in part two, run, the actual game will then start. Use the joystick to change the direction of the ship, try to destroy the computer's ship. Use the fire button on the joystick to shoot at the computer's ship. Hitting the space bar at any time in the game will cause the player's ship to go into hyperspace.

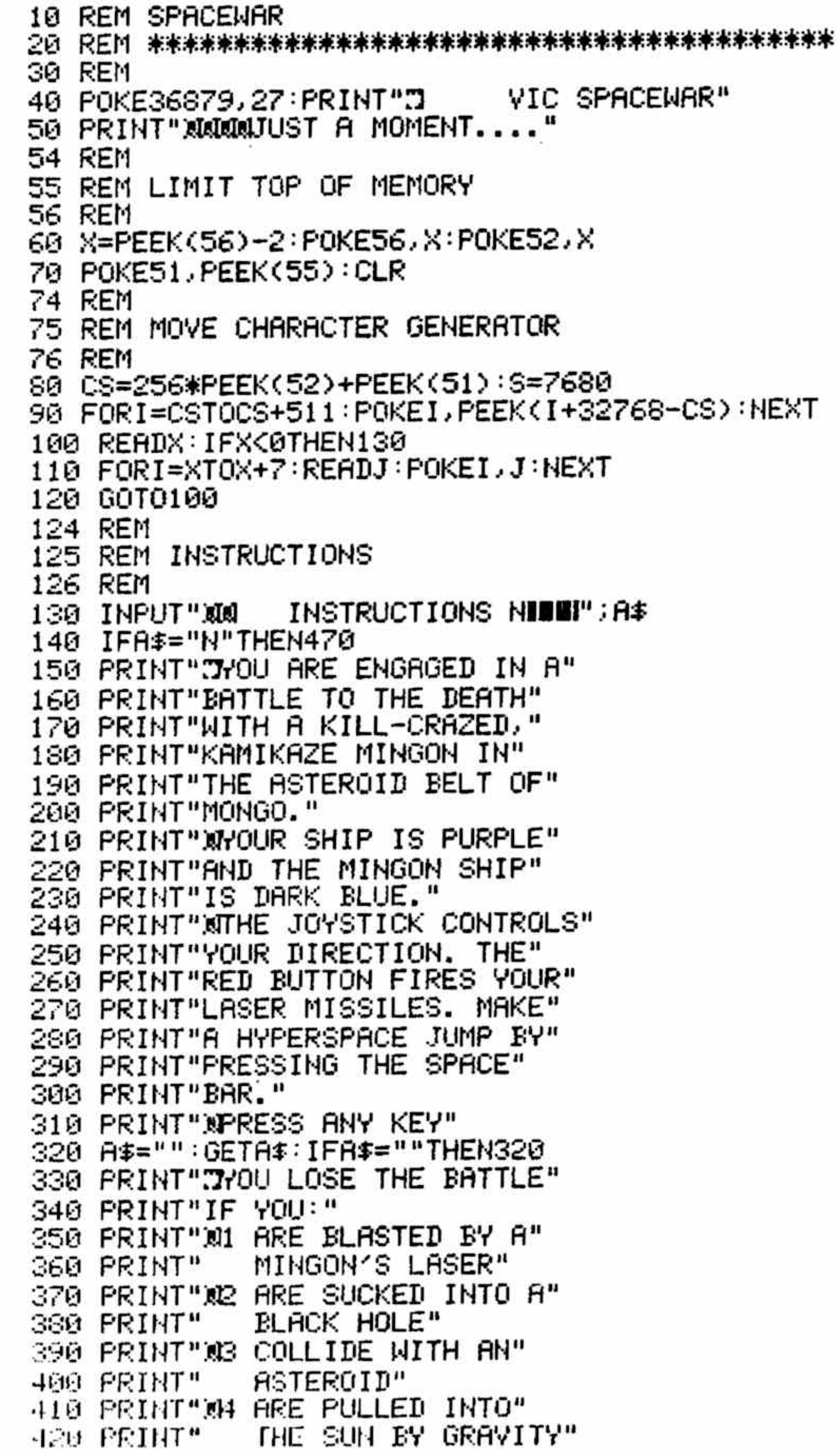
PROGRAM STRUCTURE

The lines of interest in part one are as follows.

60-70 Limit top of memory.
80-120 Move the character generator and set up other graphics characters.
130-460 Display the instructions.

470-520 Input skill level.
530-600 Input option.
610-620 POKE last two parameters into top bytes of memory. The locations are the two locations directly before the screen memory.
630-830 Character data.



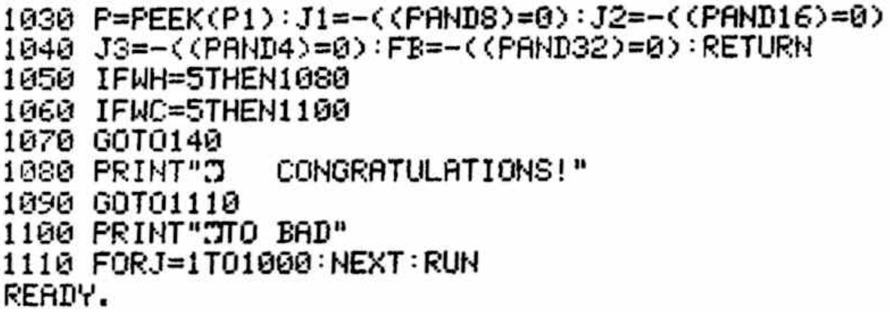


```
10 REM SPACEWAR
20 REM ****************************************
30 REM
40 POKE36879,27:PRINT"⊐      VIC SPACEWAR"
50 PRINT"QQQQJUST A MOMENT...."
54 REM
55 REM LIMIT TOP OF MEMORY
56 REM
60 X=PEEK(56)-2:POKE56,X:POKE52,X
70 POKE51,PEEK(55):CLR
74 REM
75 REM MOVE CHARACTER GENERATOR
76 REM
80 CS=256*PEEK(52)+PEEK(51):S=7680
90 FORI=CSTOCS+511:POKEI,PEEK(I+32768-CS):NEXT
100 READX:IFX<0THEN130
110 FORI=XTOX+7:READJ:POKEI,J:NEXT
120 GOTO100
124 REM
125 REM INSTRUCTIONS
126 REM
130 INPUT"QQ   INSTRUCTIONS N■■■";A$
140 IFA$="N"THEN470
150 PRINT"⊐YOU ARE ENGAGED IN A"
160 PRINT"BATTLE TO THE DEATH"
170 PRINT"WITH A KILL-CRAZED,"
180 PRINT"KAMIKAZE MINGON IN"
190 PRINT"THE ASTEROID BELT OF"
200 PRINT"MONGO."
210 PRINT"QYOUR SHIP IS PURPLE"
220 PRINT"AND THE MINGON SHIP"
230 PRINT"IS DARK BLUE."
240 PRINT"QTHE JOYSTICK CONTROLS"
250 PRINT"YOUR DIRECTION. THE"
260 PRINT"RED BUTTON FIRES YOUR"
270 PRINT"LASER MISSILES. MAKE"
280 PRINT"A HYPERSPACE JUMP BY"
290 PRINT"PRESSING THE SPACE"
300 PRINT"BAR."
310 PRINT"QPRESS ANY KEY"
320 A$="":GETA$:IFA$=""THEN320
330 PRINT"⊐YOU LOSE THE BATTLE"
340 PRINT"IF YOU:"
350 PRINT"Q1 ARE BLASTED BY A"
360 PRINT"   MINGON'S LASER"
370 PRINT"Q2 ARE SUCKED INTO A"
380 PRINT"   BLACK HOLE"
390 PRINT"Q3 COLLIDE WITH AN"
400 PRINT"   ASTEROID"
410 PRINT"Q4 ARE PULLED INTO"
420 PRINT"   THE SUN BY GRAVITY"
```

```
430 PRINT"▒5 ARE RAMMED BY THE"
440 PRINT"   THE MINGON SHIP"
450 PRINT"▒PRESS ANY KEY"
460 A$="":GETA$:IFA$=""THEN460
470 PRINT"▒  ENTER SKILL LEVEL"
480 PRINT"▒▒     ▒1▒-TRIVIAL"
490 PRINT"▒        TO"
500 PRINT"▒     ▒9▒-IMPOSSIBLE"
510 A$="":GETA$:IFA$=""THEN510
520 SL=VAL(A$):IFSL<1ORSL>9THEN470
530 PRINT"▒ SPACEWAR OPTIONS"
540 PRINT"▒▒     ▒S▒UN'S GRAVITY"
550 PRINT"▒     ▒B▒LACK HOLE"
560 PRINT"▒     ▒N▒O GRAVITY"
570 PRINT"▒▒  SELECT ONE OPTION":G=0
580 A$="":GETA$:IFA$=""THEN580
590 IFA$="S"THENG=1
600 IFA$="B"THENG=2
610 POKES-2,SL:POKES-1,G
620 END
630 DATA7168,192,240,127,103,34,54,63,50
640 DATA7176,4,14,62,227,227,62,14,4
650 DATA7184,50,63,54,34,103,127,240,192
660 DATA7192,24,24,60,36,102,231,126,24
670 DATA7208,24,126,231,102,36,60,24,24
680 DATA7216,3,15,254,230,68,108,252,76
690 DATA7224,32,112,124,199,199,124,112,32
700 DATA7232,76,252,108,68,230,254,15,3
710 DATA7240,0,0,12,28,56,48,0,0
720 DATA7248,0,0,48,56,28,12,0,0
730 DATA7256,0,24,24,24,24,24,0,0
740 DATA7264,0,0,0,62,62,0,0,0
750 DATA7296,0,112,126,102,32,48,48,0
760 DATA7304,228,18,37,68,36,18,33,198
770 DATA7312,0,48,48,32,102,126,112,0
780 DATA7320,68,170,145,0,34,85,137,129
790 DATA7328,153,90,60,255,255,60,90,153
800 DATA7336,129,137,85,34,0,145,170,68
810 DATA7344,0,14,126,102,4,12,12,0
820 DATA7352,198,33,18,36,68,37,18,228
830 DATA7360,0,12,12,4,102,126,14,0,-1
READY.
```

```
10 REM SPACEWAR 2
20 REM ****************************************
30 REM
40 C=22:R=23:S=7680:A=30720:DD=37154
50 P1=37151:P2=37152:SL=PEEK(S-2)
60 POKE36879,27:T=3:CY=4:CE=6:CA=0
70 CS=7:CQ=0:CM=0:X=-1:G=PEEK(S-1)
80 V=36878:S1=V-2:S2=V-1:DIMDC%(2,2),OC%(2,2)
90 FORI=0TO2:FORJ=0TO2:X=X+1
100 DC%(I,J)=X:OC%(I,J)=X+16:NEXTJ,I
110 DEFFNA(Z)=S+X+C*Y
120 DEFFNB(Z)=PEEK(FNA(Z))
130 DEFFNR(Z)=INT(RND(1)*Z)
140 POKEV-9,255:PRINT"♥":X=S+A
150 FORI=XTOX+505:POKEX,T:NEXT
160 POKE36879,62:FORI=1TO3+2*SL
170 X=FNR(C):Y=FNR(R)
180 POKEFNA(0),46:POKEFNA(0)+A,CA:NEXT
190 IFG<>1THEN220
200 X=11:Y=12:POKEFNA(0),20
210 POKEFNA(0)+A,CS:SX=X:SY=Y
220 IFG=2THENSX=FNR(C):SY=FNR(R)
230 X=FNR(C):Y=FNR(R):IFFNB(0)<>32THEN230
240 D=X:E=Y:Q=FNA(0):U=-1:O=FNR(3)-1
250 POKEFNA(0),DC%(U+1,O+1):POKEFNA(0)+A,CY
260 X=FNR(C):Y=FNR(R):IFFNB(0)<>32THEN260
270 H=X:L=Y:K=FNA(0):M=1:N=FNR(3)-1
280 POKEFNA(0),DC%(M+1,N+1):POKEFNA(0)+A,CE
290 A$="":GETA$:IFA$=" "THEND=FNR(C):E=FNR(R)
300 GOSUB1010:IFFBTHENX=D:Y=E:PX=U:PY=O:GOSUB680
310 B=0:F=0:IFJ0THENB=1
320 IFJ2THENB=-1
330 IFJ1THENF=1
340 IFJ3THENF=-1
350 IFB=0ANDF=0THENB=U:F=O
360 U=B:O=F:IFG=0THEN410
370 B=SX-D:F=SY-E:J=SQR(B*B+F*F)
380 J=(20-J)/30:J=1-J*J
390 IFRND(1)<JTHEN410
400 B=SGN(B):F=SGN(F):D=D+B:E=E+F:GOTO420
410 D=D+U:E=E+O
420 IFE<0THENE=R
430 IFE>RTHENE=0
440 IFD>CTHEND=0
450 IFD<0THEND=C
460 X=D:Y=E:J=FNB(0):IFJ=32THEN480
470 IFJ=46ORJ=20ORFNA(0)=KTHEN820
480 IFNOT(X=SXANDY=SY)THEN510
490 A$="WERE SUCKED INTO     A BLACK HOLE!"
500 WC=WC+1:GOTO970
510 POKEQ,32:POKEQ+A,T:Q=FNA(0)
```

```
520 POKEQ,DC%(U+1,O+1):POKEQ+A,CY
530 J=0:IFFNR(9)>SLTHEN560
540 M=D-H:N=E-L:M=SGN(M):N=SGN(N)
550 IFM=0ORN=0THENJ=1
560 H=H+M:L=L+N:IFH<0THENH=C
570 IFH>CTHENH=0
580 IFL>RTHENL=0
590 IFL<0THENL=R
600 X=H:Y=L
610 IFFNB(0)=32THEN630
620 M=FNR(3)-1:N=FNR(3)-1:GOTO560
630 POKEK,32:POKEK+A,T:K=FNA(0)
640 POKEK,DC%(M+1,N+1):POKEK+A,CE
650 IFNOT(RND(1)<.1OR(J=1ANDFNR(9)<SL))THEN670
660 PX=M:PY=N:GOSUB680
670 GOTO290
680 Z=PX*PY:POKEV,8:IFZ=1THENJ=10
690 IFZ=-1THENJ=9
700 IFZ=0ANDPX=0THENJ=11
710 IFZ=0ANDPY=0THENJ=12
720 FORI=1TO10:X=X+PX:Y=Y+PY:POKES2,230-I
730 IFI<>1THENPOKEZ,32:POKEZ+A,T:IFX>CTHENX=0
740 IFX<0THENX=C
750 IFY>RTHENY=0
760 IFY<0THENY=R
770 B=FNB(0):IFB=32THEN800
780 IFNOT(B=46ORB=20ORFNA(0)=KORFNA(0)=Q)THEN800
790 I=10:NEXT I:GOTO 820
800 Z=FNA(0):POKEZ,J:POKEZ+A,CM:NEXTI
810 POKEZ,32:POKEZ+A,T:POKEV,0:RETURN
820 POKES2,230:SC=X-1:IFSC<0THENSC=0
830 FC=X+1:IFFC>CTHENFC=C
840 SR=Y-1:IFSR<0THENSR=0
850 FR=Y+1:IFFR>RTHENFR=R
860 FORX=SCTOFC:FORY=SRTOFR:J=OC%(X-SC,Y-SR)
870 POKEFNA(0),J:POKEFNA(0)+A,CQ:NEXTY,X
880 POKES1,220:FORJ=15TO0STEP-1
890 POKEV,J:FORJ1=1TO50:NEXTJ1,J
900 POKEV,0:FORX=SCTOFC:FORY=SRTOFR
910 POKEFNA(0),32:POKEFNA(0)+A,T:NEXTY,X
920 IFPEEK(Q)<>32THEN940
930 A$="WERE VAPORIZED!":WC=WC+1:GOTO970
940 IFPEEK(K)<>32THEN960
950 A$="TRIUMPHED!":WH=WH+1:GOTO970
960 RETURN
970 POKEV-9,240:PRINT"]QQQYOU   ";A$
980 PRINT"QQSCORE: VIC";WC;"  YOU";WH
990 FORJ=1TO500:NEXT:GOTO1050
1000 END
1010 POKEDD,127:P=PEEK(P2)AND128
1020 J0=-(P=0):POKEDD,255
```



```
1030 P=PEEK(P1):J1=-((PAND8)=0):J2=-((PAND16)=0)
1040 J3=-((PAND4)=0):FB=-((PAND32)=0):RETURN
1050 IFWH=5THEN1080
1060 IFWC=5THEN1100
1070 GOTO140
1080 PRINT"♥   CONGRATULATIONS!"
1090 GOTO1110
1100 PRINT"♥TO BAD"
1110 FORJ=1TO1000:NEXT:RUN
READY.
```

JOYSTICK TEST

DESCRIPTION

One valuable addition to the VIC is the joystick. It can make game playing much more enjoyable, and often easier.

This program enables the user to check the position of the joystick as it is moved through the four cardinal compass points, and to see when the fire button is being pressed.

The program can also be used as a short input routine within a main program, to see which direction the joystick is currently being pointed in, and again to check for depression of the fire button.

This program will only work on a simple paddle switch joystick. To gain finer control you would require a potentiometer joystick, but programming for that is beyond the scope of this book.

EQUIPMENT REQUIRED

Basic VIC-20 plus joystick.

RUNNING THE PROGRAM

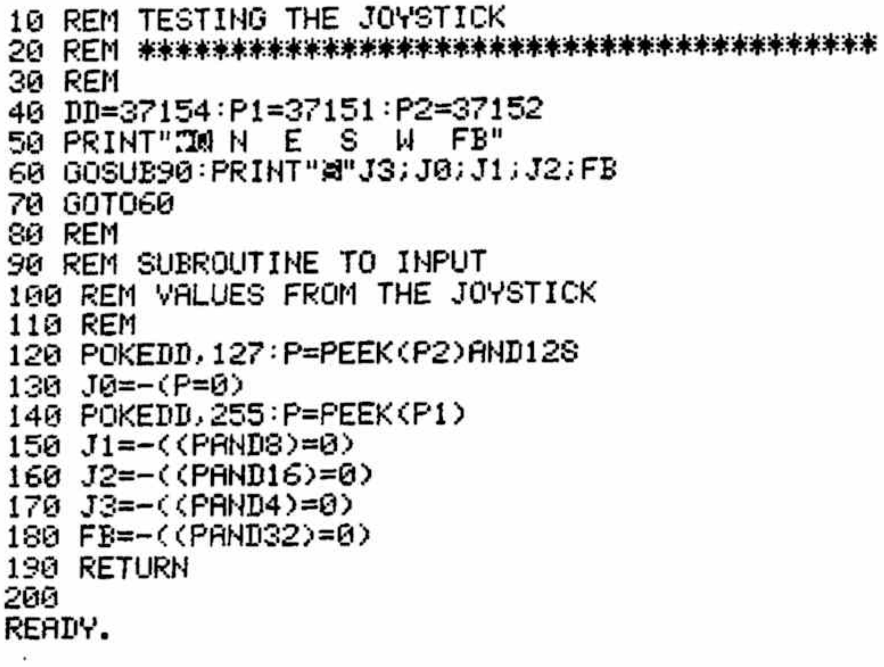
The data direction register, DD, is defined in line 40, along with variables for the output registers (P1 and P2).

When run, the program displays the cardinal compass points, together with an F (for fire!) at the top of the screen, and underneath that a value indicating which way the joystick had been moved: a value of zero would indicate off, and a value of one, on.

The routine in lines 120 to 180 checks for current direction from the output register, having previously set it back to 'normal' i.e. the fire button is not depressed, and the joystick is stationary in a vertically upright position.

PROGRAM STRUCTURE

10 - 30	Message and title of program line
40	Set up variables
50 - 60	Print display on screen
120 - 190	Joystick position reading subroutine



```
10 REM TESTING THE JOYSTICK
20 REM ****************************************
30 REM
40 DD=37154:P1=37151:P2=37152
50 PRINT"▚■ N  E  S  W  FB"
60 GOSUB90:PRINT"■"J3;J0;J1;J2;FB
70 GOTO60
80 REM
90 REM SUBROUTINE TO INPUT
100 REM VALUES FROM THE JOYSTICK
110 REM
120 POKEDD,127:P=PEEK(P2)AND128
130 J0=-(P=0)
140 POKEDD,255:P=PEEK(P1)
150 J1=-((PAND8)=0)
160 J2=-((PAND16)=0)
170 J3=-((PAND4)=0)
180 FB=-((PAND32)=0)
190 RETURN
200
READY.
```

DEFINE KEYS

DESCRIPTION

The four function keys on the right hand side of the VIC are probably one of the most neglected parts of the machine. Normally they are only used for simple messages e.g. Press F1 To Start The Game, normally to do anything else with them you would have to buy one of the expander cartridges from Commodore.

However, this short (160 byte) machine code routine, which sits at the top of memory, will allow you to assign a separate function up to 8 characters long to each of the 8 keys.

Take great care over the data statements in lines 100 to 220, as these are the heart of the program: if a mistake is made, the program will not work.

Each function is defined in lines 450 to 460, where the back arrow stands for Carriage Return. In other words, pressing function key 1 (using the examples as given) will list a program, pressing function key 2 will just print up the word GOSUB, and so on.

EQUIPMENT REQUIRED

Basic VIC-20.

RUNNING THE PROGRAM

Enter the program exactly as shown, and then SAVE a copy in the normal way before running it.

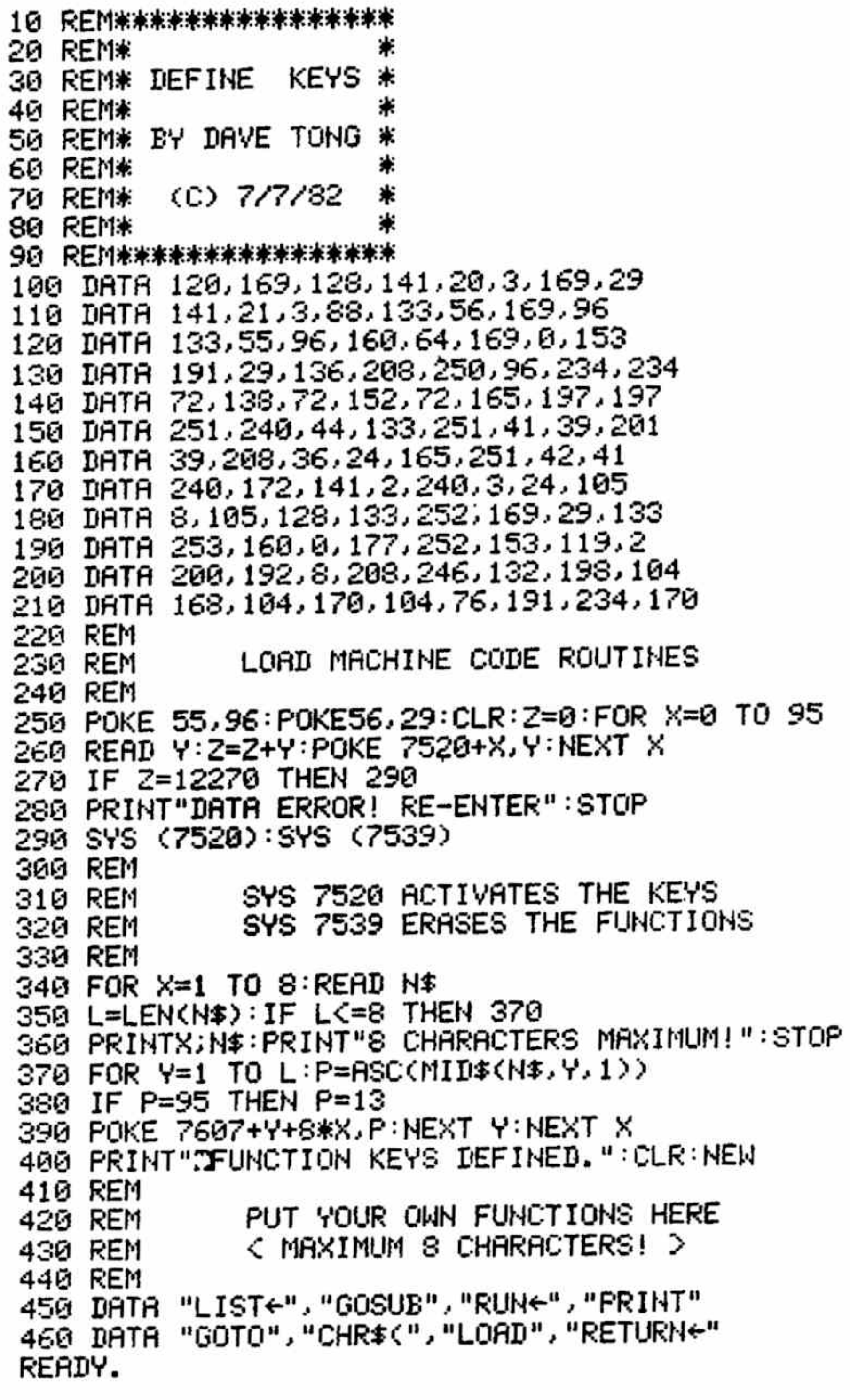
When RUN, provided you've entered everything correctly, the screen will clear and the message 'function keys defined' will appear, together with the friendly flashing cursor. Just pressing the keys will now bring the functions to life.

The program NEWs itself after running, so that you only lose 160 bytes from your VIC's Basic memory. To de-activate the function keys, hit RUN-STOP and RESTORE together. To get them back again, type SYS 7520.

If you wish to have different functions assigned, simply change the data statements in lines 450 and 460. Putting a left-arrow after the new keyword will prompt a carriage return automatically. To get more than 8 characters, use the keyword abbreviations as found in your manual e.g. P shift O for POKE, and so on.

PROGRAM STRUCTURE

10 - 90	REM statements and copyright message
100 - 210	Data statements containing machine code routine
250 - 260	POKE data into memory and calculate checksum
270	If checksum disagrees, print error message
290	Activate keys, erase previous functions
300 - 330	Instructions in REM statements
340 - 390	Perform function key definition
400	End, and perform NEW
450 - 460	Definitions for function keys



```
10 REM****************
20 REM*              *
30 REM* DEFINE  KEYS *
40 REM*              *
50 REM* BY DAVE TONG *
60 REM*              *
70 REM*  (C) 7/7/82  *
80 REM*              *
90 REM****************
100 DATA 120,169,128,141,20,3,169,29
110 DATA 141,21,3,88,133,56,169,96
120 DATA 133,55,96,160,64,169,0,153
130 DATA 191,29,136,208,250,96,234,234
140 DATA 72,138,72,152,72,165,197,197
150 DATA 251,240,44,133,251,41,39,201
160 DATA 39,208,36,24,165,251,42,41
170 DATA 240,172,141,2,240,3,24,105
180 DATA 8,105,128,133,252;169,29,133
190 DATA 253,160,0,177,252,153,119,2
200 DATA 200,192,8,208,246,132,198,104
210 DATA 168,104,170,104,76,191,234,170
220 REM
230 REM       LOAD MACHINE CODE ROUTINES
240 REM
250 POKE 55,96:POKE56,29:CLR:Z=0:FOR X=0 TO 95
260 READ Y:Z=Z+Y:POKE 7520+X,Y:NEXT X
270 IF Z=12270 THEN 290
280 PRINT"DATA ERROR! RE-ENTER":STOP
290 SYS (7520):SYS (7539)
300 REM
310 REM       SYS 7520 ACTIVATES THE KEYS
320 REM       SYS 7539 ERASES THE FUNCTIONS
330 REM
340 FOR X=1 TO 8:READ N$
350 L=LEN(N$):IF L<=8 THEN 370
360 PRINTX;N$:PRINT"8 CHARACTERS MAXIMUM!":STOP
370 FOR Y=1 TO L:P=ASC(MID$(N$,Y,1))
380 IF P=95 THEN P=13
390 POKE 7607+Y+8*X,P:NEXT Y:NEXT X
400 PRINT"ℑFUNCTION KEYS DEFINED.":CLR:NEW
410 REM
420 REM       PUT YOUR OWN FUNCTIONS HERE
430 REM       < MAXIMUM 8 CHARACTERS! >
440 REM
450 DATA "LIST←","GOSUB","RUN←","PRINT"
460 DATA "GOTO","CHR$(","LOAD","RETURN←"
READY.
```

U.S.A. SONG

DESCRIPTION

The program when running displays the U.S.A. flag and plays the 'Star Spangled Banner'.

EQUIPMENT REQUIRED

Basic VIC-20.

RUNNING THE PROGRAM

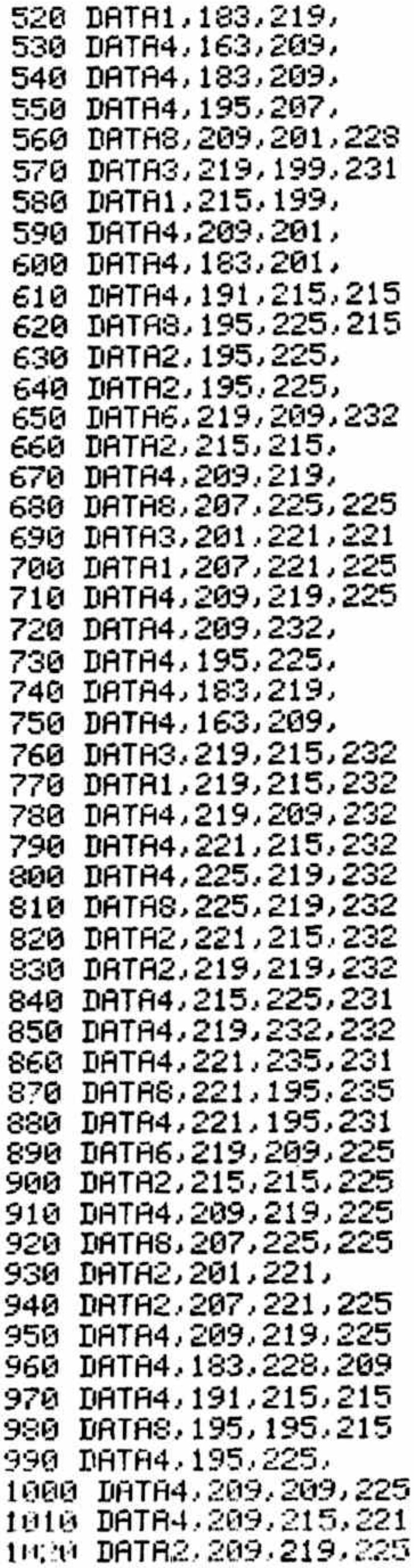
As no expansion is required it is just a matter of loading the program and typing run.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows.

60 Sets registers for three voices and indicates GOSUB to draw the flag.
70 Puts 10 in to volume register.
80-150 Main loop.
80 Read music data.
90 Set sound registers to 0.
100 Goto 160 at end of the music.
110 Pause.
120 Print value X.
130 Play Music.
140 Loop for duration.
150 Increase X and loop.
160 Check for break.
260-1270 Data for music configuration: Duration, Voice 1, Voice 2, and Voice 3.
1450-1560 Draw flag.

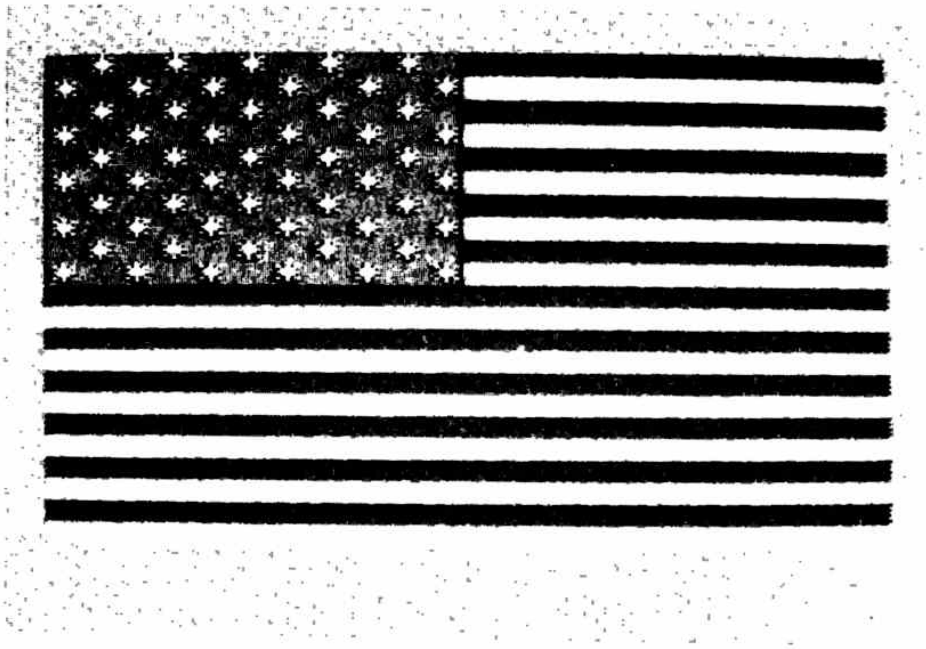
```
10 REM USA SONG
20 REM ****************************************
30 REM
40 REM PLAY MUSIC
50 REM
60 G1=36876:G2=36874:G3=36875:X=300:GOSUB1400
70 POKE36878,10
80 READDC,V1,V2,V3
90 POKEG1,0:POKEG2,0:POKEG3,0
100 IFDC=0THEN160
110 FORI=1TO50:NEXT
120 PRINT"SQQQQQQQQQQQQ"X
130 POKEG1,V1:POKEG2,V2:POKEG3,V3
140 FORI=1TO100*DC:NEXT
150 X=X+5:GOTO80
160 SYS65234
170 REM
180 REM SONGTABLE FOR
190 REM THE STAR SPANGLED
200 REM BANNER
210 REM
220 REM DURATION COUNT
230 REM IS -4- FOR A
240 REM QUARTER NOTE
250 REM
260 DATA6,195,225,
270 DATA1,183,219,
280 DATA4,163,209,
290 DATA4,183,209,
300 DATA4,195,207,
310 DATA8,209,201,228
320 DATA3,219,199,231
330 DATA1,215,199,
340 DATA4,209,201,
350 DATA4,183,201,
360 DATA4,191,215,215
370 DATA8,195,225,215
380 DATA2,195,225,
390 DATA2,195,225,
400 DATA6,219,209,232
410 DATA2,215,215,
420 DATA4,209,219,
430 DATA8,207,225,225
440 DATA3,201,221,221
450 DATA1,207,221,225
460 DATA4,209,219,225
470 DATA4,209,232,
480 DATA4,195,225,
490 DATA4,183,219,
500 DATA4,163,209,
510 DATA3,195,225,
```



```
520 DATA1,183,219,
530 DATA4,163,209,
540 DATA4,183,209,
550 DATA4,195,207,
560 DATA8,209,201,228
570 DATA3,219,199,231
580 DATA1,215,199,
590 DATA4,209,201,
600 DATA4,183,201,
610 DATA4,191,215,215
620 DATA8,195,225,215
630 DATA2,195,225,
640 DATA2,195,225,
650 DATA6,219,209,232
660 DATA2,215,215,
670 DATA4,209,219,
680 DATA8,207,225,225
690 DATA3,201,221,221
700 DATA1,207,221,225
710 DATA4,209,219,225
720 DATA4,209,232,
730 DATA4,195,225,
740 DATA4,183,219,
750 DATA4,163,209,
760 DATA3,219,215,232
770 DATA1,219,215,232
780 DATA4,219,209,232
790 DATA4,221,215,232
800 DATA4,225,219,232
810 DATA8,225,219,232
820 DATA2,221,215,232
830 DATA2,219,219,232
840 DATA4,215,225,231
850 DATA4,219,232,232
860 DATA4,221,235,231
870 DATA8,221,195,235
880 DATA4,221,195,231
890 DATA6,219,209,225
900 DATA2,215,215,225
910 DATA4,209,219,225
920 DATA8,207,225,225
930 DATA2,201,221,
940 DATA2,207,221,225
950 DATA4,209,219,225
960 DATA4,183,228,209
970 DATA4,191,215,215
980 DATA8,195,195,215
990 DATA4,195,225,
1000 DATA4,209,209,225
1010 DATA4,209,215,221
1020 DATA2,209,219,225
```

```
1030 DATA2,207,219,225
1040 DATA4,201,221,221
1050 DATA4,201,221,221
1060 DATA4,201,219,225
1070 DATA4,215,215,221
1080 DATA2,221,215,221
1090 DATA2,219,219,219
1100 DATA2,215,221,215
1110 DATA2,209,223,209
1120 DATA4,209,225,225
1130 DATA8,207,195,225
1140 DATA2,195,225,
1150 DATA2,195,221,
1160 DATA6,209,219,225
1170 DATA2,215,225,
1180 DATA2,219,235,
1190 DATA2,221,235,
1200 DATA8,225,237,237
1210 DATA2,209,228,228
1220 DATA2,215,227,223
1230 DATA6,219,225,232
1240 DATA2,221,225,232
1250 DATA7,215,225,231
1260 DATA14,209,232,225
1270 DATA,,,
1280 REM
1290 REM ************
1300 REM *INTERPRET *
1310 REM *YOUR  DATA*
1320 REM ************
1330 REM
1340 INPUTA:FORI=ATOA+95STEP5:PRINTI"D♠":NEXT:END
1350 REM
1360 REM ************
1370 REM *INTERPRET *
1380 REM *YOUR  DATA*
1390 REM ************
1400 REM
1410 REM ***********
1420 REM *DRAW FLAG*
1430 REM ***********
1440 REM
1450 PRINT"J";
1460 POKE36867,44
1470 FORI=1TO11
1480 PRINT"▨▨                                   "
1490 NEXT
1500 PRINT"▨";
1510 FORI=1TO5
1520 PRINT"▨▨ * * * * * "
1530 PRINT"J";
```

```
1540 PRINT"32* * * * * *"
1550 NEXT
1560 RETURN
READY.
```



DIGICLOCK

DESCRIPTION

A digital clock showing hours, minutes and seconds in large characters is displayed on the screen. The clock will display the correct time (entered by the user) until the machine is switched off. A program which could be modified to act as a stop watch or display any digital value.

EQUIPMENT REQUIRED

A basic unexpanded VIC-20.

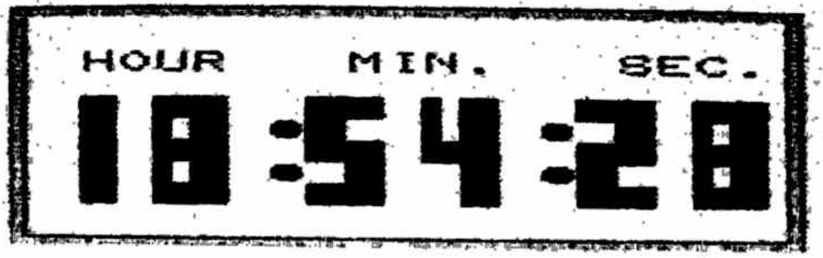
RUNNING THE PROGRAM

The program starts with a prompt to input the correct time, this is input in the format HHMMSS where H is the hour (using a 24 hour clock), M is the minutes, and S the seconds. As soon as the return key is pressed the clock is started. While the face is displayed it is possible to change the shown time at any point by simply entering return.

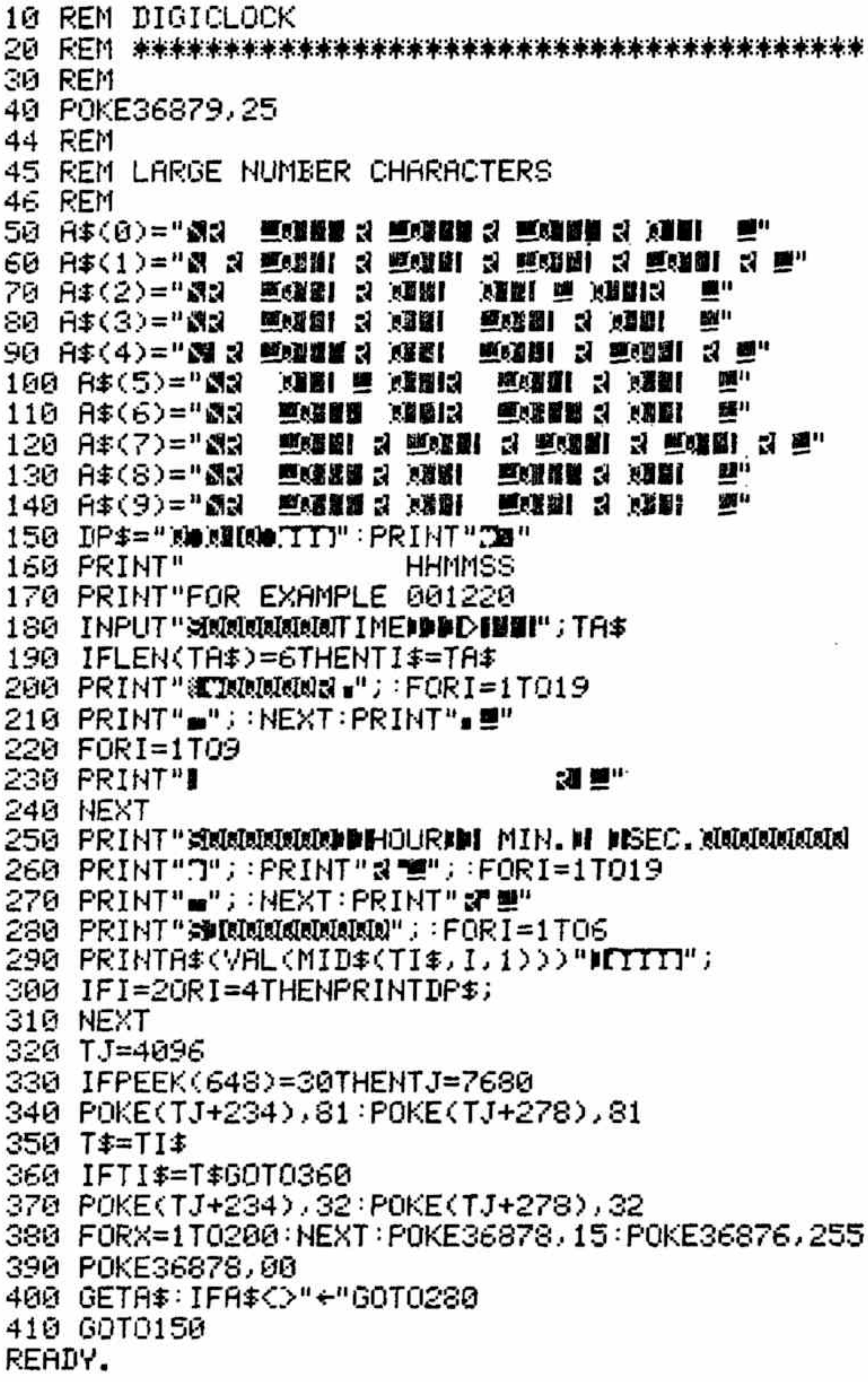
PROGRAM STRUCTURE

The more interesting lines for this program are as given.

40 Displays a white screen.
50-140 String array holding large numbers.
150 Two dots to separate hours from minutes and minutes from seconds. The dot separating minutes from seconds flashes every second.
160-190 Inputs the time.
200-270 Display around the numbers.
280-310 Prints enlarged numbers for time plus the dots.
320-330 Sets start of video map.
340-370 Makes dots between minutes and seconds flash.
380-390 Pause and beep each second.
400-410 Loop again or enter another time.



```
HOUR    MIN.    SEC.
18   :  54   :  28
```



```
10 REM DIGICLOCK
20 REM ****************************************
30 REM
40 POKE36879,25
44 REM
45 REM LARGE NUMBER CHARACTERS
46 REM
50 A$(0)="■■  ■■■■■ ■ ■■■■■ ■ ■■■■■ ■ ■■■   ■"
60 A$(1)="■ ■ ■■■■ ■ ■■■■ ■ ■■■■ ■ ■■■■ ■ ■"
70 A$(2)="■■  ■■■■ ■ ■■■  ■■■ ■ ■■■■  ■"
80 A$(3)="■■  ■■■■ ■ ■■■  ■■■■ ■ ■■■  ■"
90 A$(4)="■ ■ ■■■■■ ■ ■■■  ■■■■ ■ ■■■■ ■ ■"
100 A$(5)="■■  ■■■ ■ ■■■■  ■■■■ ■ ■■■  ■"
110 A$(6)="■■  ■■■■■ ■■■■  ■■■■■ ■ ■■■  ■"
120 A$(7)="■■  ■■■■ ■ ■■■■ ■ ■■■■ ■ ■■■■ ■ ■"
130 A$(8)="■■  ■■■■■ ■ ■■■  ■■■■■ ■ ■■■  ■"
140 A$(9)="■■  ■■■■■ ■ ■■■  ■■■■ ■ ■■■  ■"
150 DP$="■●■■●TTT":PRINT"■■"
160 PRINT"                    HHMMSS
170 PRINT"FOR EXAMPLE 001220
180 INPUT"■■■■■■■■TIME■■■▷■■■";TA$
190 IFLEN(TA$)=6THENTI$=TA$
200 PRINT"■■■■■■■■■▪";:FORI=1TO19
210 PRINT"▬";:NEXT:PRINT"▪■"
220 FORI=1TO9
230 PRINT"▮                         ■ ■"
240 NEXT
250 PRINT"■■■■■■■■■■■HOUR■■ MIN.■ ■SEC.■■■■■■■■
260 PRINT"■";:PRINT"■■";:FORI=1TO19
270 PRINT"▬";:NEXT:PRINT"■■"
280 PRINT"■■■■■■■■■■■";:FORI=1TO6
290 PRINTA$(VAL(MID$(TI$,I,1)))"■TTTT";
300 IFI=2ORI=4THENPRINTDP$;
310 NEXT
320 TJ=4096
330 IFPEEK(648)=30THENTJ=7680
340 POKE(TJ+234),81:POKE(TJ+278),81
350 T$=TI$
360 IFTI$=T$GOTO360
370 POKE(TJ+234),32:POKE(TJ+278),32
380 FORX=1TO200:NEXT:POKE36878,15:POKE36876,255
390 POKE36878,00
400 GETA$:IFA$<>"←"GOTO280
410 GOTO150
READY.
```

LEAP FROG

DESCRIPTION

Leap Frog encourages the player to develop mental agility. There are two sets of frogs, five of one colour and five of another. The player has to move them one at a time to the other side, until the positions are reversed. There are a minimal number of moves in which the game can be completed but praise is always given at the end of the game. An interesting feature of the program is the use of small size numerals, created like the frogs by using programmable characters.

EQUIPMENT REQUIRED

Basic VIC-20 plus a minimum of 3K memory expansion.

RUNNING THE PROGRAM

There are eleven numbered spaces displayed on the screen, above the five spaces on the right side are five green frogs, and above the five spaces on the left are five white frogs. A frog can only be moved into an empty space and can not jump over more than one frog to get there. A prompt requests the move, simply enter the number of the frog you wish to move, invalid moves will generate the relevant error messages. When you have completed the game the computer will display your score.

PROGRAM STRUCTURE

The lines of interest in this game are as follows.

40-50 Limits the memory.

70 GOSUB to routine for setting up the frogs characters.

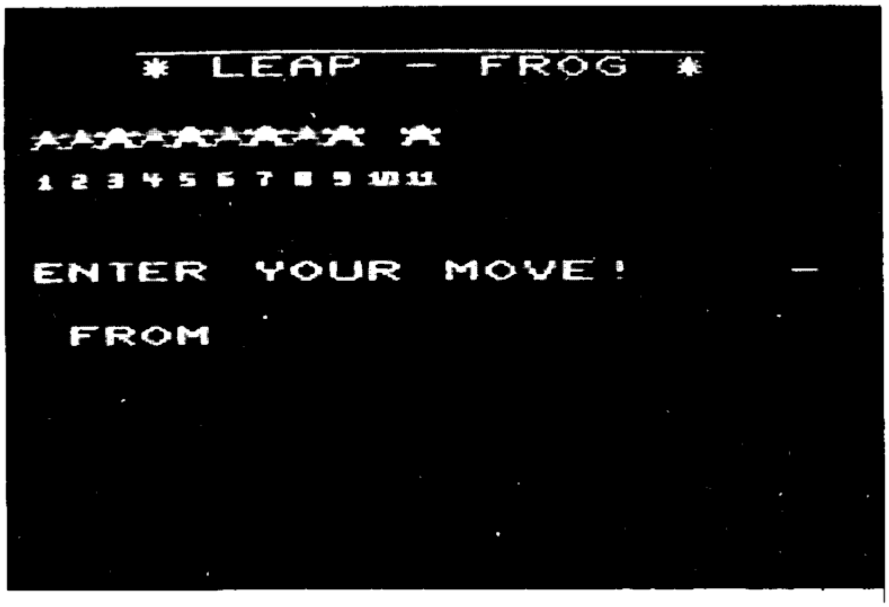
100-160 Sets up different strings for different frogs -white or green, jumping or stationary.

210-230 Sets up the array for where the frog is and which colour it is.

260-320 Prints actual display on the screen.

330 GOSUB to routine for inputting your move.

350-380	Checks for legality of your move.
400-510	Motivates the frog.
520-560	Reviews the line up at the end of the game.
570-900	Prints informative messages.
910-950	End of program.
960-1050	Inputs moves using routine at line 1390.
1060-1380	Prints messages of illegal moves and reprints display.
1390-1480	Actual input from Keyboard.
1490-1550	Sets up graphics characters.
1560-1700	Data for characters.



```
    * LEAP - FROG *

 1 2 3 4 5 6 7 8 9 10 11

ENTER YOUR MOVE !          -
 FROM
```

```
10 REM LEAP FROG
20 REM ****************************************
30 REM
34 REM
35 REM LIMIT TOP OF MEMORY
36 REM
40 POKE 56,PEEK(56)-2:POKE 52,PEEK(56)-2
50 POKE 51,PEEK(55):CLR
60 PRINT"□":POKE 36879,42
70 GOSUB 1490
80 POKE 36869,255
90 DIMA(19)
94 REM
95 REM SET UP STRINGS
96 REM
100 F$(0)="  "
110 F$(1)="▨D"
120 F$(2)="▤D"
130 AN$(1)="▨?"
140 AN$(2)="▤?"
150 AN$(3)=F$(0)
160 PP$="▚▨▨▨▨"
170 REM
180 T1$="□▨▨      _________________▆"
190 T1$=T1$+CHR$(13)+"    "
200 T1$=T1$+"▚* LEAP - FROG *▆"
204 REM
205 REM PUT FROGS INTO AN ARRAY
206 REM
210 FORK=1TO5
220 A(K)=1:A(K+6)=2
230 NEXT:A(6)=0
240 C=0
250 PRINTT1$:PRINT:PRINT
254 REM
255 REM PRINT FROGS ON THE SCREEN
256 REM
260 FORK=1TO11
270 PRINTF$(A(K));
280 NEXT:PRINT"▨▨"
290 PRINT
300 PRINT"T▨123456789:;"
310 PRINTPP$"▨▨▨▨▨"
320 PRINT"▤▨ENTER YOUR MOVE!     - ▆";
330 GOSUB960:FF=0
340 PRINT"▨":TD=1500
350 IFA(S)=0THEN1060
360 IFABS(S-E)>2THEN1090
370 IFA(E)THEN1110
380 IFE>11THENFF=1:E=12
390 X=0
```

```
394 REM
395 REM JUMP FROG
396 REM
400 PRINTPP$;TAB(S-1);:
410 IF A(S)=1 THEN PRINTAN$(1);:GOTO430
420 PRINTAN$(2);
430 FORL=1TO500:NEXT
440 PRINT"▮"AN$(3);
450 FORL=1TO50:NEXT
460 PRINTPP$;TAB(E-1);AN$(A(S));
470 FOR L=1 TO 300:NEXT L
480 FORK=1TO150:NEXT
490 PRINT"▮"F$(A(S));
500 C=C+1
510 A(E)=A(S):A(S)=0
520 FORI=1TO6
530 X=X+A(I)*10↑I
540 NEXT
550 IFFFTHEN1220
560 IFINT(X)<>222220THEN310
564 REM
565 REM FINISHED
566 REM
570 POKE36869,240:PRINT"▩▩▩▩▩▩":IFC<60THEN620
580 POKE36869,240
590 PRINT"▓⊐▤________________________"
600 PRINT"⁊▓YOU FINISHED, AT       LAST!!!"
610 GOTO650
620 IF C<50 THEN690
630 PRINT"▓⊐▤________________________"
640 PRINT"⁊▓NOT A BAD RESULT!!!"
650 PRINT"▩ YOU NEEDED"C"MOVES"
660 PRINT"▩ TO SOLVE THE PROBLEM"
670 PRINT"YOU REALLY SHOULD DO  BETTER!!"
680 GOTO880
690 IFC<40THEN760
700 PRINT"▓⊐▤________________________"
710 PRINT"▓VERY WELL DONE!!!"
720 PRINT"▩YOU SUCCEEDED TO"
730 PRINT"COMPLETE THE GAME IN"
740 PRINT"ONLY"C"MOVES - AN"
750 PRINT"ABOVE AVERAGE RESULT!":GOTO880
760 IFC=35THEN830
770 PRINT"▓⊐▤________________________"
780 PRINT"⁊▓EXCELLENT!! YOU ARE A REAL EXPERT!!!"
790 PRINT"▩YOU HAVE DONE IT IN"
800 PRINT"ONLY"C"MOVES, THIS"
810 PRINT"IS ALMOST THE BEST POSSIBLE RESULT."
820 GOTO880
830 PRINT"▓⊐▤________________________"
840 PRINT"⁊▓CONGRATULATIONS!"
```

```
850 PRINT"NOBODY CAN DO IT       BETTER"
860 PRINT"■YOU COMPLETED THE GAME IN 35 STEPS."
870 PRINT"THIS IS THE ABSOLUTE  MINIMUM."
880 PRINT"■■WOULD YOU LIKE TO PLAY AGAIN ? (Y/N)"
890 GETA$:IFA$<>"Y"ANDA$<>"N"THEN890
900 IFA$="Y"THENPOKE36869,255:GOTO210
904 REM
905 REM END
906 REM
910 POKE 56,PEEK(56)+2
920 POKE 36869,240
930 POKE 36879,27
940 PRINT"■■THANKS FOR PLAYING     'LEAP-FROG' -"
950 END
954 REM
955 REM CALCULATE MOVE FROM ROUTINE AT 1390
956 REM
960 POKE198,0
970 DEL=0:PRINT"■"
980 PRINTTAB(23);"■FROM                    ■■";
990 PRINTTAB(28);"■ ■■";
1000 GOSUB1390:IF DEL THEN PRINT"■";:GOTO 970
1010 S=VAL(C$):PRINT"■ TO  ■■";
1020 FOR I=1 TO 11:IF A(I)=0 THEN C$=STR$(I)
1030 NEXT
1040 PRINT"■■"C$;
1050 E=VAL(C$):RETURN
1054 REM
1055 REM ILLEGAL MOVE
1056 REM
1060 PRINT"■■■THERE IS NOBODY AT"
1070 PRINT"■POSITION #"S"■.■"
1080 PRINT"■PLEASE TRY AGAIN!":GOTO1190
1090 PRINT"■■■HEY, I CANNOT JUMP"
1100 PRINT"■THAT FAR!!":GOTO1190
1110 PRINT"■■■HEY, WHAT DO YOU THINK";
1120 PRINT"■YOU ARE DOING??"
1130 PRINT"■■THAT IS MY BEST"
1140 PRINT"■FRIEND, SITTING ON"
1150 PRINT"■■SEAT NR. "E"■!! YOU"
1160 PRINT"■CAN'T EXPECT ME TO"
1170 PRINT"■■LAND ON HIM!?! PLEASE"
1180 PRINT"■TRY AGAIN":TD=6000:GOTO1200
1190 TD=3000
1200 FORK=1TOTD:NEXT
1210 TD=3000:GOTO250
1220 PRINTPP$;TAB(11);F$(0);"■";
1230 PRINTF$(A(E));
1240 A$="%#*%!!■■■■■■■"
1250 PRINT"■■■■";TAB(12);
1260 FORK=1TO20
```

```
1270 PRINT"▣"A$;:A=SIN(3)
1280 PRINT"▀"A$;:A=SIN(3)
1290 NEXT:PRINT"          ▩▩▩▩▩"
1300 PRINT"________________________________"
1310 PRINT"▣NOW LOOK WHAT YOU HAVE DONE!!"
1320 PRINTPP$;"⌝";TAB(34);AN$(0);
1330 FORK=72TO0STEP-4
1340 PRINTF$(A(E));"▌";AN$(0);
1350 FORL=1TOK:NEXT
1360 PRINTF$(0)+"▌▩";
1370 NEXT:A(12)=0
1380 PRINTF$(0);"⌝TTTTT⌝":GOTO880
1384 REM
1385 REM INPUT MOVE FROM KEYBOARD
1386 REM
1390 GETC$:IFC$=""THEN1390
1400 IFC$=CHR$(20)THENDEL=1:RETURN
1410 IFC$<"1"ORC$>"9"THEN1390
1420 PRINT"▣"C$;:IFC$<>"1"THENRETURN
1430 IFA(8)*A(9)*A(10)*A(11)THENRETURN
1440 PRINT"▣ ▆▌";
1450 GETCC$:IFCC$=""THEN1450
1460 IFCC$=CHR$(20)THENDEL=1:RETURN
1470 IFCC$<"0"ORCC$>"1"THEN1450
1480 C$=C$+CC$:PRINT"▣"CC$;:RETURN
1484 REM
1485 REM SET UP GRAPHIC CHARACTERS
1486 REM
1490 FOR I=1 TO 15
1500 READ  X
1510 FOR J=0 TO 7
1520 READ K
1530 POKE X+J,K
1540 NEXT J,I
1550 RETURN
1554 REM
1555 REM CHARACTER DATA
1556 REM
1560 DATA 7424,0,0,0,0,0,0,0,0
1570 DATA 7560,16,48,16,16,56,0,0,0
1580 DATA 7568,56,8,56,32,56,0,0,0
1590 DATA 7576,56,8,56,8,56,0,0,0
1600 DATA 7584,40,40,60,8,8,0,0,0
1610 DATA 7592,56,32,56,8,56,0,0,0
1620 DATA 7600,56,32,56,40,56,0,0,0
1630 DATA 7608,56,8,16,16,16,0,0,0
1640 DATA 7616,56,40,56,40,56,0,0,0
1650 DATA 7624,56,40,56,8,56,0,0,0
1660 DATA 7632,46,106,42,42,126,0,0,0
1670 DATA 7640,36,108,36,36,126,0,0,0
1680 DATA 7664,24,189,255,60,126,66,195,0
```

```
1690 DATA 7672,153,189,126,60,126,66.66,195
1700 DATA 7552,56,40,40,40,56,0,0,0
READY.
```

RUBIK'S CUBE

DESCRIPTION

A two dimensional version of the popular cube designed by Messer Rubik. The game demands a high degree of concentration to solve the puzzle. All six faces of the cube are shown and one has to remember where and which side joins what.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

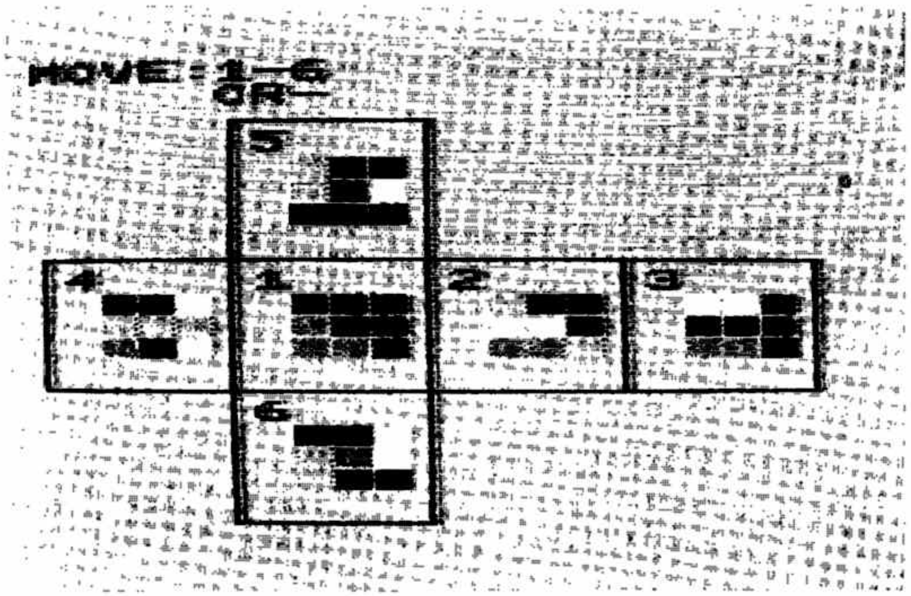
The primary instructions involved in the running of the game are to rotate a face of the (flat) cube. An anti-clock wise rotation is achieved by entering the number of the face to be moved. To rotate a face in a clockwise rotation a minus sign must be typed before the number.
Once you have completed the cube, or have become stuck, hit the space bar, this will result in the end of the game. A choice of a different number of twists is given at the beginning of the next game to facilitate ease or difficulty in completing the next game.

PROGRAM STRUCTURE

Below is a listing of the more interesting lines that make up the program for this method of playing the Rubik's Cube.

40-50 Limits the top of memory for graphics characters.
70-80 Inputs the initial number of twists to mix up the cube.
100-230 Displays the cube on the screen.
240-290 Array F holds the center poke location for each face.
300-310 Holds the character array for each face.
320-420 POKE locations for the rest of the pieces in each face.
520 GOSUB to create characters.

530-570 POKE for character face and colour.
610-810 Rotates the faces.
890-1010 Inputs moves and decisions.
1020-1070 Inputs the cycle for mixing the cube rather than have this as a random factor.
1080-1150 Sets up graphics characters.
1160-1220 Character data.



```
10 REM RUBIK'S CUBE
20 REM ************************************
30 REM
34 REM
35 REM LIMIT TOP OF MEMORY
36 REM
40 POKE 56,PEEK(56)-2:POKE 52,PEEK(56)-2
50 POKE 51,PEEK(55):CLR
60 DIMG(5,12):ZZ=30720:POKE36879,127
70 PRINT"■NUMBER OF TWISTS TO MIX"
80 INPUT"■ ■■";P
90 M=1
94 REM
95 REM DISPLAY CUBE
96 REM
100 PRINT"■"
110 DIMF(5,8)
120 PRINT"■■■   ┌────┐"
130 PRINT"■     |5   |"
140 FORI=1TO4
150 PRINT"■     |    |":NEXT
160 PRINT"■┌────┼────┼────┬────┐"
170 PRINT"■|4   |1   |2   |3   |"
180 FORI=1TO4
190 PRINT"■|    |    |    |    |":NEXT
200 PRINT"■└────┼────┼────┴────┘"
210 PRINT"■     |6   |"
220 FORI=1TO4:PRINT"■      |    |":NEXT
230 PRINT"■     └────┘■"
234 REM
235 REM SCREEN LOCATIONS OF CENTRE OF EACH FACE
236 REM
240 F(0,0)=7680+8+(12*22)
250 F(1,0)=7680+13+(12*22)
260 F(2,0)=7680+18+(12*22)
270 F(3,0)=7680+3+(12*22)
280 F(4,0)=7680+8+(6*22)
290 F(5,0)=7680+8+(18*22)
294 REM
295 REM CHARACTER FOR EACH FACE
296 REM
300 C(0)=58:C(1)=59:C(2)=60:
310 C(3)=61:C(4)=62:C(5)=63
314 REM
315 REM RELATIVE POSITIONS OF PIECES TO CENTRE
316 REM
320 FORI=0TO5
330 K=F(I,0)
340 F(I,1)=K-23
350 F(I,2)=K-22
360 F(I,3)=K-21
```

```
370 F(I,4)=K+1
380 F(I,5)=K+23
390 F(I,6)=K+22
400 F(I,7)=K+21
410 F(I,8)=K-1
420 NEXT
430 FORI=0TO5
440 FORJ=0TO9STEP3
450 READN
460 FORK=1TO3
470 READN2
480 G(I,J+K)=F(N,N2)
490 NEXT
500 NEXT
510 NEXT
520 GOSUB 1080:POKE 36869,255
524 REM
525 REM DISPLAY ACTUAL PIECES OF CUBE
526 REM
530 FORJ=0TO5
540 FORI=0TO8
550 POKEF(J,I),C(J):POKEF(J,I)+ZZ,C(J)-58
560 NEXT
570 NEXT
580 IFP<=0THEN1010
590 GOTO880
599 REM
600 REM MOVE PIECES
601 REM
610 S1=PEEK(F(I,1))
620 S2=PEEK(F(I,2))
630 FORJ=1TO6
640 POKEF(I,J),PEEK(F(I,J+2)):NEXT
650 POKEF(I,7),S1
660 POKEF(I,8),S2
670 S1=PEEK(G(I,1))
680 S2=PEEK(G(I,2))
690 S3=PEEK(G(I,3))
700 FORJ=1TO9
710 POKEG(I,J),PEEK(G(I,J+3))
720 NEXT
730 POKEG(I,10),S1
740 POKEG(I,11),S2
750 POKEG(I,12),S3
760 FOR J=0 TO 5
770 FOR I=0 TO 8
780 POKE F(J,I)+ZZ,PEEK(F(J,I))-58
790 NEXT I
800 NEXT J
810 RETURN
820 DATA4,7,6,5,1,1,8,7,5,3,2,1,3,5,4,3
```

```
830 DATA4,5,4,3,2,1,8,7,5,5,4,3,0,5,4,3
840 DATA4,3,2,1,3,1,8,7,5,7,6,5,1,5;4,3
850 DATA4,1,8,7,0,1,8,7,5,1,8,7,2,5,4,3
860 DATA2,3,2,1,1,3,2,1,0,3,2,1,3,3,2,1
870 DATA0,7,6,5,1,7,6,5,2,7,6,5,3,7,6,5
880 FORX=1TOP:I=INT(RND(1)*6):GOSUB600:NEXT
884 REM
885 REM INPUT MOVE
886 REM
890 PRINT"▓▓▓MOVE:1-6":PRINT"▓      OR-"
900 IFM>2THENPOKE7680,173
910 IFM<2THENPOKE7680,160
920 POKE38400,0:GETA$
930 IFA$=""THEN920
940 IFA$="-"THENM=M+2:M=MAND3:GOTO890
950 IFA$<>" "THEN970:REM RESTART
960 POKE 56,PEEK(56)+2:POKE36869,240:RUN
970 A=VAL(A$):IFA=0THEN920
980 IFA>6THEN920
990 FORN=1TOM:I=A-1:GOSUB600:NEXT
1000 M=1:GOTO890
1010 IFP=0THEN890
1020 INPUT"▓GROUP CYCLE- STRING";G$
1030 INPUT"▓NUMBER OF CYCLES                ▓▓▓▓▓▓▓▓▓▓";NC
1040 IFNC<1THEN1020
1050 FORCY=1TONC:FORGG=1TOLEN(G$)
1060 I=VAL(MID$(G$,GG,1))-1:GOSUB600:NEXT:NEXT
1070 GOTO1030
1074 REM
1075 REM SET UP GRAPHICS CHARACTERS
1076 REM
1080 FOR I=1 TO 7
1090 READ X
1100 FOR J=0 TO 7
1110 READ K
1120 POKE X+J,K
1130 NEXT J
1140 NEXT I
1150 RETURN
1154 REM
1155 REM CHARACTER DATA
1156 REM
1160 DATA 7424,0,0,0,0,0,0,0,0
1170 DATA 7632,0,127,127,127,127,127,127,127
1180 DATA 7640,0,127,127,127,127,127,127,127
1190 DATA 7648,0,127,127,127,127,127,127,127
1200 DATA 7656,0,127,127,127,127,127,127,127
1210 DATA 7664,0,127,127,127,127,127,127,127
1220 DATA 7672,0,127,127,127,127,127,127,127
READY.
```

BOSS PUZZLE

DESCRIPTION

A test of mental agility. The aim of the game is to rearrange numbers from 0 to 15 and one space into the following order:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

To start, the numbers are randomly placed in the grid. By moving numbers into and out of spaces, they can be rearranged. While the game is being played, a clock ticks away in the top right corner of the screen and a counter counting the number of moves taken is displayed in the top left corner. A colourful display, and quite enjoyable to play.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

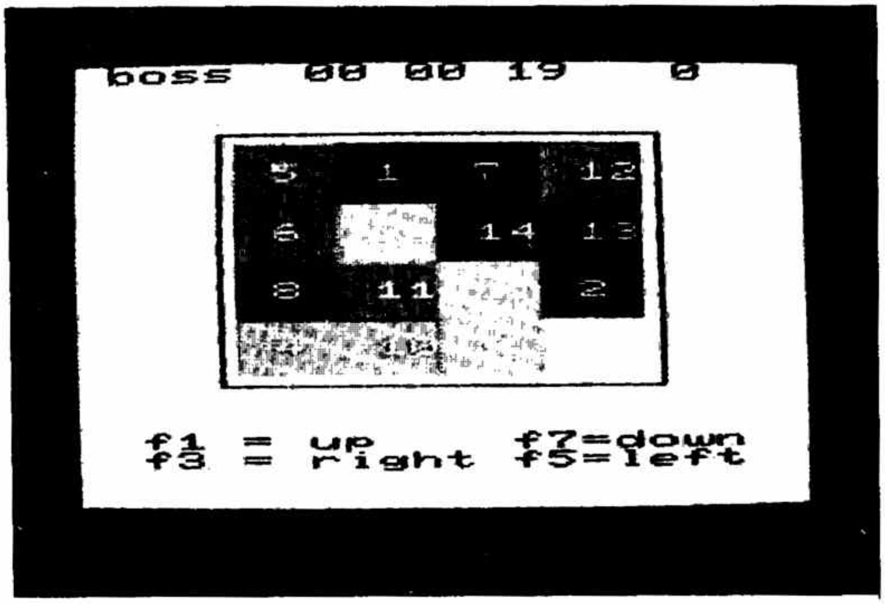
After typing run, the program displays one screenful of instructions. After hitting return, the grid with numbers in order will appear and then start to shuffle. When the shuffling is completed, the clock starts and the game begins. The directions are obtained by pressing the function keys. The directions corresponding to each function key are displayed constantly on the top of the screen.

PROGRAM STRUCTURE

The lines of interest in this game are as follows:

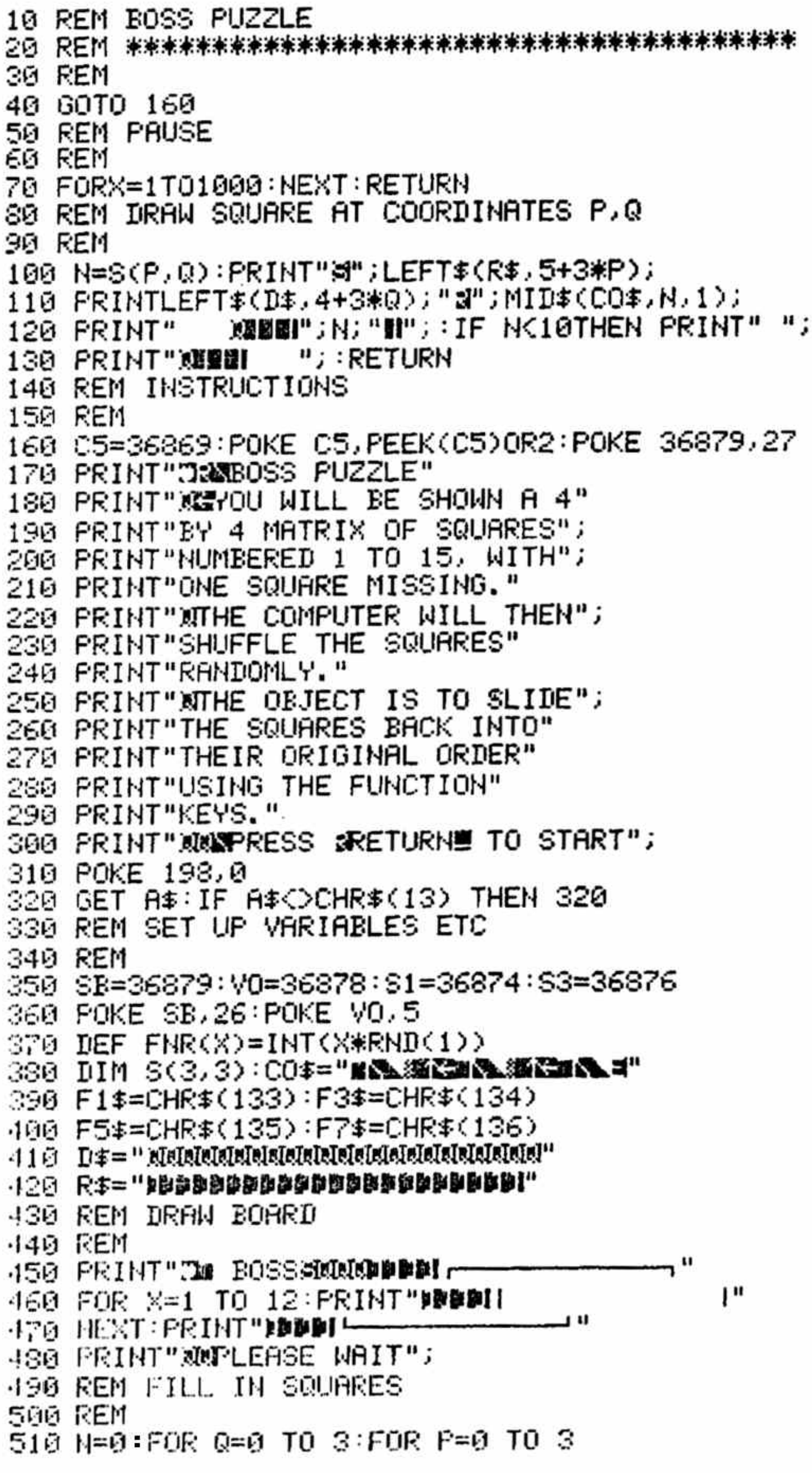
70	Subroutine to pause.
100-130	Subroutine to display square containing number at coordinates p,q.
160-320	Print instructions.
350-420	Set up variables required.
450-480	Draw board on screen.

510-530 Fill in squares.
560-620 Shuffle squares with random moves.
650-670 Start game.
700-720 Check to see if the squares are in order.
750-940 Accept and perform move input using function keys.
970-1070 Give nasty display.
1100-1190 Give result and input for another game.

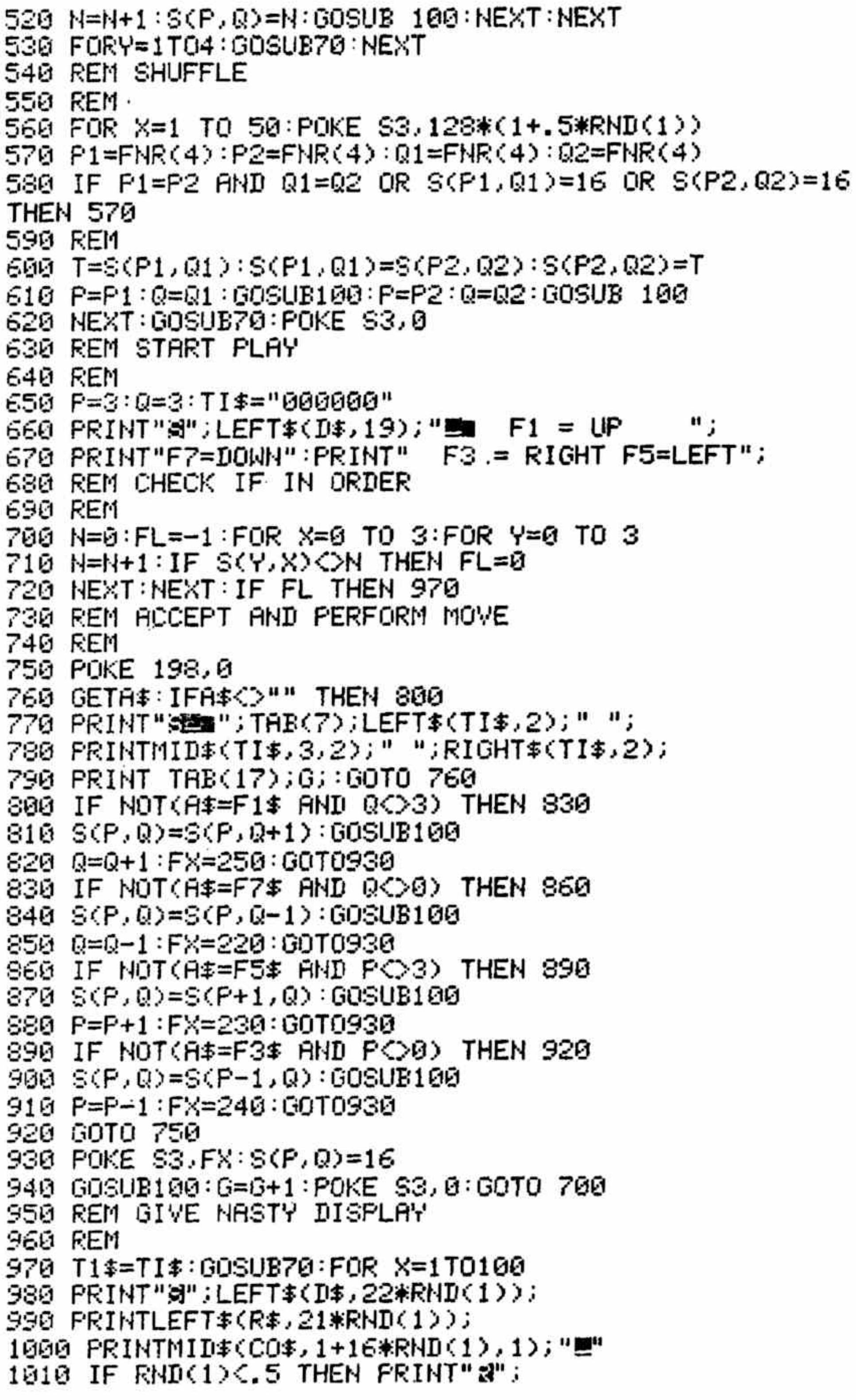


```
boss   00 00 19    0

f1 = up      f7=down
f3 = right   f5=left
```



```
10 REM BOSS PUZZLE
20 REM ****************************************
30 REM
40 GOTO 160
50 REM PAUSE
60 REM
70 FORX=1TO1000:NEXT:RETURN
80 REM DRAW SQUARE AT COORDINATES P,Q
90 REM
100 N=S(P,Q):PRINT"▮";LEFT$(R$,5+3*P);
110 PRINTLEFT$(D$,4+3*Q);"▮";MID$(CO$,N,1);
120 PRINT"   ▮▮▮▮";N;"▮";:IF N<10THEN PRINT" ";
130 PRINT"▮▮▮▮   ";:RETURN
140 REM INSTRUCTIONS
150 REM
160 C5=36869:POKE C5,PEEK(C5)OR2:POKE 36879,27
170 PRINT"▮▮▮BOSS PUZZLE"
180 PRINT"▮▮YOU WILL BE SHOWN A 4"
190 PRINT"BY 4 MATRIX OF SQUARES";
200 PRINT"NUMBERED 1 TO 15, WITH";
210 PRINT"ONE SQUARE MISSING."
220 PRINT"▮THE COMPUTER WILL THEN";
230 PRINT"SHUFFLE THE SQUARES"
240 PRINT"RANDOMLY."
250 PRINT"▮THE OBJECT IS TO SLIDE";
260 PRINT"THE SQUARES BACK INTO"
270 PRINT"THEIR ORIGINAL ORDER"
280 PRINT"USING THE FUNCTION"
290 PRINT"KEYS."
300 PRINT"▮▮▮PRESS ▮RETURN▮ TO START";
310 POKE 198,0
320 GET A$:IF A$<>CHR$(13) THEN 320
330 REM SET UP VARIABLES ETC
340 REM
350 SB=36879:V0=36878:S1=36874:S3=36876
360 POKE SB,26:POKE V0,5
370 DEF FNR(X)=INT(X*RND(1))
380 DIM S(3,3):CO$="▮▮▮▮▮▮▮▮▮▮▮"
390 F1$=CHR$(133):F3$=CHR$(134)
400 F5$=CHR$(135):F7$=CHR$(136)
410 D$="▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
420 R$="▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
430 REM DRAW BOARD
440 REM
450 PRINT"▮▮ BOSS▮▮▮▮▮▮▮▮┌────────────┐"
460 FOR X=1 TO 12:PRINT"▮▮▮▮│            │"
470 NEXT:PRINT"▮▮▮▮└────────────┘"
480 PRINT"▮▮PLEASE WAIT";
490 REM FILL IN SQUARES
500 REM
510 N=0:FOR Q=0 TO 3:FOR P=0 TO 3
```



```
520 N=N+1:S(P,Q)=N:GOSUB 100:NEXT:NEXT
530 FORY=1TO4:GOSUB70:NEXT
540 REM SHUFFLE
550 REM
560 FOR X=1 TO 50:POKE S3,128*(1+.5*RND(1))
570 P1=FNR(4):P2=FNR(4):Q1=FNR(4):Q2=FNR(4)
580 IF P1=P2 AND Q1=Q2 OR S(P1,Q1)=16 OR S(P2,Q2)=16
THEN 570
590 REM
600 T=S(P1,Q1):S(P1,Q1)=S(P2,Q2):S(P2,Q2)=T
610 P=P1:Q=Q1:GOSUB100:P=P2:Q=Q2:GOSUB 100
620 NEXT:GOSUB70:POKE S3,0
630 REM START PLAY
640 REM
650 P=3:Q=3:TI$="000000"
660 PRINT"♥";LEFT$(D$,19);"██  F1 = UP     ";
670 PRINT"F7=DOWN":PRINT"  F3 .= RIGHT F5=LEFT";
680 REM CHECK IF IN ORDER
690 REM
700 N=0:FL=-1:FOR X=0 TO 3:FOR Y=0 TO 3
710 N=N+1:IF S(Y,X)<>N THEN FL=0
720 NEXT:NEXT:IF FL THEN 970
730 REM ACCEPT AND PERFORM MOVE
740 REM
750 POKE 198,0
760 GETA$:IFA$<>"" THEN 800
770 PRINT"S██";TAB(7);LEFT$(TI$,2);" ";
780 PRINTMID$(TI$,3,2);" ";RIGHT$(TI$,2);
790 PRINT TAB(17);G;:GOTO 760
800 IF NOT(A$=F1$ AND Q<>3) THEN 830
810 S(P,Q)=S(P,Q+1):GOSUB100
820 Q=Q+1:FX=250:GOTO930
830 IF NOT(A$=F7$ AND Q<>0) THEN 860
840 S(P,Q)=S(P,Q-1):GOSUB100
850 Q=Q-1:FX=220:GOTO930
860 IF NOT(A$=F5$ AND P<>3) THEN 890
870 S(P,Q)=S(P+1,Q):GOSUB100
880 P=P+1:FX=230:GOTO930
890 IF NOT(A$=F3$ AND P<>0) THEN 920
900 S(P,Q)=S(P-1,Q):GOSUB100
910 P=P-1:FX=240:GOTO930
920 GOTO 750
930 POKE S3,FX:S(P,Q)=16
940 GOSUB100:G=G+1:POKE S3,0:GOTO 700
950 REM GIVE NASTY DISPLAY
960 REM
970 T1$=TI$:GOSUB70:FOR X=1TO100
980 PRINT"♥";LEFT$(D$,22*RND(1));
990 PRINTLEFT$(R$,21*RND(1));
1000 PRINTMID$(CO$,1+16*RND(1),1);"█"
1010 IF RND(1)<.5 THEN PRINT"♥";
```

```
1020 PRINT"DONE IT!";
1030 POKE SB,1+255*RND(1):POKE VO,1+X/10
1040 FOR Y=0TO2:POKE S1+Y,128*(1+RND(1))
1050 NEXT:NEXT:POKE VO,15:POKE S1+3,220
1060 GOSUB70:GOSUB70:POKE SB,27
1070 FORY=0TO4:POKE S1+Y,0:NEXT
1080 REM GIVE RESULT
1090 REM
1100 PRINT"▩▩IT TOOK YOU";G;"MOVES"
1110 PRINT"AND ";:A$=LEFT$(T1$,2)
1120 IF A$<>"00" THEN PRINT A$;" HOURS"
1130 PRINT MID$(T1$,3,2);" MINUTES"
1140 PRINT RIGHT$(T1$,2);" SECONDS."
1150 PRINT"▩▩♠NOTHER GAME? ";:POKE 198,0
1160 GET A$:IF A$<>"Y" AND A$<>"N" THEN 1160
1170 IF A$="N" THEN 1190
1180 PRINT "YES":GOSUB 70:CLR:GOTO 350
1190 PRINT"NO":PRINT"▩THANKS FOR PLAYING.":END
READY.
```

SKETCHING 1

DESCRIPTION

A program which allows the user to put Commodore graphics symbols anywhere on the screen.

EQUIPMENT REQUIRED

Basic VIC-20.

RUNNING THE PROGRAM

When loaded and running the program will display a selection of characters at the top of the screen accompanied by a cursor. By manipulating the joystick and depressing the 'fire button' over the users choice of character, the user may then position the character anywhere on the screen he chooses by, again, depressing the 'fire button'.
To change the colour of the screen and the border the user may depress the 'C' key and, entering the correct co-ordinates, may alter the screen to his satisfaction.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows.

70-80 Sets up control registers.

120-140 Puts characters up on the first two lines of the screen and reverses the first character to signify the cursor.

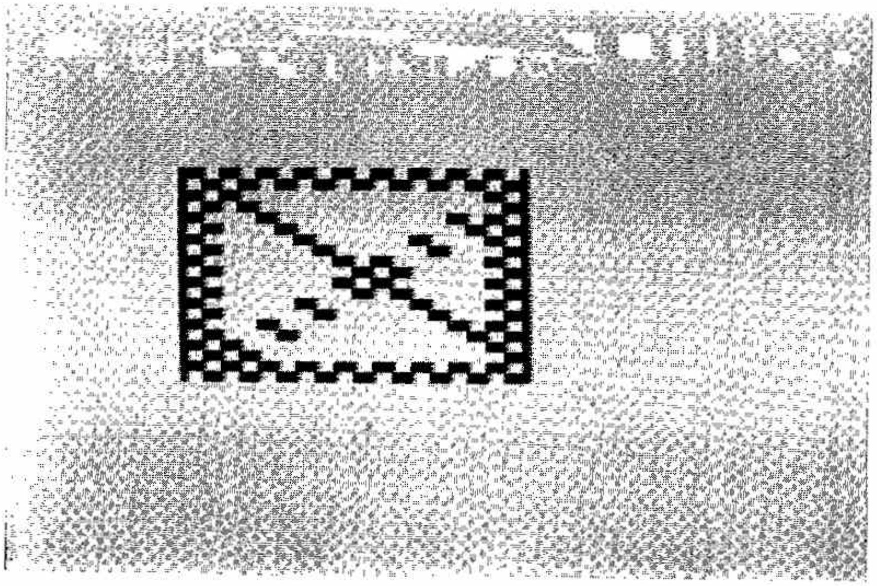
200-290 Checks the joystick.

350 When 'fire button' is depressed and situated on the top two lines, the joystick will pick up the character.

360-320 If the 'fire button' is depressed while the cursor is in the rest of the screen the joystick will drop the character and/or stop dropping the character.

380-420 If a movement is input checks whether it is in the screen and then moves the cursor.

450-500 If 'C' is depressed, input 2 values to change the screen and border colours. 0-9.



```
10 REM SKETCHING 1
20 REM ****************************************
30 REM
40 REM
50 REM SET ALL SOUND REGISTERS TO ZERO
60 REM
70 FORJ=36874TO36876:POKEJ,0:NEXT:POKE36878,15
80 POKE36879,8:PRINT"▚▜";:V=36876:POKE36878,15:R=0
90 REM
100 REM PRINT GRAPHICS CHARACTERS ON SCREEN
110 REM
120 PRINT"▒⁂H┴┬▔▔▀▄▄▁◥▐▌|||▪▗▖┌┘▀▚|┼|◤●○▔";"
130 S=7680:C=38400:W=0
140 X=PEEK(S):POKES,X+128:POKEV,200
150 FORJ=0TO25:NEXTJ
160 REM
170 REM LOOK FOR INSTRUCTION FROM
180 REM KEYBOARD OR JOYSTICK
190 REM
200 POKE37154,127:J0=-((PEEK(37152)AND128)=0)
210 POKE37154,255:P=PEEK(37151)
220 GETA$:IFA$="C"THENGOSUB480
230 A=VAL(A$)-1:IFA<>-1THENR=A
240 J1=-((PAND8)=0)
250 IFWANDS>7743THENX=CR
260 J2=-((PAND16)=0)
270 J3=-((PAND4)=0)
280 FB=-((PAND32)=0)
290 POKES,X:POKEV,0:IFW=1ANDS>6721THENPOKEC,R
300 REM
310 REM CHECK MOVEMENT IS WITHIN TOP TWO
320 REM LINES AND PICK UP CHARACTER IF
330 REM FIRE BUTTON IS PRESSED
340 REM
350 IFFBANDS<7743THENCR=X:W=0:GOTO140
360 IFFBANDS>7742ANDM=1THENW=W*-1+1:M=0
370 IFFB=0THENM=1
380 IFJ0ANDS<8185THENS=S+1:C=C+1
390 IFJ1ANDS<8164THENS=S+22:C=C+22
400 IFJ2ANDS>7680THENS=S-1:C=C-1
410 IFJ3ANDS>7701THENS=S-22:C=C-22
420 GOTO140
430 REM
440 REM CHANGE SCREEN AND BORDER COLOURS
450 REM BY INPUTING TWO VALUES AFTER
460 REM PRESSING 'C'
470 REM
480 POKE36876,0:FORJ=1TO100:NEXT
490 GETA$:IFA$=""THEN490
500 A=VAL(A$):A=A-1
510 POKE36879,(PEEK(36879)AND15)+A*16
```

```
520 FORJ=1TO100:NEXT:A$=""
530 GETA$:IFA$=""THEN530
540 A=VAL(A$):A=A-1
550 POKE36879,(PEEK(36879)AND240)+A+8
560 A$="":RETURN
READY.
```

SKETCHING 2

DESCRIPTION

This program allows the user to draw High Resolution characters in a 64 × 64 pixel block, in the centre of the screen with the use of the joystick.

EQUIPMENT REQUIRED

Basic VIC-20 + joystick.

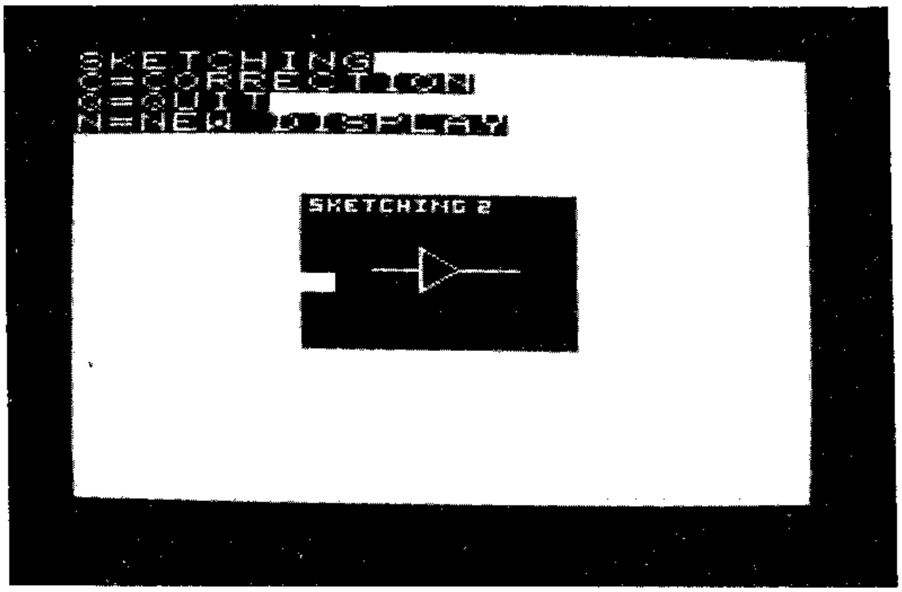
RUNNING THE PROGRAM

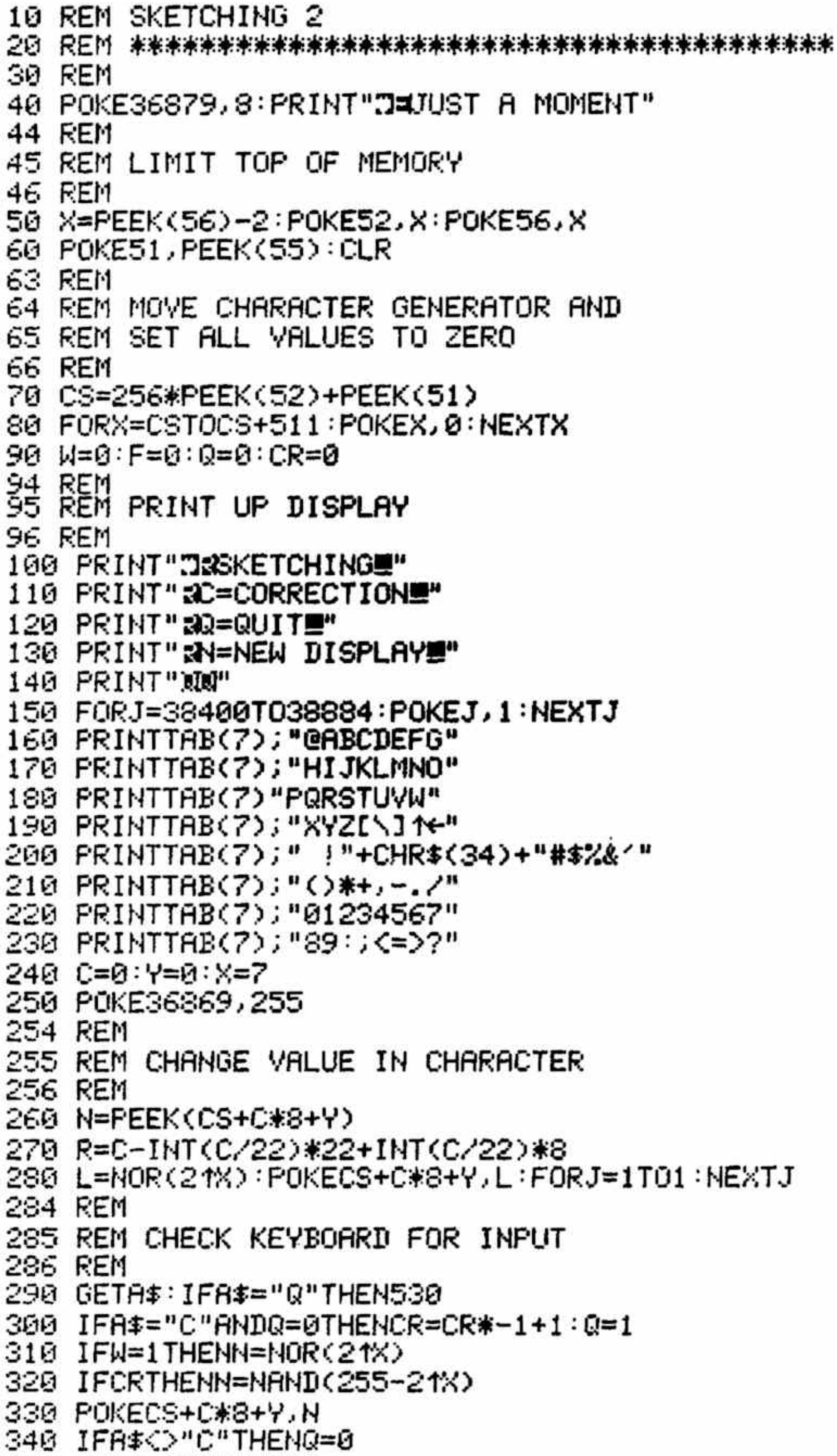
Following the same principles as Sketching 1 but does not pick up characters.
N Sets the block to original.
C When the cursor is moved in this mode any points it moves over will be cleared. C then returns to original.
Q To quit.

PROGRAM STRUCTURE

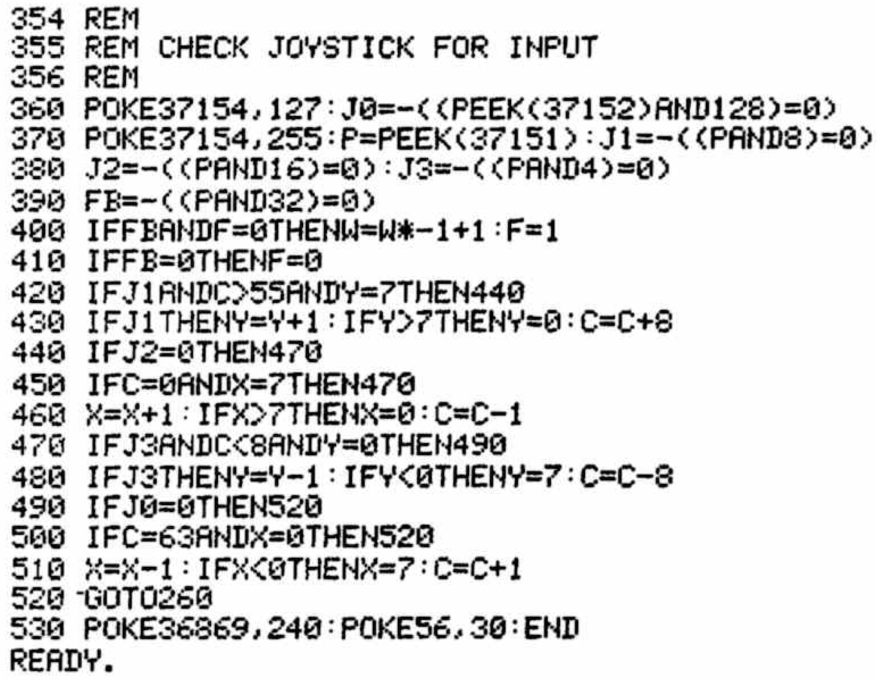
The following lines are of the most interest in this program.

40-90	Sets up control registers, limits memory, moves character generator and sets all values to 0.
100-250	Display.
260-280	Changes character.
290-350	Check for keyboard input.
360-520	Check for joystick input.
530	End.


SKETCHING
C=CORRECTION
Q=QUIT
N=NEW DISPLAY
SKETCHING 2



```
10 REM SKETCHING 2
20 REM ****************************************
30 REM
40 POKE36879,8:PRINT"■■JUST A MOMENT"
44 REM
45 REM LIMIT TOP OF MEMORY
46 REM
50 X=PEEK(56)-2:POKE52,X:POKE56,X
60 POKE51,PEEK(55):CLR
63 REM
64 REM MOVE CHARACTER GENERATOR AND
65 REM SET ALL VALUES TO ZERO
66 REM
70 CS=256*PEEK(52)+PEEK(51)
80 FORX=CSTOCS+511:POKEX,0:NEXTX
90 W=0:F=0:Q=0:CR=0
94 REM
95 REM PRINT UP DISPLAY
96 REM
100 PRINT"■■SKETCHING■"
110 PRINT"■C=CORRECTION■"
120 PRINT"■Q=QUIT■"
130 PRINT"■N=NEW DISPLAY■"
140 PRINT"■■"
150 FORJ=38400TO38884:POKEJ,1:NEXTJ
160 PRINTTAB(7);"@ABCDEFG"
170 PRINTTAB(7);"HIJKLMNO"
180 PRINTTAB(7)"PQRSTUVW"
190 PRINTTAB(7);"XYZ[\]↑←"
200 PRINTTAB(7);" !"+CHR$(34)+"#$%&'"
210 PRINTTAB(7);"()*+,-./"
220 PRINTTAB(7);"01234567"
230 PRINTTAB(7);"89:;<=>?"
240 C=0:Y=0:X=7
250 POKE36869,255
254 REM
255 REM CHANGE VALUE IN CHARACTER
256 REM
260 N=PEEK(CS+C*8+Y)
270 R=C-INT(C/22)*22+INT(C/22)*8
280 L=NOR(2↑X):POKECS+C*8+Y,L:FORJ=1TO1:NEXTJ
284 REM
285 REM CHECK KEYBOARD FOR INPUT
286 REM
290 GETA$:IFA$="Q"THEN530
300 IFA$="C"ANDQ=0THENCR=CR*-1+1:Q=1
310 IFW=1THENN=NOR(2↑X)
320 IFCRTHENN=NAND(255-2↑X)
330 POKECS+C*8+Y,N
340 IFA$<>"C"THENQ=0
350 IFA$="N"THEN80
```



```
354 REM
355 REM CHECK JOYSTICK FOR INPUT
356 REM
360 POKE37154,127:J0=-((PEEK(37152)AND128)=0)
370 POKE37154,255:P=PEEK(37151):J1=-((PAND8)=0)
380 J2=-((PAND16)=0):J3=-((PAND4)=0)
390 FB=-((PAND32)=0)
400 IFFBANDF=0THENW=W*-1+1:F=1
410 IFFB=0THENF=0
420 IFJ1ANDC>55ANDY=7THEN440
430 IFJ1THENY=Y+1:IFY>7THENY=0:C=C+8
440 IFJ2=0THEN470
450 IFC=0ANDX=7THEN470
460 X=X+1:IFX>7THENX=0:C=C-1
470 IFJ3ANDC<8ANDY=0THEN490
480 IFJ3THENY=Y-1:IFY<0THENY=7:C=C-8
490 IFJ0=0THEN520
500 IFC=63ANDX=0THEN520
510 X=X-1:IFX<0THENX=7:C=C+1
520 GOTO260
530 POKE36869,240:POKE56,30:END
READY.
```

KALEIDOSCOPE

DESCRIPTION

This program displays a constantly changing colourful pattern on the screen, reminiscent of the childrens' toy Kaleidoscope.

EQUIPMENT REQUIRED

Basic VIC-20.

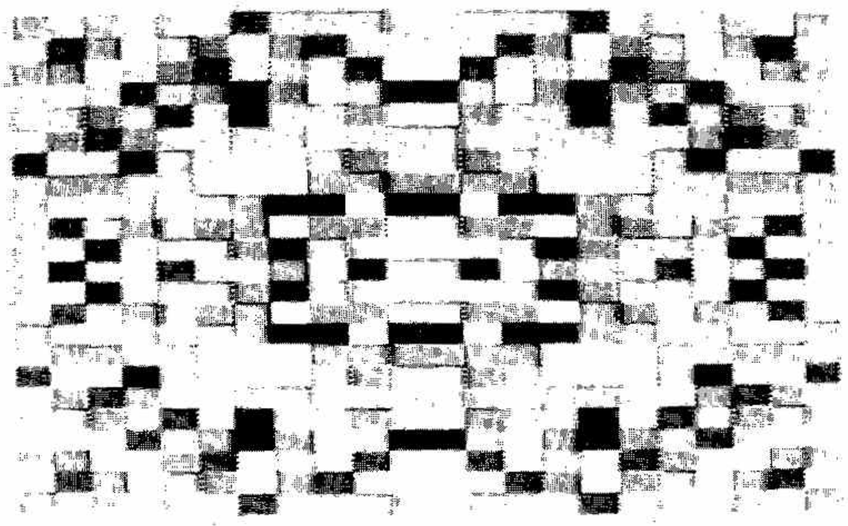
RUNNING THE PROGRAM

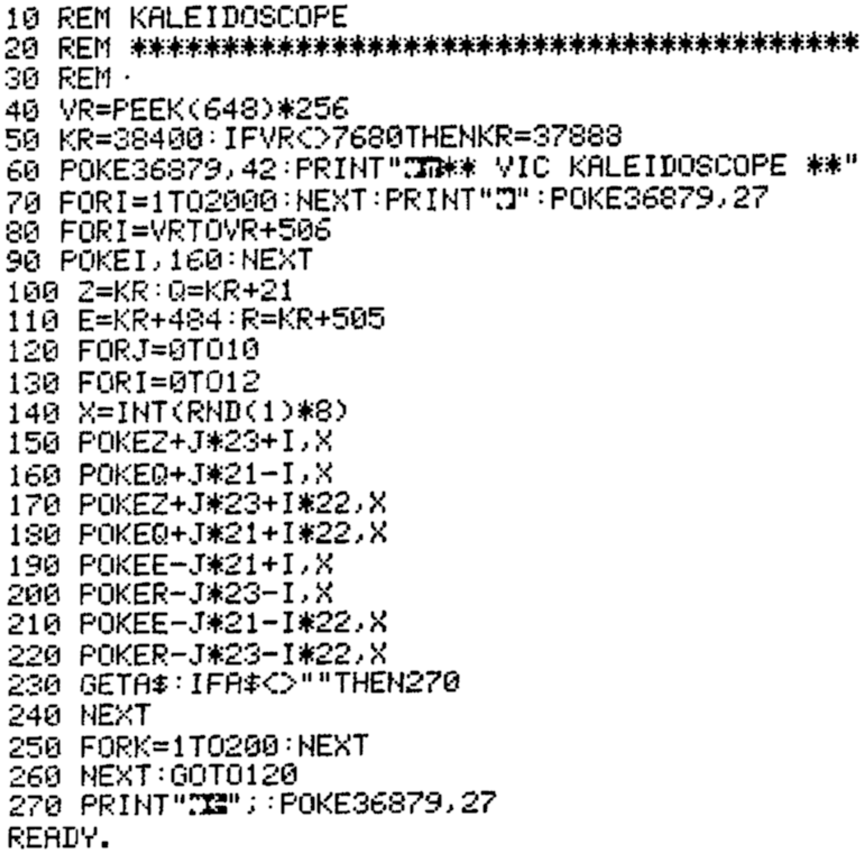
This program requires no expansion.
After loading the program type run and watch the pattern grow. To stop the program hit any key to end.

PROGRAM STRUCTURE

There are a few lines of interest in this program. These lines are as follows.

40-50	Choose correct screen and colour map starts.
80-260	Displays Kaleidoscope pattern.





```
10 REM KALEIDOSCOPE
20 REM ****************************************
30 REM ·
40 VR=PEEK(648)*256
50 KR=38400:IFVR<>7680THENKR=37888
60 POKE36879,42:PRINT"♥R** VIC KALEIDOSCOPE **"
70 FORI=1TO2000:NEXT:PRINT"♥":POKE36879,27
80 FORI=VRTOVR+506
90 POKEI,160:NEXT
100 Z=KR:Q=KR+21
110 E=KR+484:R=KR+505
120 FORJ=0TO10
130 FORI=0TO12
140 X=INT(RND(1)*8)
150 POKEZ+J*23+I,X
160 POKEQ+J*21-I,X
170 POKEZ+J*23+I*22,X
180 POKEQ+J*21+I*22,X
190 POKEE-J*21+I,X
200 POKER-J*23-I,X
210 POKEE-J*21-I*22,X
220 POKER-J*23-I*22,X
230 GETA$:IFA$<>""THEN270
240 NEXT
250 FORK=1TO200:NEXT
260 NEXT:GOTO120
270 PRINT"♥S";:POKE36879,27
READY.
```

HI-RES DEMO

DESCRIPTION

Displays an attractive high resolution display on the screen. No values are input.

EQUIPMENT REQUIRED

Basic VIC-20 plus super expander cartridge.

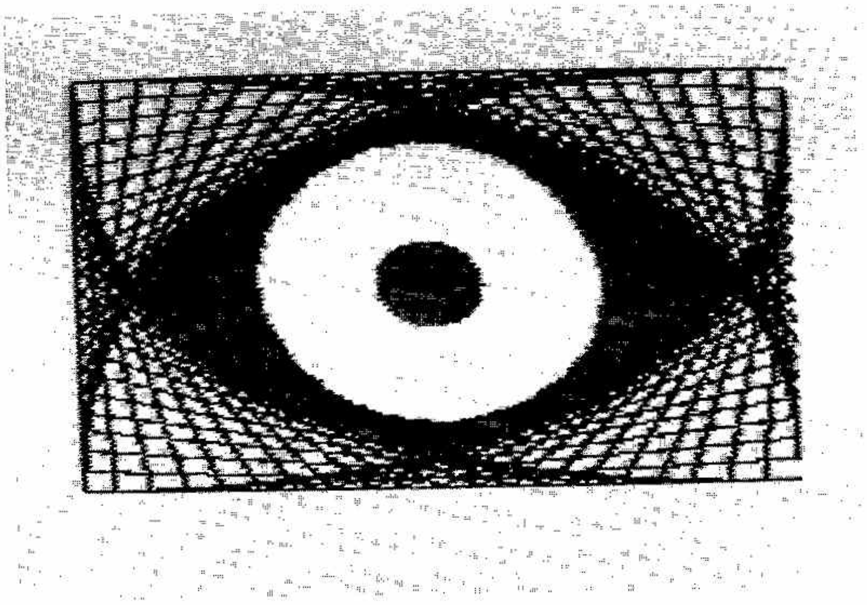
RUNNING THE PROGRAM

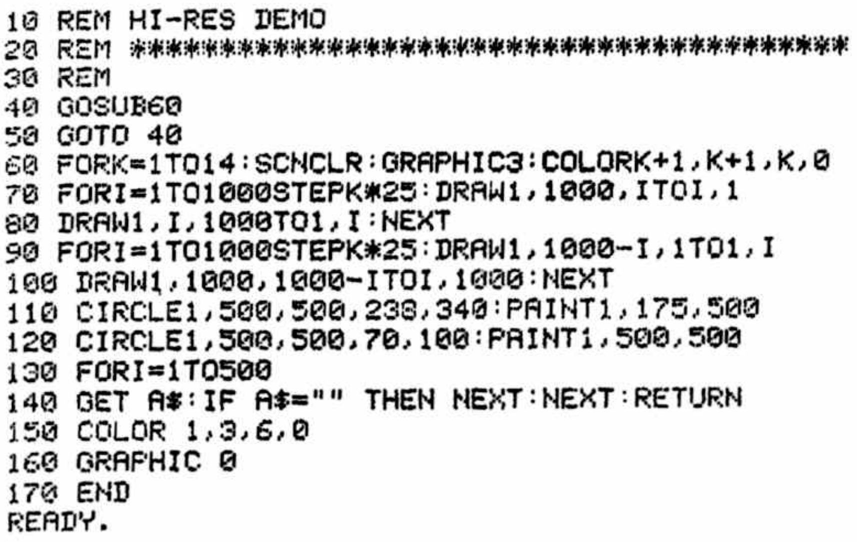
When loaded type RUN and watch the pattern appear on the screen. To stop hit any key.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows:

40-50 Main program loop.
60 Clears screen and sets graphic mode in colour.
70-100 Draws lines on the screen.
110 Draws large circle and colours in around the circle.
120 Draws small circle and colours inside the circle.
130-140 Pause and check for key press.
150-170 If key press, the end.





```
10 REM HI-RES DEMO
20 REM ****************************************
30 REM
40 GOSUB60
50 GOTO 40
60 FORK=1TO14:SCNCLR:GRAPHIC3:COLORK+1,K+1,K,0
70 FORI=1TO1000STEPK*25:DRAW1,1000,ITOI,1
80 DRAW1,I,1000TO1,I:NEXT
90 FORI=1TO1000STEPK*25:DRAW1,1000-I,1TO1,I
100 DRAW1,1000,1000-ITOI,1000:NEXT
110 CIRCLE1,500,500,238,340:PAINT1,175,500
120 CIRCLE1,500,500,70,100:PAINT1,500,500
130 FORI=1TO500
140 GET A$:IF A$="" THEN NEXT:NEXT:RETURN
150 COLOR 1,3,6,0
160 GRAPHIC 0
170 END
READY.
```

BANDIT

DESCRIPTION

This game is a graphical representation of a one armed bandit. The machine has five different symbols all in different colours. One of the symbols will pay out if it is just the first, the first and second, or the first, second, and third. All other symbols pay out only if all three reels show the same. The game allows the user to bet up to 12 units, and the win is multiplied by how many units bet. e.g. if 12 units are bet and three bar symbols appear, the computer will pay out 1200 units.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

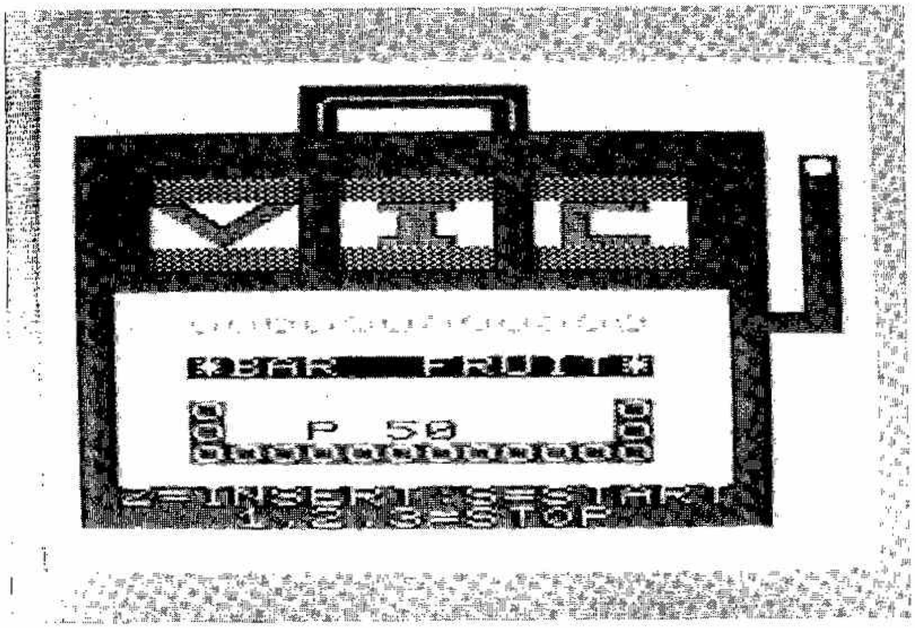
RUNNING THE PROGRAM

After typing run, the display will show a one armed bandit. Use the key 'Z' to input the money, 'S' to start the reels spinning, and the keys '1, 2, 3' to stop the respective reels. The keys to stop the reels do not act immediately but allow the reels to spin for a couple more symbols so that the user cannot cheat. The game is over when all the money has run out.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

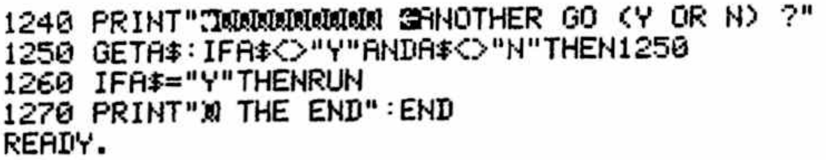
50-260	Display Bandit.
270-360	Input cash or start.
370-560	The array A$ holds the actual symbols.
620-900	Spin reels.
920-950	Input keys to stop reels.
960-1040	When all reels have stopped, check for a win.
1050-1140	Win,display payout.
1150-1210	Choose the next symbol to appear.
1220-1270	Run out of money.


BAR FRUIT
P 50
Z=INSERT S=START
1,2,3=STOP

```
·10 REM BANDIT 1
 20 REM ***************************************
 30 REM
 34 REM
 35 REM DISPLAY FRUIT MACHINE
 36 REM
 40 Q=5:SC= 50:GOSUB1150
 50 PRINT"▚▚"SPC(7)"▚▚╭──────╮▀
 60 PRINTSPC(7)"▚|▀      ▚|▀▚
 70 A$=" ▚                      ▀
 80 B$=" ▚ ▀▚▒▒▒▒▒▒▚▚ ▀▚▒▒▒▒▒▒▚▚ ▀▚▒▒▒▒▒▒▚▚  ▀ ▚ ▀
 90 C$=" ▚ ▀                      ▚ ▀
100 PRINTA$:PRINTA$" ▚▚":PRINTB$
110 PRINTA$" ▚ ▀":PRINTA$" ▚ ▀"
120 PRINT"▚▚ ▚▚▚▚▼▚▚| ▚ ▀ ▚|▚▀▀ "
130 PRINT" ▚▚ ▼ ▚ ▚ ▀ ▚|▚▀▀ ▚"
140 PRINTB$:PRINTA$" ▚ "
150 PRINTC$"▚ ▀"
160 PRINT" ▚ ▀  ▚OOOOOOOOOOOO▚  ▚   ▀
170 PRINTC$
180 PRINT" ▚ ▀  ▚▚*BAR  FRUIT*▚▀  ▚ ▀
190 PRINTC$
200 PRINT" ▚ ▀  ▚▚O▀▚"SPC(10)"▚▚O▀▚  ▚ ▀
210 PRINT" ▚ ▀  ▚▚O▀▚  P 50    ▚▚O▀▚  ▚ ▀
220 PRINT" ▚ ▀  ▚▚OOOOOOOOOOOO▀▚  ▚ ▀
230 PRINTC$:PRINT" ▚ Z=INSERT:S=START "
240 PRINT" ▚    1,2,3=STOP     "
250 C$="▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
260 CO=0:PRINTLEFT$(C$,12)"▚▚▚▚OOOOOOOOOOOO
264 REM
265 REM INPUT CASH OR START
266 REM
270 GETA$:IFA$<>"Z"ANDA$<>"S"THEN270
280 IFA$="Z"THEN310
290 IFCO<>0ANDA$<>"Z"THEN370
300 GOTO270
310 IFCO=12THEN270
320 IFSC=0THEN270
330 CO=CO+1:PRINTLEFT$(C$,12)SPC(4)"▚";
340 PRINTLEFT$("●●●●●●●●●●●●",CO)
350 SC=SC-1:PRINTC$SPC(7)"P"STR$(SC)" "
360 GOTO270
364 REM
365 REM PREPARE STRINGS FOR EACH SYMBOL
366 REM
370 A$(1,1)="▚   ⁄⁄"
380 A$(1,2)="▚⁄⁄ ◡"
390 A$(1,3)="▚◡ ⁄⁄"
400 A$(1,4)="▚   ◡"
410 A$(2,1)="     "
420 A$(2,2)="▚ ◠ "
```

```
430 A$(2,3)="▥◤▼   ▼▀"
440 A$(2,4)="▥◡ ◡"
450 A$(3,1)="▦ ▨▼▀ "
460 A$(3,2)="▦╱●●╲"
470 A$(3,3)="▦| ◡ |"
480 A$(3,4)="▦╰──╯"
490 A$(4,1)="▩ ▨▼▀ "
500 A$(4,2)="▩=▨●●▀="
510 A$(4,3)="▩  ▨▀▄"
520 A$(4,4)="     "
530 A$(5,1)="▓▒▒▒▒▒"
540 A$(5,2)="▓▨BAR▀▒"
550 A$(5,3)="▓▒▨BAR▀"
560 A$(5,4)="▓▒▒▒▒▒"
564 REM
565 REM PULL AND RELEASE ARM
566 REM
570 PRINT"▨▨▨▨▨"SPC(20);
580 FORI=1TO6:PRINT" ▨▌▨●▀▌";
590 FORJ=1TO40:NEXT:NEXT
600 FORI=1TO6:PRINT"▨ ▀▌▌▨●▀▌";
610 FORJ=1TO30:NEXT:NEXT
614 REM
615 REM SPIN REELS
616 REM
620 AH=0:BH=0:CH=0
630 IFAH=3THEN720
640 AP=AP+1:IFAP=5THENA=AR:AP=1
650 AX=A:AY=AP:AZ=AR
660 PRINT"▨▨▨▨▨▨"SPC(3)A$(A,AP)
670 FORI=1TO3:AP=AP+1:IFAP=5THENA=AR:AP=1
680 PRINTSPC(3)A$(A,AP):NEXT
690 IFAP=4ANDAH<>0THENAH=AH+1
700 IFAP=4ANDA=ARTHENAZ=INT(RND(1)*Q)+1
710 A=AX:AP=AY:AR=AZ
720 IFBH=3THEN810
730 BP=BP+1:IFBP=5THENB=BR:BP=1
740 BX=B:BY=BP:BZ=BR
750 PRINT"▨▨▨▨▨▨"SPC(8)A$(B,BP)
760 FORI=1TO3:BP=BP+1:IFBP=5THENB=BR:BP=1
770 PRINTSPC(8)A$(B,BP):NEXT
780 IFBP=4ANDBH<>0THENBH=BH+1
790 IFBP=4ANDB=BRTHENBZ=INT(RND(1)*Q)+1
800 B=BX:BP=BY:BR=BZ
810 IFCH=3THEN900
820 CP=CP+1:IFCP=5THENC=CR:CP=1
830 CX=C:CY=CP:CZ=CR
840 PRINT"▨▨▨▨▨▨"SPC(13)A$(C,CP)
850 FORI=1TO3:CP=CP+1:IFCP=5THENC=CR:CP=1
860 PRINTSPC(13)A$(C,CP):NEXT
870 IFCP=4ANDCH<>0THENCH=CH+1
```

```
880 IFCP=4ANDC=CRTHENCZ=INT(RND(1)*Q)+1
890 C=CX:CP=CY:CR=CZ
900 IFAH=3ANDBH=3ANDCH=3THEN960
904 REM
905 REM INPUT KEYS TO STOP REELS
906 REM
910 GETA$
920 IFAH<>3ANDA$="1"THENAH=1
930 IFBH<>3ANDA$="2"THENBH=1
940 IFCH<>3ANDA$="3"THENCH=1
950 GOTO630
954 REM
955 REM WHEN ALL REELS STOPPED, CHECK FOR WIN
956 REM
960 HT=0:XX=0:IFA=1THENXX=2
970 IFA=1ANDB=1THENXX=5
980 IFA=1ANDB=1ANDC=1THENXX=10
990 IFA=2ANDB=2ANDC=2THENXX=20
1000 IFA=3ANDB=3ANDC=3THENXX=50
1010 IFA=4ANDB=4ANDC=4THENXX=20
1020 IFA=5ANDB=5ANDC=5THENXX=100
1030 XX=XX*CO
1040 IFXX=0THEN1220
1044 REM
1045 REM WIN, DISPLAY PAYOUT
1046 REM
1050 PRINT"▓";:FORI=1TOXX
1060 FORJ=1TO10:NEXT
1070 IFHT=0THENPRINT"▚▩▩"SPC(8)"◪  ■  "
1080 IFHT=1ORHT=5THENPRINT"▚▩▩"SPC(8)" ◪ ■"
1090 IFHT=2ORHT=4THENPRINT"▚▩▩"SPC(8)"  ◪ ■ "
1100 IFHT=3THENPRINT"▚▩▩"SPC(8)"  ◪  ■"
1110 HT=HT+1:IFHT=6THENHT=0
1120 SC=SC+1:PRINTC$SPC(7)"P"STR$(SC)
1130 NEXT:PRINT"▓"
1140 PRINT"▚▩▩"SPC(8)"      ":GOTO260
1144 REM
1145 REM CHOSE NEXT SYMBOL TO APPEAR
1146 REM
1150 AP=0:BP=0:CP=0
1160 A=INT(RND(1)*Q)+1
1170 AR=INT(RND(1)*Q)+1
1180 B=INT(RND(1)*Q)+1
1190 BR=INT(RND(1)*Q)+1
1200 C=INT(RND(1)*Q)+1
1210 CR=INT(RND(1)*Q)+1:RETURN
1214 REM
1215 REM OUT OF MONEY
1216 REM
1220 IFSC<>0THEN260
1230 FORJ=1TO1000:NEXT
```



```
1240 PRINT"]QQQQQQQQQQ 3ANOTHER GO (Y OR N) ?"
1250 GETA$:IFA$<>"Y"ANDA$<>"N"THEN1250
1260 IFA$="Y"THENRUN
1270 PRINT"Q THE END":END
READY.
```

MOON LANDER

DESCRIPTION

The aim of the game is to land a space module on the surface of the moon without crashing. It requires a great deal of skill to place the space ship on the surface, caution is advisable when judging the speed of your approach.

The display on the screen for this game is very simple, consisting of a lunar module in the right hand corner of the screen with an information gauge on the left, indicating the height, speed, time and fuel use.

Another feature of the game is the sound when fuel is being used.

One final note, it is possible to land the space ship without crashing.

For those interested the game could be modified to use a joystick instead of the keyboard, better graphics could also be added.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

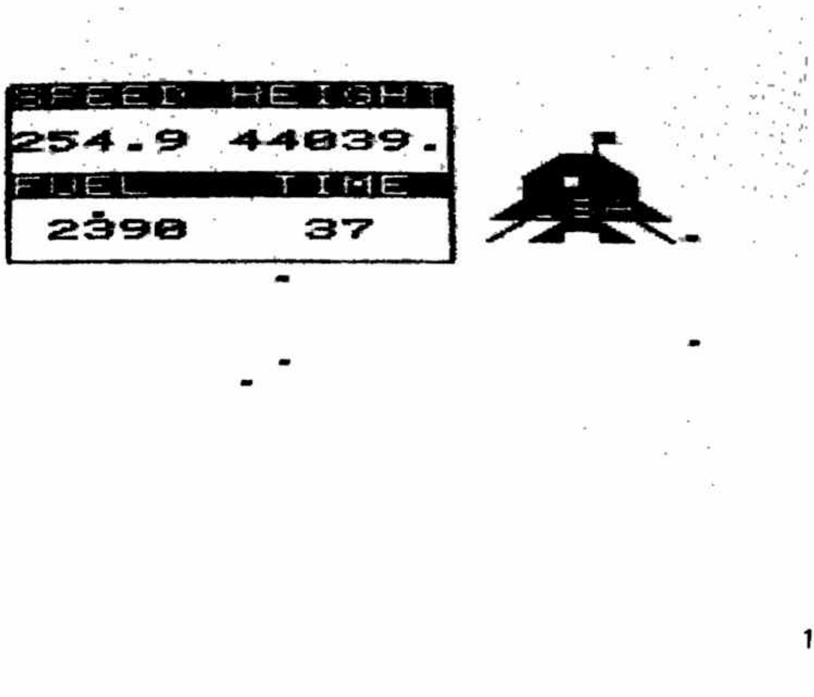
RUNNING THE PROGRAM

After typing run the program displays a set of instructions for playing the game. Pressing the space bar will start the game. The display shows an information gauge on the left hand side of the screen and a lunar module on the right hand side. There is an initial countdown of several seconds after which the game starts. The speed of the lander's descent can be controlled by burning fuel in the retro-rocket; the amount of fuel used is controlled by the number keys 1 to 9. Key 1 uses very little fuel and therefore slows the descent only slightly; note that the speed shown on the gauge will increase for all fuel usages below 5 since we are having to counteract the effect of gravity. A value of 5 will just counteract gravitational pull and the spacecraft speed will remain constant. Keep an eye on the fuel gauge since running out of fuel will result in an uncontrolled crash to the surface. A successful landing on the surface will be achieved if your speed is less than 15 Km/hr.

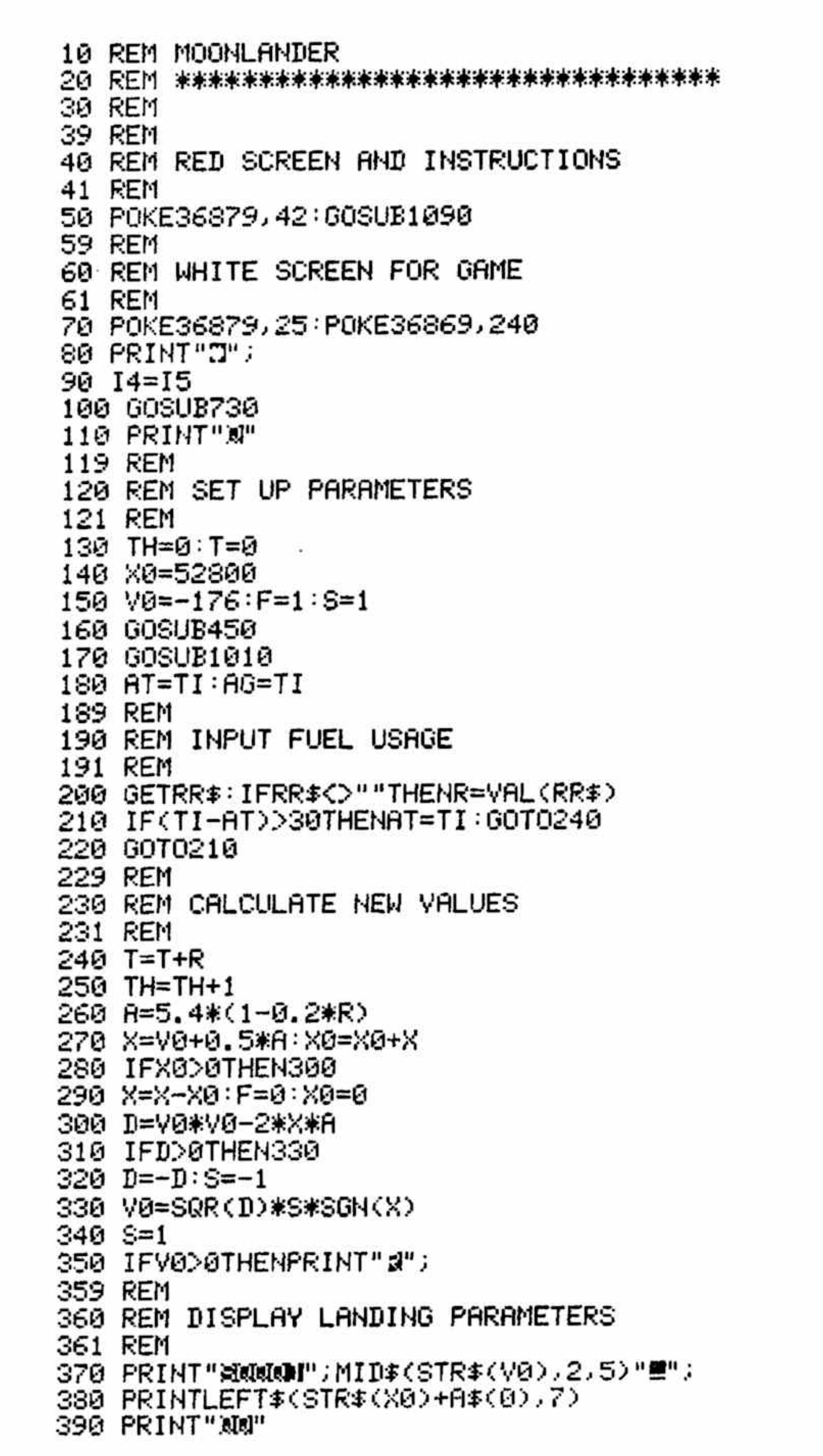
PROGRAM STRUCTURE.

The lines of interest in this game are as follows.

50 Colours the screen red and uses a subroutine to print the instructions for the game.
70 Returns the screen to white for play to commence.
130-170 Sets up land parameters for intial fuel, speed and height information.
200-220 Inputs fuel usage from the keyboard.
240-350 Calculates new values ofspeed and height.
370-430 Displays the above values.
450-560 Prints the space ship on the screen.
580-630 Prints box around landing parameters.
660-710 Displays burn out combustion lines.
730-770 Prints random dots to represent stars.
790-850 Displays moon surface when height becomes 0.
880-920 Displays landed ship.
940-960 Displays crashed ship.
970-990 Inputs new go.
1010-1060 Inputs the 5 second count down at the beginning of the game.
1090-1230 Prints instructions for the game.



```
SPEED  HEIGHT
254.9  44039.
FUEL     TIME
 2390     37
```



```
10 REM MOONLANDER
20 REM *********************************
30 REM
39 REM
40 REM RED SCREEN AND INSTRUCTIONS
41 REM
50 POKE36879,42:GOSUB1090
59 REM
60 REM WHITE SCREEN FOR GAME
61 REM
70 POKE36879,25:POKE36869,240
80 PRINT"■";
90 I4=I5
100 GOSUB730
110 PRINT"■"
119 REM
120 REM SET UP PARAMETERS
121 REM
130 TH=0:T=0
140 X0=52800
150 V0=-176:F=1:S=1
160 GOSUB450
170 GOSUB1010
180 AT=TI:AG=TI
189 REM
190 REM INPUT FUEL USAGE
191 REM
200 GETRR$:IFRR$<>""THENR=VAL(RR$)
210 IF(TI-AT)>30THENAT=TI:GOTO240
220 GOTO210
229 REM
230 REM CALCULATE NEW VALUES
231 REM
240 T=T+R
250 TH=TH+1
260 A=5.4*(1-0.2*R)
270 X=V0+0.5*A:X0=X0+X
280 IFX0>0THEN300
290 X=X-X0:F=0:X0=0
300 D=V0*V0-2*X*A
310 IFD>0THEN330
320 D=-D:S=-1
330 V0=SQR(D)*S*SGN(X)
340 S=1
350 IFV0>0THENPRINT"■";
359 REM
360 REM DISPLAY LANDING PARAMETERS
361 REM
370 PRINT"■■■■■";MID$(STR$(V0),2,5)"■";
380 PRINTLEFT$(STR$(X0)+A$(0),7)
390 PRINT"■■"
```

```
400 PRINT"█";2500-T"█    ██"TAB(8);TH
410 GOSUB660:IFF=0THENGOSUB790:GOTO880
420 IFT>2500THENR=0:GOTO250
430 GOTO200
439 REM
440 REM INITIALISE STRINGS FOR SHIP
441 REM
450 A$="██████"
460 A$(0)="      "
470 A$(1)="█  ▛ "
480 A$(2)="▪ ◩ ▀▀ "
490 A$(3)="▪ ◩ ▪ ▀ "
500 A$(4)="▪◩USA▀"
510 A$(5)="▪/◩▄▀\"
520 A$(6)=A$(0)
530 PRINT"███"TAB(15);
540 FORI=0TO6
550 PRINTA$(I)+A$;
560 NEXT
569 REM
570 REM PRINT DISPLAY AROUND PARAMETERS
571 REM
580 PRINT"██ ______________"
590 PRINT" █SPEED HEIGHT█ ":GOSUB620
600 PRINT" █FUEL   TIME █ ":GOSUB620
610 RETURN
620 FORI=1TO2:PRINT" |"TAB(13)"| ":NEXT
630 PRINT" L______________J ":RETURN
640 REM IF KEY GREATER THAN 0 PRESSED,
650 REM DISPLAY 'BURN'
660 PRINTTAB(15);:IFR=0THENPOKE36878,0:RETURN
670 POKE36877,150:POKE36874,200:POKE36878,R
680 FORI=1TOR:PRINT"█/‾\█████";:NEXT
690 FORI=RTO10:PRINT"    ████";:NEXT
700 PRINT"TTTTTTTTTTT";:FORI=1TO10
710 PRINT"    ████";:NEXT:RETURN
719 REM
720 REM PRINT 14 RANDOM '.' ON SCREEN
721 REM
730 PRINT"██"
740 FORI1=1TO14
750 PRINTTAB(20*RND(1));"."
760 NEXTI1
770 RETURN
779 REM
780 REM DISPLAY SURFACE
781 REM
790 H=4.5:H$(1)="_":H$(2)="▁":H$(3)="▂"
800 H$(4)="▄":H$(5)="◩▔▀":H$(6)="◩▔▀"
810 H$(7)="◩▔▀":H$(8)="◩ ▀"
820 PRINT"█████████████████████":FORI=1TO21
```

```
830 N=INT(RND(1)*7)+1:IFABS(N-H)>2THEN830
840 POKE36877,0:POKE36874,0:POKE36878,0
850 H=N:PRINTH$(H);:NEXTI:RETURN
859 REM
860 REM SHIP HAS LANDED OR CRASHED
870 REM DISPLAY SHIP AS NORMAL
871 REM
880 PRINT"█████"SPC(15);
890 FORI=1TO10:FORJ=0TO5:PRINTA$(J)+A$;
900 NEXT:PRINT"▌▌▌▌▌▌ ";:NEXT
910 IFV0<-15THEN940
920 PRINT:PRINT"███████GOOD LANDING.":GOTO970
929 REM
930 REM DISPLAY CRASHED SHIP
931 REM
940 PRINT"█    ██████     ██████     ████████    ";
950 PRINT"██████▌▖ ███████████████▀▚▚ ▀▚███▌└ ▚ ▀▀"
960 PRINT"██"
970 INPUT"█ANOTHER GO ";G$
980 IFLEFT$(G$,1)="Y"THEN90
990 END
999 REM
1000 REM COUNTDOWN TO START
1001 REM
1010 PRINT"█████████████████████████"
1020 A=TI
1030 PRINT"█START OVER"5-INT((TI-A)/60);
1040 PRINT"█ SEC.▌"
1050 IF(TI-A)<300THEN1030
1060 PRINT"                              ":R=0
1070 RETURN
1079 REM
1080 REM INSTRUCTIONS
1081 REM
1090 POKE36869,242:REM IN LOWER CASE
1100 PRINT"██***** MOONLANDER *****
1110 PRINT"YOU MUST LAND YOUR"
1120 PRINT" SPACE SHIP ON THE      MOON."
1130 PRINT"█YOU HAVE CONTROL OVER  THE ";
1140 PRINT"SHIP BY CHANGING  THE FUEL USAGE."
1150 PRINT"█YOU CHANGE THE FUEL    USAGE ";
1160 PRINT"WITH THE KEYS   0..9 - 5 KEEPS"
1170 PRINT" CONSTANT SPEED."
1180 PRINT"█YOU MUST LAND WITH A   SPEED ";
1190 PRINT"OF LESS THAN     15 ⁄M/H."
1200 PRINT"█GOOD LUCK."
1210 PRINT"██START WITH A SPACE.█"
1220 GETA$:IFA$=""THEN1220
1230 RETURN
READY.
```

CIRCLE DEMO

DESCRIPTION

This program plots two high resolution circles on the screen of equal radii but with different centre co-ordinates. No values are input.

EQUIPMENT REQUIRED

Basic VIC-20.

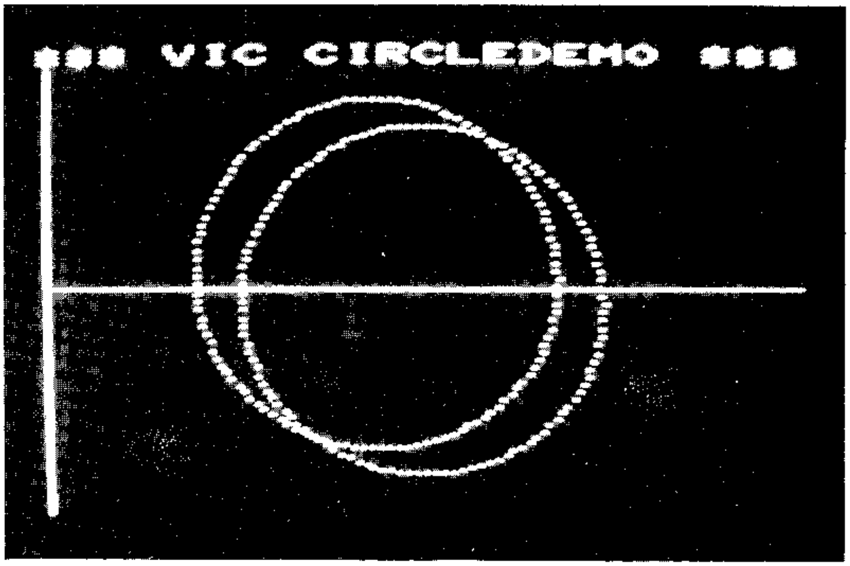
RUNNING THE PROGRAM

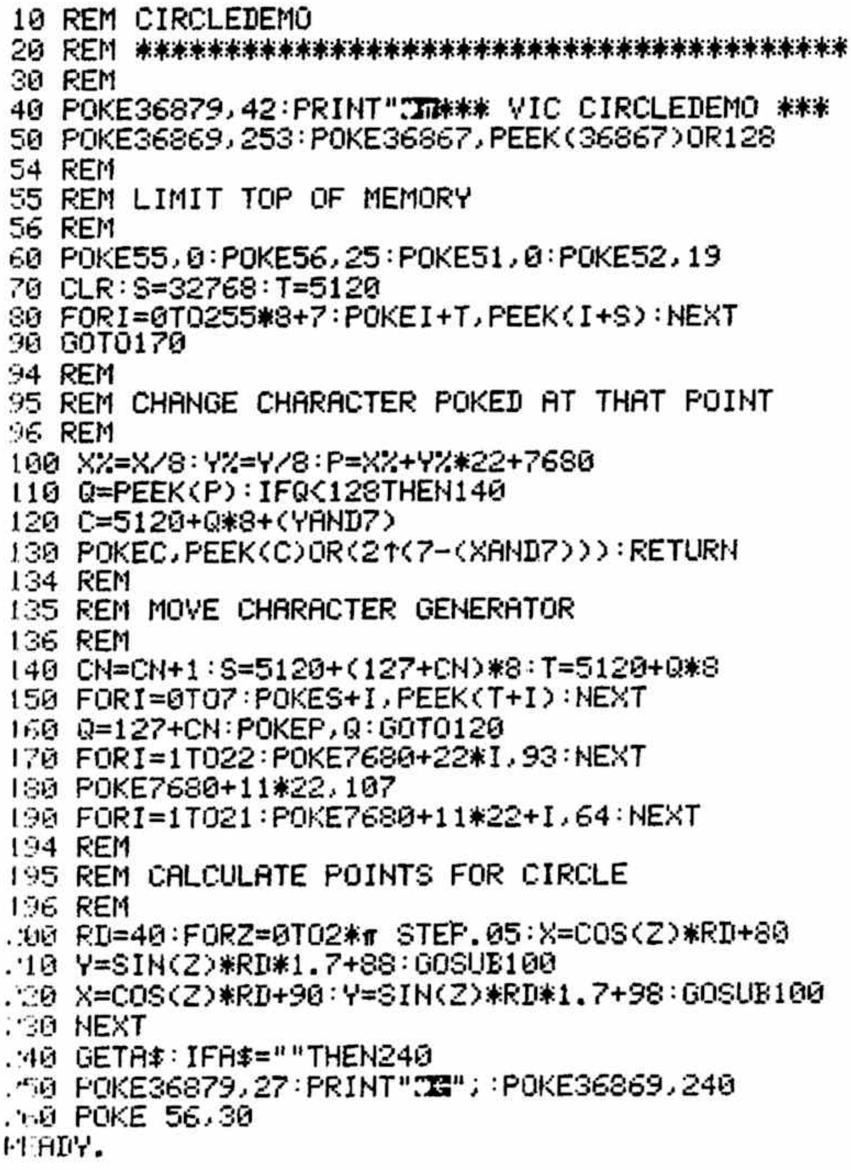
This program will not work with expansion. After loading the program all the user has to do is type run and watch the circles appear on the screen. To stop the program hit any Key.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows.

40-50 Sets up the control registers.
60-90 Limits memory and moves character values down from ROM.
100-190 Changes the character in X%,Y% point on the screen.
200-230 Calculates the points on the circle.
240-260 If key press, end.


$$$ VIC CIRCLEDEMO $$$



```
10 REM CIRCLEDEMO
20 REM ****************************************
30 REM
40 POKE36879,42:PRINT"▮▮*** VIC CIRCLEDEMO ***
50 POKE36869,253:POKE36867,PEEK(36867)OR128
54 REM
55 REM LIMIT TOP OF MEMORY
56 REM
60 POKE55,0:POKE56,25:POKE51,0:POKE52,19
70 CLR:S=32768:T=5120
80 FORI=0TO255*8+7:POKEI+T,PEEK(I+S):NEXT
90 GOTO170
94 REM
95 REM CHANGE CHARACTER POKED AT THAT POINT
96 REM
100 X%=X/8:Y%=Y/8:P=X%+Y%*22+7680
110 Q=PEEK(P):IFQ<128THEN140
120 C=5120+Q*8+(YAND7)
130 POKEC,PEEK(C)OR(2↑(7-(XAND7))):RETURN
134 REM
135 REM MOVE CHARACTER GENERATOR
136 REM
140 CN=CN+1:S=5120+(127+CN)*8:T=5120+Q*8
150 FORI=0TO7:POKES+I,PEEK(T+I):NEXT
160 Q=127+CN:POKEP,Q:GOTO120
170 FORI=1TO22:POKE7680+22*I,93:NEXT
180 POKE7680+11*22,107
190 FORI=1TO21:POKE7680+11*22+I,64:NEXT
194 REM
195 REM CALCULATE POINTS FOR CIRCLE
196 REM
200 RD=40:FORZ=0TO2*π STEP.05:X=COS(Z)*RD+80
210 Y=SIN(Z)*RD*1.7+88:GOSUB100
220 X=COS(Z)*RD+90:Y=SIN(Z)*RD*1.7+98:GOSUB100
230 NEXT
240 GETA$:IFA$=""THEN240
250 POKE36879,27:PRINT"▮▮";:POKE36869,240
260 POKE 56,30
READY.
```

HI-RES PLOT

DESCRIPTION

This program allows the user to plot high resolution points on the screen using P, U, F and E keys.
P: Plots points.
U: Inverts the point at those co-ordinates.
F: Plots up function defined in line 500.
E: End of program.

EQUIPMENT REQUIRED

Basic VIC-20.

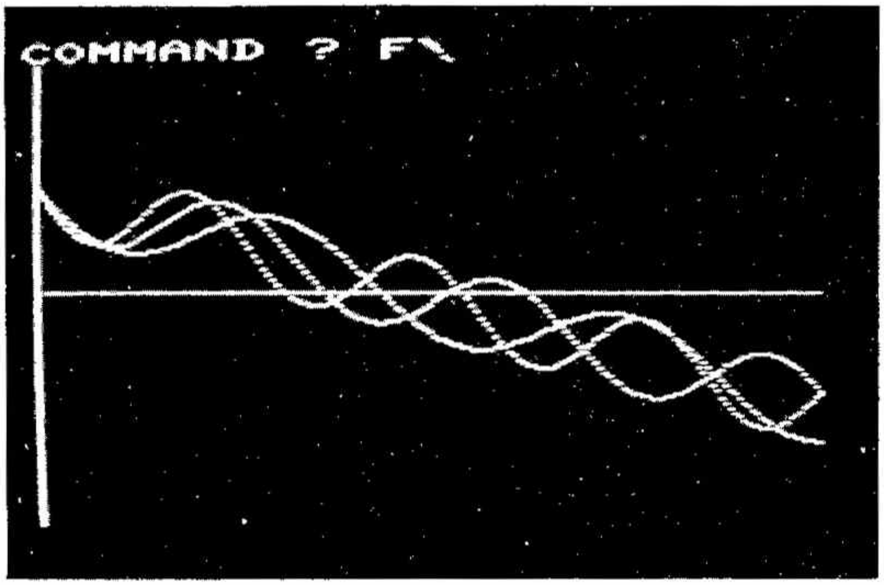
RUNNING THE PROGRAM

This program requires no expansion. When typing in the program leave out the REMs, type RUN, use the commands above and follow the instruction which will then be given.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows:

100-110	Sets up control registers.
150-290	Limits memory, moves character values down from ROM, inputs option, and decides where to go.
330-340	Plot the point P option.
380-390	Plot the point U option.
460-510	Plot the function defined in line 500.



```
COMMAND ? F\
 10 REM HI-RES PLOT
 20 REM ************************************
 30 REM
 40 REM
 50 REM THIS PROGRAM WILL ONLY WORK
 60 REM WITH ALL THE REM STATEMENTS
 70 REM REMOVED
 80 REM
 90 PRINT"██*** VIC HIRES-PLOT ***"
 100 POKE36879,42:POKE36869,253
 110 POKE36867,PEEK(36867)OR128
 120 REM
 130 REM LIMIT TOP OF MEMORY
 140 REM
 150 POKE55,0:POKE56,25:POKE51,0
 160 POKE52,19:CLR:S=32768:T=5120
 170 FORI=0TO255*8+7:POKEI+T,PEEK(I+S)
 180 NEXT:PRINT"█
 190 INPUT"█COMMAND ";A$
 200 IFA$<>"E"THEN230
 210 POKE36879,27:PRINT"██";
 220 POKE 36869,240:POKE56,30:END
 230 IFA$<>"P"ANDA$<>"U"THEN250
 240 INPUT"█X,Y      ";X,Y:GOSUB270
 250 IFA$="F"THEN460
 260 GOTO190
 270 X%=X/8:Y%=Y/8:P=X%+Y%*22+7680
 280 Q=PEEK(P):IFQ<128THEN400
 290 IFA$<>"U"THEN380
 300 REM
 310 REM PLOT THE POINT (0,0=TOP LEFT)
 320 REM
 330 C=5120+Q*8+(YAND7)
 340 POKEC,PEEK(C)AND(255-2↑(7-(XAND7))):RETURN
 350 REM
 360 REM PLOT THE POINT INVERSE TO WHAT IT IS
 370 REM
 380 C=5120+Q*8+(YAND7)
 390 POKEC,PEEK(C)OR(2↑(7-(XAND7))):RETURN
 400 CN=CN+1:S=5120+(127+CN)*8:T=5120+Q*8
 410 FORI=0TO7:POKES+I,PEEK(T+I):NEXT
 420 Q=127+CN:POKEP,Q:GOTO380
 430 REM
 440 REM PLOT FUNCTION IN LINE 500
 450 REM
 460 PRINT"█":FORI=0TO22:POKE7680+22*I,93
 470 NEXT:POKE7680+11*22,107:DATA15,23,19
 480 FORI=1TO21:POKE7680+11*22+I,64:NEXT
 490 FORZ=1TO3:READB:FORX=4TO175
 500 Y=.5*X+50+SIN((X-4)/176*B)*15
 510 GOSUB270:NEXT:NEXT:RESTORE:GOTO190
READY.
```

HANGMAN

DESCRIPTION

Hangman is the computerised version of the popular childs game, although playing against the computer enlivens the proceeding well enough for adults to find it challenging. The player guessing the word has ten moves in which to work his miracle (the computer always plays as the questioner!), no second chances are allowed as first the ground appears, followed by the upright, then the crossbar, the strut, the noose, the hanged man's head, his body, arms, legs, and finally his feet. If the correct answer has not been found by this time then the player who is trying to find the word loses.
In this particular program the computer has a vocabulary comprising some of the 200 most commonly misspelled words in the English language.
It is possible to cheat by looking at the vocabulary listing but most people would, I hope, try to use the game in a more sporting way. By changing the words in the Data statements in lines 9500 to 9998 - always ensuring that there are 200 words contained therein - it is possible to use this game as an aid to learning a foreign language.

EQUIPMENT REQUIRED

Basic VIC-20 with 3K expansion.

RUNNING THE PROGRAM

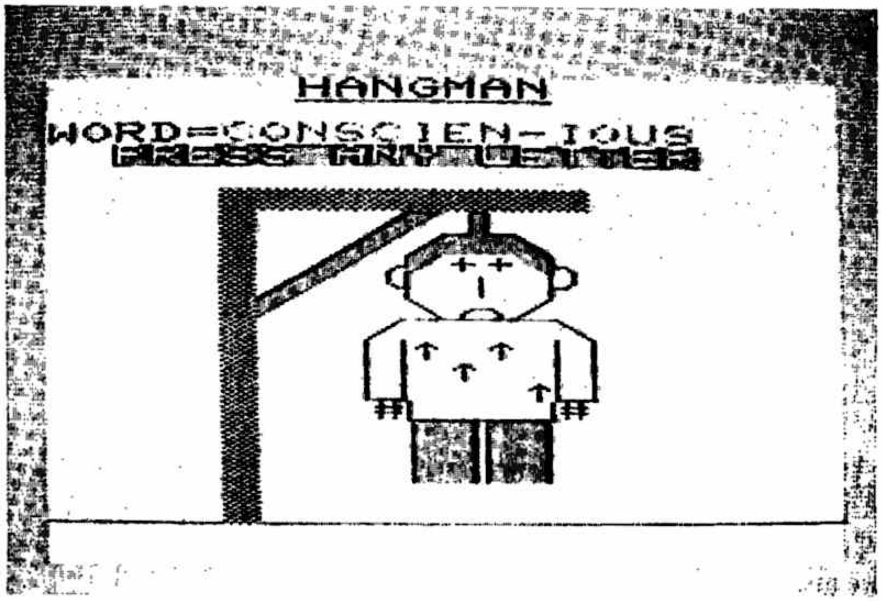
This program will only work with 3K expansion if the REM statements are removed. Any expansion above 3K will work with the REM statements included.
Once the program is loaded, type run and follow the instructions.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

40-380	Instructions.
420-560	Choose a word from the data statements.

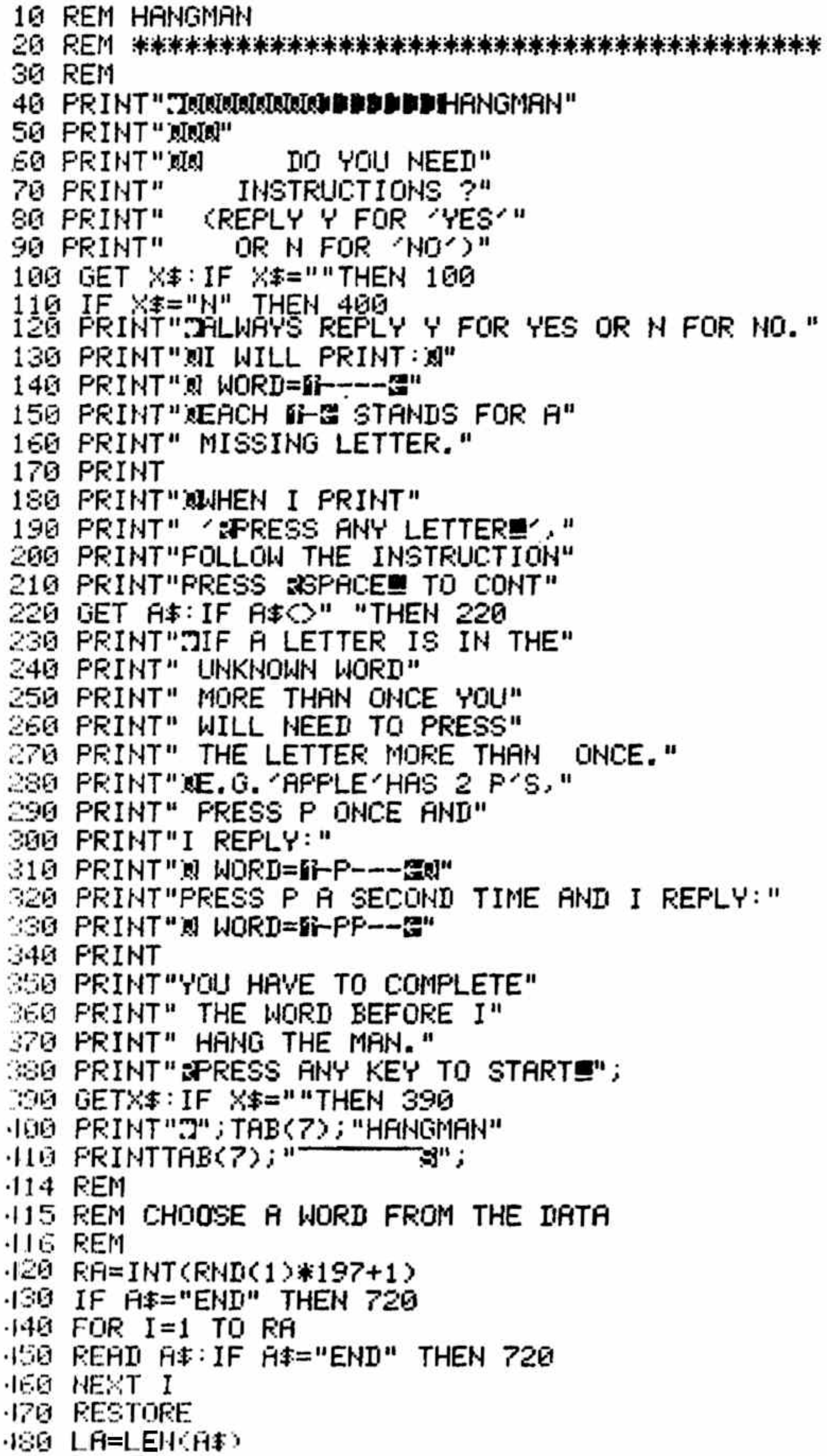
570-590	Input a letter.
600-660	Not a letter.
670-760	Guessed the word.
770-800	Letter in word.
810-890	Print letter in the correct position on the screen.
900-950	Letter is not in the word.
960-1000	Display the man's feet.
1010-1030	Failed to guess the word before the man was hung.
1040-1070	Display the base of the rig.
1080-1140	Display the upright.
1150-1180	Display the horizontal.
1190-1220	Display the diagonal support.
1230-1260	Display the rope.
1270-1340	Display the man's head.
1350-1420	Display the man's body.
1430-1460	Display the man's hands.
1470-1530	Display the man's legs.
1720-2360	Data containing about 200 words.



HANGMAN

WORD=CONSCIEN-TIOUS

PRESS ANY LETTER



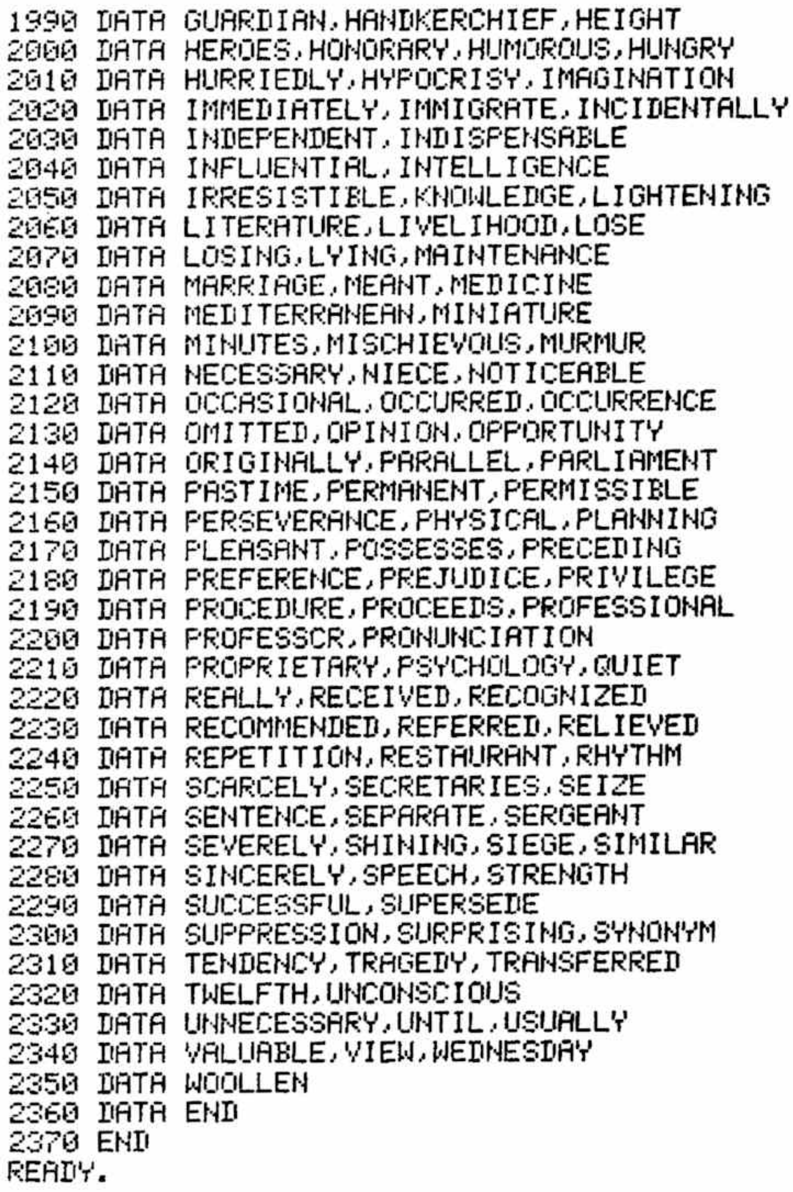
```
10 REM HANGMAN
20 REM ****************************************
30 REM
40 PRINT"♥◙◙◙◙◙◙◙◙◙▶▶▶▶▶▶HANGMAN"
50 PRINT"◙◙◙"
60 PRINT"◙◙      DO YOU NEED"
70 PRINT"     INSTRUCTIONS ?"
80 PRINT"  (REPLY Y FOR 'YES'"
90 PRINT"     OR N FOR 'NO')"
100 GET X$:IF X$=""THEN 100
110 IF X$="N" THEN 400
120 PRINT"♥ALWAYS REPLY Y FOR YES OR N FOR NO."
130 PRINT"◙I WILL PRINT:◙"
140 PRINT"◙ WORD=▚-----■"
150 PRINT"◙EACH ▚-■ STANDS FOR A"
160 PRINT" MISSING LETTER."
170 PRINT
180 PRINT"◙WHEN I PRINT"
190 PRINT" '▚PRESS ANY LETTER■',"
200 PRINT"FOLLOW THE INSTRUCTION"
210 PRINT"PRESS ▚SPACE■ TO CONT"
220 GET A$:IF A$<>" "THEN 220
230 PRINT"♥IF A LETTER IS IN THE"
240 PRINT" UNKNOWN WORD"
250 PRINT" MORE THAN ONCE YOU"
260 PRINT" WILL NEED TO PRESS"
270 PRINT" THE LETTER MORE THAN  ONCE."
280 PRINT"◙E.G.'APPLE'HAS 2 P'S,"
290 PRINT" PRESS P ONCE AND"
300 PRINT"I REPLY:"
310 PRINT"◙ WORD=▚-P---■◙"
320 PRINT"PRESS P A SECOND TIME AND I REPLY:"
330 PRINT"◙ WORD=▚-PP--■"
340 PRINT
350 PRINT"YOU HAVE TO COMPLETE"
360 PRINT" THE WORD BEFORE I"
370 PRINT" HANG THE MAN."
380 PRINT"▚PRESS ANY KEY TO START■";
390 GETX$:IF X$=""THEN 390
400 PRINT"♥";TAB(7);"HANGMAN"
410 PRINTTAB(7);"▔▔▔▔▔▔▔▔▔▔▔◙";
414 REM
415 REM CHOOSE A WORD FROM THE DATA
416 REM
420 RA=INT(RND(1)*197+1)
430 IF A$="END" THEN 720
440 FOR I=1 TO RA
450 READ A$:IF A$="END" THEN 720
460 NEXT I
470 RESTORE
480 LA=LEN(A$)
```

```
490 Z$=CHR$(18)+A$
500 D$=""
510 FOR I=1 TO LA
520 D$=D$+"-"
530 NEXT I
540 PRINT"▒"
550 PRINT"WORD=▒";D$;"▒▒";
560 N=0:L=0
564 REM
565 REM INPUT A LETTER
566 REM
570 PRINT"▒▒▒  ▒PRESS ANY LETTER■     ▒";
580 GET L$:IF L$=""THEN 580
590 IF ASC(L$)>64 AND ASC(L$)<91 THEN 640
594 REM
595 REM NOT A LETTER
596 REM
600 PRINT"▒▒▒"L$" IS NOT A LETTER      "
610 GOSUB 1690
620 PRINT"⊐                        ▒";
630 GOTO 570
640 GOSUB 730
650 IF N<LA THEN 570
660 IF N=(LA+1) THEN T1=TI:GOTO 680
664 REM
665 REM FINISHED
666 REM
670 PRINT"▒▒▒YOU WIN!!              ":T1=TI
680 IF(TI-T1)<200 THEN 680
690 PRINT"⊐ DO YOU WANT ANOTHER    GO?";
700 GET B$:IF B$=""THEN 700
710 IF B$="Y" THEN 400
720 PRINT"⊐ THANK-YOU FOR PLAYING":END
730 LF=0
740 FOR I=1 TO LA
750 B$=MID$(A$,I,1)
760 IF B$<>L$ THEN 870
764 REM
765 REM LETTER IN WORD
766 REM
770 PRINT"▒▒▒LETTER IN WORD        ▒";
780 N=N+1
790 GOSUB 1690
800 IA=4+I
804 REM
805 REM PRINT LETTER IN POSITION
806 REM
810 PRINT"▒"
820 PRINT TAB(IA);"▒"L$"▒";"▒";
830 II=I
840 GOSUB 1540
```

```
850 LF=LF+1
860 RETURN
870 NEXTI
880 IF LF=0 THEN GOSUB 900
890 RETURN
894 REM
895 REM LETTER NOT IN WORD
896 REM
900 L=L+1
910 PRINT"▨▨▨LETTER NOT IN WORD   ▨"
920 GOSUB 1690
930 ON L GOSUB 1040,1080,1150,1190,1230
940 IF L>5 THENON L-5 GOSUB1270,1350,1430,1470
950 IF L<10 THEN RETURN
954 REM
955 REM PRINT FEET OF MAN
956 REM
960 K=19
970 GOSUB 1650
980 PRINTTAB(9);" ▨▨ ▨ ▨ ▨ ▨";
990 PRINT"▨"
1000 PRINTTAB(5)"▨"Z$"▨"
1004 REM
1005 REM FAILED TO GET WORD
1006 REM
1010 PRINT"SORRY,YOU LOST         ▨";
1020 N=LA+1
1030 RETURN
1034 REM
1035 REM PRINT BASE
1036 REM
1040 K=20
1050 GOSUB 1650
1060 PRINT"____________________▨";
1070 RETURN
1074 REM
1075 REM PRINT UPRIGHT
1076 REM
1080 K=6
1090 GOSUB 1650
1100 FOR I=1 TO 14
1110 PRINTTAB(5);"▒"
1120 NEXT I
1130 PRINTTAB(5);"▒▨";
1140 RETURN
1144 REM
1145 REM PRINT HORIZONTAL
1146 REM
1150 K=5
1160 GOSUB 1650
1170 PRINTTAB(5);"▒▒▒▒▒▒▒▒▒▒▒▨";
```

```
1180 RETURN
1184 REM
1185 REM PRINT DIAGONAL SUPPORT
1186 REM
1190 K=10
1200 GOSUB 1650
1210 PRINTTAB(6);"◤▌⊓◪◢▬◤▌⊓◪◢▬◤▌⊓◪◢▬◤▌⊓◪◢▬◤▨";
1220 RETURN
1224 REM
1225 REM PRINT ROPE
1226 REM
1230 K=6
1240 GOSUB 1650
1250 PRINTTAB(11);" ▌▨";
1260 RETURN
1264 REM
1265 REM PRINT HEAD
1266 REM
1270 K=7
1280 GOSUB 1650
1290 PRINTTAB(10);"▨◤   ◥"
1300 PRINTTAB(9);" ◢◤++◥◣"
1310 PRINTTAB(9);" ╕ │ ╒"
1320 PRINTTAB(10);"╲⌒╱"
1330 PRINTTAB(11);"‾‾▨";
1340 RETURN
1344 REM
1345 REM PRINT BODY
1346 REM
1350 K=11
1360 GOSUB 1650
1370 PRINTTAB(9);"╱‾‾‾‾‾╲"
1380 PRINTTAB(8);" │|↑ ↑ ││ "
1390 PRINTTAB(8);" ││ ↑  ││ "
1400 PRINTTAB(8);" └┘   ↑└┘ "
1410 PRINTTAB(10);"└────▨";
1420 RETURN
1424 REM
1425 REM PRINT HANDS
1426 REM
1430 K=15
1440 GOSUB 1650
1450 PRINTTAB(9);"#└───┘#▨";
1460 RETURN
1464 REM
1465 REM PRINT LEGS
1466 REM
1470 K=16
1480 GOSUB 1650
1490 FOR I=1 TO 3
1500 PRINTTAB(10);"▨ ▌ ╷"
```

```
1510 NEXT I
1520 PRINT"█";
1530 RETURN
1540 X=II-1:Y=LA-II
1550 IF X=0 AND Y=0 THEN 1600
1560 IF X=0 THEN A$=" "+RIGHT$(A$,Y):RETURN
1570 IF Y=0 THEN A$=LEFT$(A$,X)+" ":RETURN
1580 A$=LEFT$(A$,X)+" "+RIGHT$(A$,Y)
1590 RETURN
1600 A$=""
1610 FOR I=1 TO LA
1620 A$=A$+" "
1630 NEXTI
1640 RETURN
1650 FOR I=2 TO K
1660 PRINT"█";
1670 NEXT I
1680 RETURN
1690 T=TI
1700 IF(TI-T)<100 THEN 1700
1710 RETURN
1714 REM
1715 REM WORD DATA
1716 REM
1720 DATA ABSENCE,ACCEPT,ACCIDENTALLY
1730 DATA ACCOMMODATE,ACHIEVED,ACKNOWLEDGE
1740 DATA ACQUAINTED,ADDRESSES,AERIAL
1750 DATAAGGRAVATE,AGREEABLE,AMATEUR,AMONG
1760 DATA ANTARCTIC,ANXIETY,APPARENT,APPEARANCE
1770 DATAAPPROPRIATE,ARCTIC,ARGUMENT,ARRANGEMENTS
1780 DATA ASCENT,ATHLETIC,AWFUL,BACHELOR
1790 DATA BEGINNING,BELIEVED,BENEFITED,BREATHE
1800 DATA BRITAIN,BUSINESS,CAPTAIN,CEILING
1810 DATA CEMETERY,CERTAIN,CHOICE,CLOTHES
1820 DATA COLLEGE,COMING,COMMITTEE,COMPARATIVE
1830 DATA COMPETENT,COMPLETELY,CONSCIENTIOUS
1840 DATA CONSCIOUS,CONSISTENT,CONVENIENCE
1850 DATA COPIES,COURSE,COURTEOUS,COURTESY
1860 DATA CRITICISM,DECEIVE,DECISION,DEFINITE
1870 DATA DESIRABLE,DESPERATE,DISAPPEARED
1880 DATA DISAPPOINTED,DISASTROUS,DISCIPLINE
1890 DATA DISSATISFIED,EFFICIENCY,EIGHTH
1900 DATA ELIMINATED,EMBARRASSED,EMPHASIZE
1910 DATA ENTHUSIASM,EQUIPPED,ESPECIALLY
1920 DATA ESSENTIAL,EXAGGERATED,EXCELLENT
1930 DATA EXERCISE,EXHAUSTED,EXISTENCE
1940 DATA EXPENSE,EXPERIENCE,FAMILIAR
1950 DATA FEBRUARY,FINANCIAL,FOREIGN
1960 DATA FORMERLY,FORTY,FRIEND,GAUGE
1970 DATA GENIUS,GOVERNMENT,GRAMMAR
1980 DATA GRAMOPHONE,GRIEVANCE,GUARD
```



```
1990 DATA GUARDIAN,HANDKERCHIEF,HEIGHT
2000 DATA HEROES,HONORARY,HUMOROUS,HUNGRY
2010 DATA HURRIEDLY,HYPOCRISY,IMAGINATION
2020 DATA IMMEDIATELY,IMMIGRATE,INCIDENTALLY
2030 DATA INDEPENDENT,INDISPENSABLE
2040 DATA INFLUENTIAL,INTELLIGENCE
2050 DATA IRRESISTIBLE,KNOWLEDGE,LIGHTENING
2060 DATA LITERATURE,LIVELIHOOD,LOSE
2070 DATA LOSING,LYING,MAINTENANCE
2080 DATA MARRIAGE,MEANT,MEDICINE
2090 DATA MEDITERRANEAN,MINIATURE
2100 DATA MINUTES,MISCHIEVOUS,MURMUR
2110 DATA NECESSARY,NIECE,NOTICEABLE
2120 DATA OCCASIONAL,OCCURRED,OCCURRENCE
2130 DATA OMITTED,OPINION,OPPORTUNITY
2140 DATA ORIGINALLY,PARALLEL,PARLIAMENT
2150 DATA PASTIME,PERMANENT,PERMISSIBLE
2160 DATA PERSEVERANCE,PHYSICAL,PLANNING
2170 DATA PLEASANT,POSSESSES,PRECEDING
2180 DATA PREFERENCE,PREJUDICE,PRIVILEGE
2190 DATA PROCEDURE,PROCEEDS,PROFESSIONAL
2200 DATA PROFESSCR,PRONUNCIATION
2210 DATA PROPRIETARY,PSYCHOLOGY,QUIET
2220 DATA REALLY,RECEIVED,RECOGNIZED
2230 DATA RECOMMENDED,REFERRED,RELIEVED
2240 DATA REPETITION,RESTAURANT,RHYTHM
2250 DATA SCARCELY,SECRETARIES,SEIZE
2260 DATA SENTENCE,SEPARATE,SERGEANT
2270 DATA SEVERELY,SHINING,SIEGE,SIMILAR
2280 DATA SINCERELY,SPEECH,STRENGTH
2290 DATA SUCCESSFUL,SUPERSEDE
2300 DATA SUPPRESSION,SURPRISING,SYNONYM
2310 DATA TENDENCY,TRAGEDY,TRANSFERRED
2320 DATA TWELFTH,UNCONSCIOUS
2330 DATA UNNECESSARY,UNTIL,USUALLY
2340 DATA VALUABLE,VIEW,WEDNESDAY
2350 DATA WOOLLEN
2360 DATA END
2370 END
READY.
```

GOMOKO

DESCRIPTION

In this game the computer takes on the guise of a wise oriental and offers us the ancient game of GO or Gomoko. The game involves two opponents attempting to create vertical, horizontal or diagonal patterns of five disks in a row. The board has 10 vertical and 10 horizontal intersecting lines to form a grid.

Each player takes it in turns to position a disk over an intersection on the grid battling to be the first to create a line of five consecutive disks.

The player will find that the wise oriental (the computer) will take a long time - 40 seconds - to complete his move after calculating all the possible moves. It is possible to win, the player will find it very difficult to do so but will gain a tremendous amount of enjoyment when he does win.

EQUIPMENT REQUIRED

Basic VIC-20.

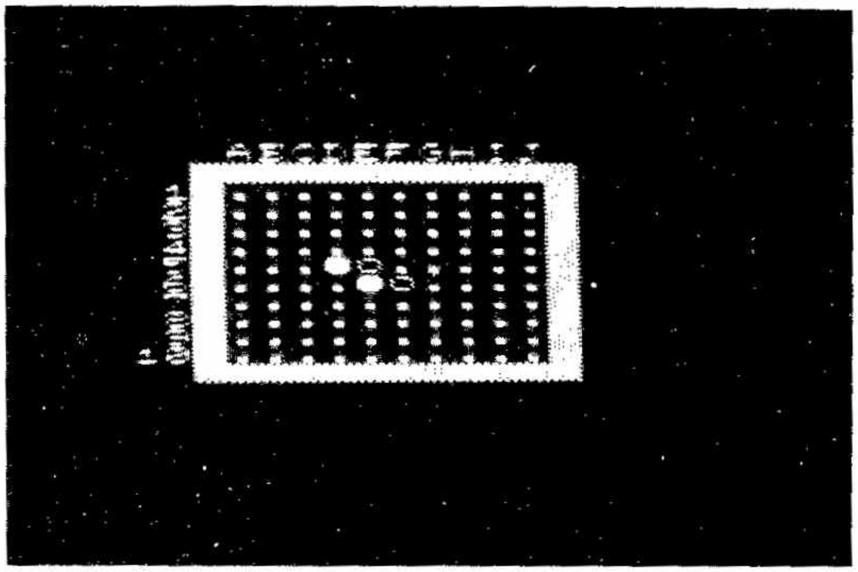
RUNNING THE PROGRAM

After loading type in run and follow the instructions.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

40-170 Main program loop.
180-220 Produce computer's move.
230-360 Input player's move and check for legality.
680-840 Check the move chosen in routine at 180 for validity and calculate for possible win.
1470-1600 Display the board of play.



A B C D E F G H I J

```
10 REM GOMOKO
20 REM ****************************************
30 REM
40 DIM VMAX(2),NF(2),WD(2,20,4)
50 DIM BD(10,10),DB(20):POKE36879,8
60 GOSUB 1470
70 GOSUB 180
80 GOSUB 230
90 GOSUB 1160
100 GOSUB 680
110 IF VMAX(2)=19 THEN WIN=2:GOTO 150
120 GOSUB 370
130 IF VMAX(1)>=14 THEN WIN =1:GOTO 150
140 GOTO 80
150 IF WIN=1 THENPRINT"█I WIN      █":STOP
160 PRINT"█YOU WIN          █"
170 END
174 REM
175 REM PRODUCE COMPUTERS FIRST MOVE
176 REM
180 R1=INT(RND(TI)*6)+3
190 C1=INT(RND(TI)*6)+3
200 PF=7796+22*R1+C1
210 POKE PF,81:POKE PF+30720,5
220 RETURN
224 REM
225 REM INPUT YOUR MOVE
226 REM
230 INPUT "██YOUR MOVE██";R$,C$
240 PRINT"█                 ";"█";
250 R=VAL(R$):C=ASC(C$)-64
260 IFABS(R-5.5)<=4.5ANDABS(C-5.5)<=4.5THEN280
270 GOSUB 1580
280 P=7796+22*R+C
290 D=PEEK(P)
300 IF D=46 THEN 350
310 PRINT"SPACE OCCUPIED"
320 PA=120:GOSUB 1580
330 PRINT"█                     █";
340 GOTO 230
350 POKE P,87:POKE P+30720,2
360 RETURN
370 FOR A=1 TO 2
380 IF VMAX(A)<14 GOTO 410
390 GOSUB 1410
400 RETURN
410 NEXT A
420 FOR A=1 TO 2
430 IF VMAX(A)<10 THEN 470
440 IF A=2 AND VMAX(2)<12 AND VMAX(1)=9THEN470
450 GOSUB 1220
```

```
460 RETURN
470 NEXT A
480 FOR A=1 TO 2
490 IF VMAX(A)<9 THEN 520
500 GOSUB 1220
510 RETURN
520 NEXT A
530 PMAX=0:NT=0
540 FOR R1=1 TO 10
550 FOR C1=1 TO 10
560 IF BD(R1,C1)=0 THEN 630
570 P=BD(R1,C1)
580 IF P<PMAX THEN 630
590 IF P=PMAX THEN 610
600 NT=0:PMAX=P
610 NT=NT+1
620 RM(NT)=R1:CM(NT)=C1
630 NEXT C1,R1
640 PN=INT(RND(1)*NT)+1
650 PF=7796+22*RM(PN)+CM(PN)
660 POKE PF,81:POKE PF+30720,5
670 RETURN
680 V=0
690 FORIL=1 TO 2
700 VMAX(IL)=0:NF(IL)=0
710 NEXT IL
720 RL=1:RU=6
730 CL=1:CU=10:F=22
740 GOSUB 850
750 RL=1:RU=10
760 CL=1:CU=6:F=1
770 GOSUB850
780 RL=1:RU=6
790 CL=1:CU=6:F=23
800 GOSUB850
810 RL=1:RU=6
820 CL=5:CU=10:F=21
830 GOSUB850
840 RETURN
850 FOR R=RL TO RU
860 S=7796+22*R
870 FOR C=CL TO CU
880 SS=S+C
890 FOR I=0 TO 4
900 CD=SS+F*I
910 D=PEEK(CD)
920 IF D=46 THEN DW(I)=CD:GOTO 970
930 IF TP=0 THEN TP=D:II=((TP-81)/6+1)
940 IFTP<>D THEN V=0:TP=0:GOTO 1140
950 V=V+5-ABS(I-2)
960 DW(I)=0
```

```
970 NEXT I
980 IF V=0 THEN 1140
990 IF V<VMAX(II) THENV=0:GOTO 1060
1000 IF V=VMAX(II) THEN 1020
1010 NF(II)=0:VMAX(II)=V
1020 NF(II)=NF(II)+1
1030 FOR J=0 TO 4
1040 WD(II,NF(II),J)=DW(J)
1050 NEXT J
1060 FOR J=0 TO 4
1070 IFDW(J)=0 THEN1120
1080 DD=DW(J)-7796
1090 R1=INT(DD/22)
1100 C1=DD-22*R1
1110 BD(R1,C1)=BD(R1,C1)+1
1120 NEXT J
1130 V=0:TP=0
1140 NEXT C,R
1150 RETURN
1160 FOR R=1 TO 10
1170 FOR C=1 TO 10
1180 BD(R,C)=0
1190 NF(II)=NF(II)+1
1200 NEXT C,R
1210 RETURN
1220 PMAX=0:NT=0
1230 FOR I=1 TO NF(A)
1240 FOR J=0 TO 4
1250 P=WD(A,I,J)
1260 IF P=0 THEN 1340
1270 R1=INT((P-7796)/22)
1280 C1=(P-7796)-22*R1
1290 IF BD(R1,C1)<PMAX THEN 1340
1300 IF BD(R1,C1)=PMAX THEN 1320
1310 NT=0:PMAX=BD(R1,C1)
1320 NT=NT+1
1330 DB(NT)=P
1340 NEXT J
1350 IF A=2 AND PMAX=1 THEN 1390
1360 PN=INT(RND(1)*NT)+1
1370 POKE DB(PN),81:POKE DB(PN)+30720,5
1380 RETURN
1390 GOSUB 520
1400 RETURN
1410 FOR J=0 TO 4
1420 IF WD(A,1,J)=0 THEN 1440
1430 P=WD(A,1,J)
1440 NEXT J
1450 POKE P,81:POKE P+30720,5
1460 RETURN
1464 REM
```

```
1465 REM DISPLAY BOARD
1466 REM
1470 PRINT"▚▚▚▚"
1480 PRINTTAB(7);"ABCDEFGHIJ"
1490 PRINTTAB(6);"▛▒▒▒▒▒▒▒▒▒▒▒▜"
1500 NN=4
1510 FOR I=1 TO 10
1520 IF I>9 THEN NN=3
1530 PRINTTAB(NN);STR$(I)+"▛▒◣..........▛▒▜"
1540 NEXT I
1550 PRINTTAB(6);"▛▒▒▒▒▒▒▒▒▒▒▒▒▜";
1560 WIN =0
1570 RETURN
1580 T1=TI
1590 IF TI-T1<PA THEN 1590
1600 RETURN
READY.
```

SUPERMIND

DESCRIPTION

This game is modelled on the popular Parker Bros. board game of Mastermind. The program allows the user to guess at its random combination of colours and will reply with a white marker if the colour is in the combination but not in that position and with a black marker if it is in the correct position. The function keys are used to choose the row, the keys 3-8 to choose the colour and the return key to enter the guess. It has nice use of colours and is easy to follow.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

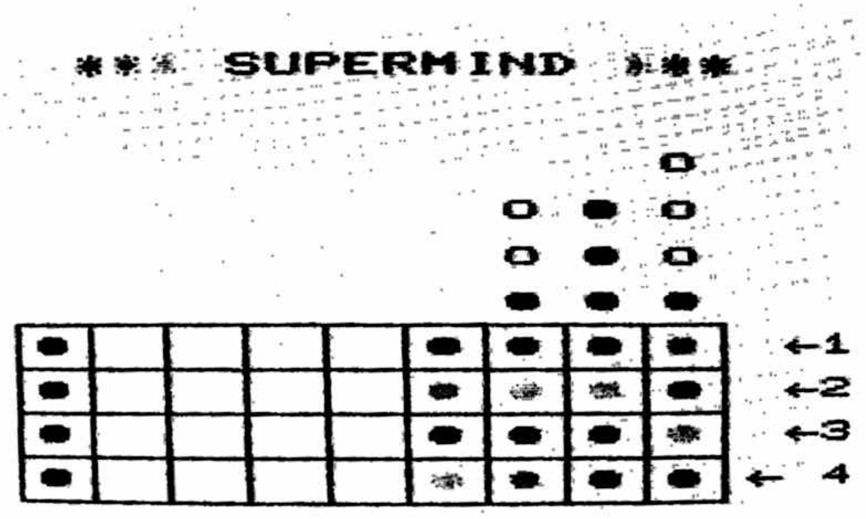
After typing run, three pages of instructions will appear separated by 'press any key' when the instructions have been read, pressing any key starts the game. Using the function keys, the row on which the colour is to be placed may be set. The computer will not accept any colours until the return key has been pressed, and only then, provided that there are colours in all four of the positions on that column. If the correct combination has been guessed, or all 14 guesses are used up, the computer will display its secret and ask if another go is required.

PROGRAM STRUCTURE

The following lines may be of interest.

50-90 Input for instructions.
100-280 Display the board on the screen.
290-350 Choose four random colours, all different.
410-450 Set white markers in every position.
460-500 Input and check that the key pressed is acceptable.
510 570 Display pointers so that all pointers are to the right except the pointer on the row chosen.

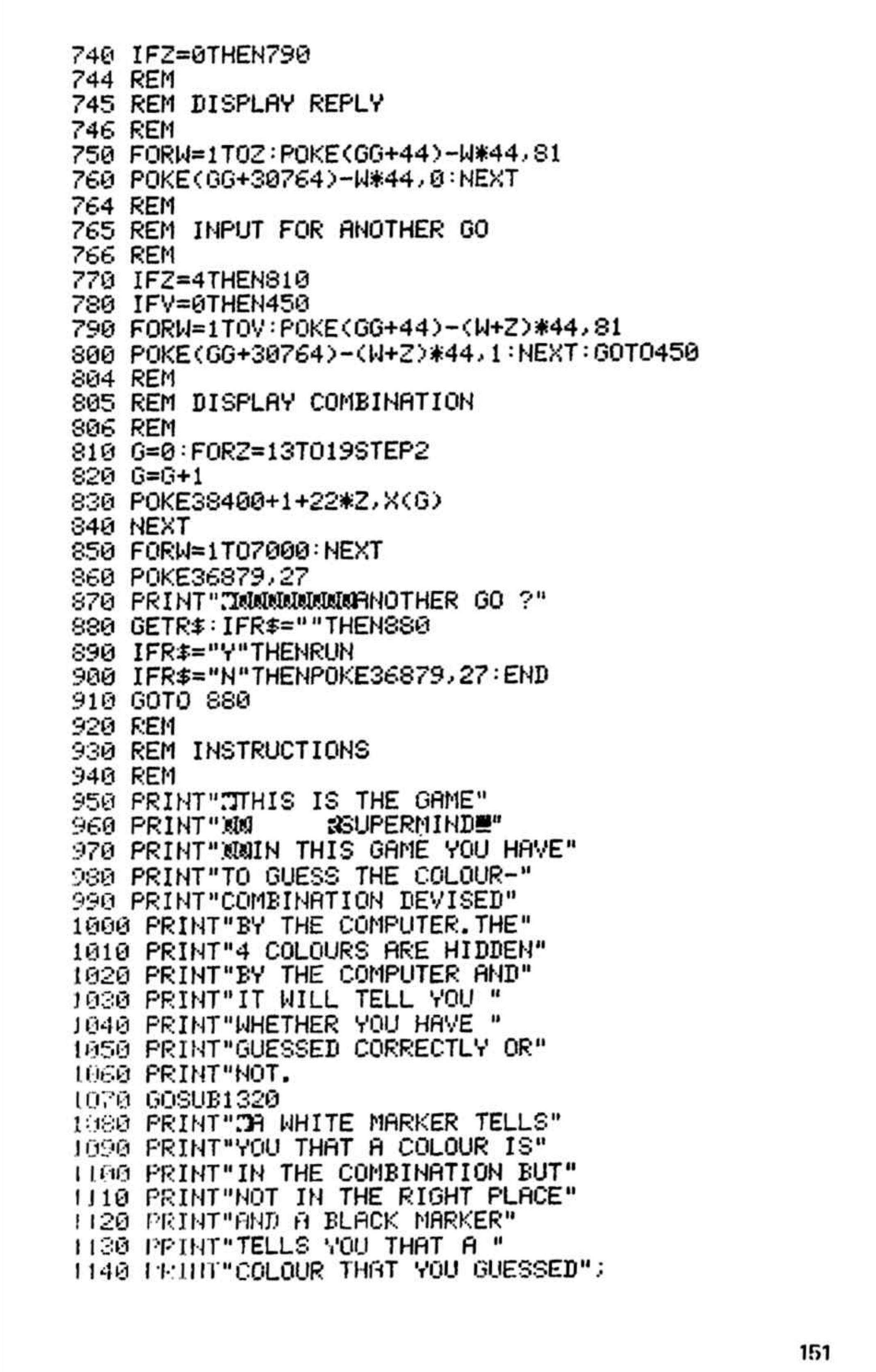
580-630 Put chosen color in the chosen position.
640-660 Failed to guess the combination.
680-690 Check for colours in the correct position.
710-740 Check for correct colours but in the wrong position.
750-760 Display the last two routines above the correponding columns.
810-860 Display the combination.
870-910 Input for another go.
950-1310 Instructions.
1320-1340 Hit any key input.



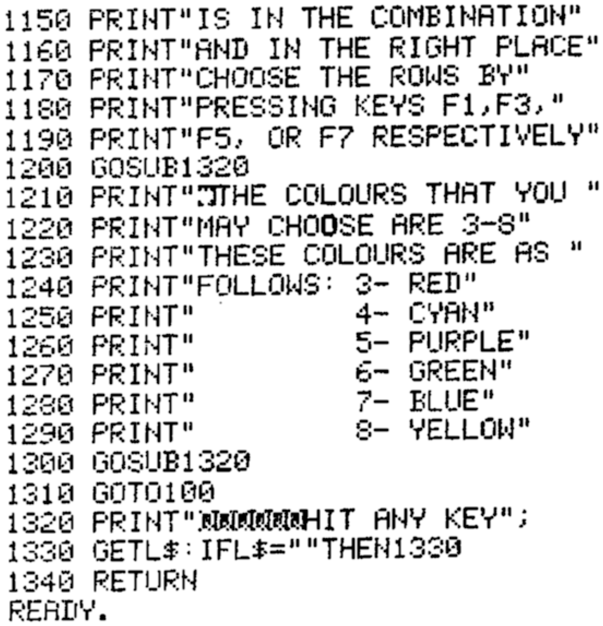
*** SUPERMIND ***

```
10 REM SUPERMIND
20 REM ****************************************
30 REM
40 ZZ=0
50 PRINT"■■■■■■■■INSTRUCTIONS(Y/N)?"
60 GETL$:IFL$=""THEN60
70 IFL$="Y"THEN920
80 IFL$="N"THEN100
90 GOTO60
94 REM
95 REM SET UP BOARD
96 REM
100 PRINT"■■■■*■*■* ■SUPERMIND ■*■*■*"
110 POKE36879,140
120 PRINT"■■■■■■■■■■■ ":DIMX(4),D(4)
130 R=7699
140 PRINT"┌─┬─┬─┬─┬─┬─┬─┬─┬─┐"
150 FORM=1TO3
160 PRINT"│";:FORW=1TO9
170 PRINT"■●■│";:NEXT:PRINT" ←"
180 PRINT"├";:FORW=1TO8
190 PRINT"─┼";:NEXT:PRINT"─┤"
200 NEXTM
210 PRINT"│";:FORW=1TO9
220 PRINT"■●■│";:NEXT:PRINT" ←"
230 PRINT"└─┴─┴─┴─┴─┴─┴─┴─┴─┘"
240 FORA=13TO19STEP2
250 POKE7701+22*A,49+B
260 POKE38421+22*A,4
270 B=B+1
280 NEXT
284 REM
285 REM CHOOSE RANDOM COLOURS,ALL DIFFERENT
286 REM
290 X(1)=INT(RND(1)*6)+2
300 X(2)=INT(RND(1)*6)+2
310 IFX(2)=X(1)THEN300
320 X(3)=INT(RND(1)*6)+2
330 IFX(3)=X(1)ORX(3)=X(2)THEN320
340 X(4)=INT(RND(1)*6)+2
350 IFX(4)=X(1)ORX(4)=X(2)ORX(4)=X(3)THEN340
360 G=0
370 FORZ=13TO19STEP2
380 G=G+1
390 POKE38400+1+22*Z,0:POKE7681+22*Z,102
400 NEXT
404 REM
405 REM WHITE "●'S" IN EVERY POSITION
406 REM
410 FORY=13TO19STEP2
420 POKE7680+1+22*Y,81
```

```
430 NEXTY
440 FORW=1TO250:NEXT:PRINT"S"
450 I=44:GG=7939-ZZ
454 REM
455 REM INPUT + CHECK FOR ILLEGAL KEY PRESS
460 GETQ$:IFQ$=""THEN450
470 IFASC(Q$)=13THEN630
480 IFASC(Q$)>132ANDASC(Q$)<137THEN510
490 IFASC(Q$)>50ANDASC(Q$)<57THEN580
500 GOTO460
504 REM
505 REM PUSH POINTERS OFF ROW
506 REM
510 FORS=7985TO8117STEP44
520 POKES+1,31
530 POKES,32
540 NEXT
544 REM
545 REM PUSH POINTER TO ROW
546 REM
550 L=2*ASC(Q$)-253
560 POKER+L*22,31:POKER+1+L*22,32
570 GOTO450
574 REM
575 REM PUT COLOUR IN POSITION
576 REM
580 F=VAL(Q$)-1
590 IFL=0THENL=13
600 POKE38417-ZZ+L*22,F
610 D((L-11)/2)=F
620 GOTO450
630 IFZZ<>14THEN650
634 REM
635 REM FILLED ALL 14 COLUMNS
636 REM
640 PRINT"YOU HAVE NOT SUCCEEDED":GOTO810
650 ZZ=ZZ+2:FORW=1TO4
660 IFD(W)=0THENZZ=ZZ-2:GOTO450
670 NEXT
674 REM
675 REM CHECK FOR BLACK REPLY
676 REM
680 Z=0:V=0
690 FORT=1TO4:IFX(T)=D(T)THENZ=Z+1
700 NEXT
704 REM
705 REM CHECK FOR WHITE REPLY
706 REM
710 FORE=1TO4:FORH=1TO4:IFX(E)=D(H)THENV=V+1
720 NEXT:NEXT:V=V-Z
730 FORW=1TO4:D(W)=0:NEXT
```



```
740 IFZ=0THEN790
744 REM
745 REM DISPLAY REPLY
746 REM
750 FORW=1TOZ:POKE(GG+44)-W*44,81
760 POKE(GG+30764)-W*44,0:NEXT
764 REM
765 REM INPUT FOR ANOTHER GO
766 REM
770 IFZ=4THEN810
780 IFV=0THEN450
790 FORW=1TOV:POKE(GG+44)-(W+Z)*44,81
800 POKE(GG+30764)-(W+Z)*44,1:NEXT:GOTO450
804 REM
805 REM DISPLAY COMBINATION
806 REM
810 G=0:FORZ=13TO19STEP2
820 G=G+1
830 POKE38400+1+22*Z,X(G)
840 NEXT
850 FORW=1TO7000:NEXT
860 POKE36879,27
870 PRINT"♥▒▒▒▒▒▒▒▒ANOTHER GO ?"
880 GETR$:IFR$=""THEN880
890 IFR$="Y"THENRUN
900 IFR$="N"THENPOKE36879,27:END
910 GOTO 880
920 REM
930 REM INSTRUCTIONS
940 REM
950 PRINT"♥THIS IS THE GAME"
960 PRINT"▒▒     ▣SUPERMIND■"
970 PRINT"▒▒IN THIS GAME YOU HAVE"
980 PRINT"TO GUESS THE COLOUR-"
990 PRINT"COMBINATION DEVISED"
1000 PRINT"BY THE COMPUTER.THE"
1010 PRINT"4 COLOURS ARE HIDDEN"
1020 PRINT"BY THE COMPUTER AND"
1030 PRINT"IT WILL TELL YOU "
1040 PRINT"WHETHER YOU HAVE "
1050 PRINT"GUESSED CORRECTLY OR"
1060 PRINT"NOT.
1070 GOSUB1320
1080 PRINT"♥A WHITE MARKER TELLS"
1090 PRINT"YOU THAT A COLOUR IS"
1100 PRINT"IN THE COMBINATION BUT"
1110 PRINT"NOT IN THE RIGHT PLACE"
1120 PRINT"AND A BLACK MARKER"
1130 PRINT"TELLS YOU THAT A "
1140 PRINT"COLOUR THAT YOU GUESSED";
```



```
1150 PRINT"IS IN THE COMBINATION"
1160 PRINT"AND IN THE RIGHT PLACE"
1170 PRINT"CHOOSE THE ROWS BY"
1180 PRINT"PRESSING KEYS F1,F3,"
1190 PRINT"F5, OR F7 RESPECTIVELY"
1200 GOSUB1320
1210 PRINT"■THE COLOURS THAT YOU "
1220 PRINT"MAY CHOOSE ARE 3-8"
1230 PRINT"THESE COLOURS ARE AS "
1240 PRINT"FOLLOWS: 3- RED"
1250 PRINT"         4- CYAN"
1260 PRINT"         5- PURPLE"
1270 PRINT"         6- GREEN"
1280 PRINT"         7- BLUE"
1290 PRINT"         8- YELLOW"
1300 GOSUB1320
1310 GOTO100
1320 PRINT"QQQQQQHIT ANY KEY";
1330 GETL$:IFL$=""THEN1330
1340 RETURN
READY.
```

CONQUEST

DESCRIPTION

In Conquest the aim of the game is to vanquish the knights protecting the castle. The display consists of a green background and a believable castle with a heart—representing the knights moving to various positions within the castle.
Frustratedly, the attacker—a cannon—seeking to demoralise the defenders and held at bay by a moat, fires on the user's command.

EQUIPMENT REQUIRED

Basic VIC-20.

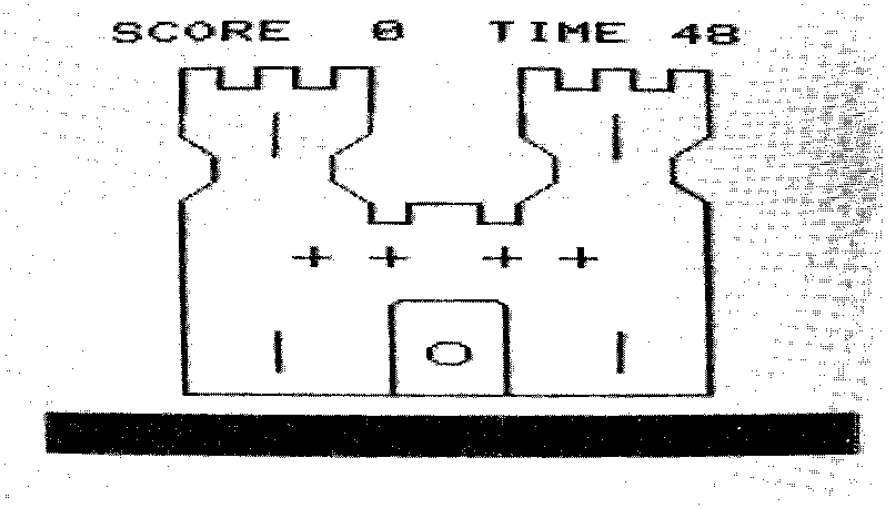
RUNNING THE PROGRAM

Simply load, type RUN and follow the instructions. The operating keys are: A = left, D = right, * = fire.
The player must also remember that 9 = easiest and 1 = hardest in the levels of difficulty. You have only 60 seconds to complete the game.

PROGRAM STRUCTURE

The lines of interest in this program are as follows:

30-90	Print instructions.
100-190	Input difficulty level.
200-370	Display castle.
380-840	Play the game.
850-970	Finished, input for another game.
1110-1190	Play tune at beginning of game.
1200-1240	Data for music.
1250-1300	Input from keyboard.
1320-1380	Run out of time.
1390-1450	Display commands.



SCORE 0 TIME 48

```
1 REM CONQUEST
2 REM ****************************************
3 REM
10 FORI=826TO835:POKEI,255:NEXT
20 POKE36879,93:POKE36869,242
24 REM
25 REM INSTRUCTIONS
26 REM
30 PRINT"▒CONQUEST":PRINT"────────"
40 PRINT"▒YOUR AIM IS"
50 PRINT"▒TO KILL ALL THE"
60 PRINT"▒NASTY KNIGHTS "
70 PRINT"▒WHO ARE HIDING "
80 PRINT"▒AROUND THE CASTLE."
90 GOSUB1380
94 REM
95 REM INPUT DIFFICULTY LEVEL
96 REM
100 PRINT"▒INPUT DIFFICULTY":PRINT"▒▒(1-9)"
110 GETDH$:IFDH$=""THEN110
120 DF=VAL(DH$):IFDF<1ORDF>9THEN110
130 DEFFNB(X)=INT(RND(4)*150+DF*10)
140 K=197:Z=PEEK(K)
150 POKE825,DF
160 DEFFNA(X)=INT(RND(8)*90+TV)
170 POKE36869,240:TL=7742:TD=7724
180 Y1=106:Y2=113:Y3=117:BL=8142
190 KN=83:AR=30:TV=7758
194 REM
195 REM DISPLAY CASTLE
196 REM
200 PRINT"▒▒▒▒▒▒▒▒⊓⊔⊓⊔⊓    ⊓⊔⊓⊔⊓ "
210 PRINT"▒▒▒▒    |   |    | "
220 PRINT"▒▒▒▒  | |   |  | | "
230 PRINT"▒▒▒▒╲ | ╱    ╲ | ╱ "
240 PRINT"▒▒▒▒|   |    |   | "
250 PRINT"▒▒▒▒╱   ╲    ╱   ╲ "
260 PRINT"▒▒▒▒     ⊔⊓⊔     | "
270 PRINT"▒▒▒▒             | "
280 PRINT"▒▒▒▒   + +  + +  | "
290 PRINT"▒▒▒▒             | "
300 PRINT"▒▒▒▒     ╭──╮    | "
310 PRINT"▒▒▒▒     |  |    | "
320 PRINT"▒▒▒▒  |  |◠ |  | | "
330 PRINT"▒▒▒▒  |  |◡ |  | | "
340 PRINT"▒▒▒▒     |  |    | "
350 PRINT"▒▒▒▒──────────────  "
360 PRINT"▒▒                     ";
370 PRINT"                      ▒▒"
374 REM
375 REM PLAY THE GAME
```

```
376 REM
380 POKEBL,Y1:POKEBL+1,Y2:POKEBL+2,Y3
390 PRINT"▒▒▒SCORE "S:TI$="000000"
400 PO=FNA(T)
410 MM=PEEK(PO):IFDF=0THENDF=PEEK(826)
420 FORI=1TOINT(RND(8)*DF*10)
430 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒TIME "RIGHT$(TI$,2)
440 IFVAL(TI$)>100THEN1320
450 GOSUB1250
460 POKEPO,KN:POKEPO+30720,2
470 POKEK,0:NEXT
480 POKEPO,MM
490 GOTO400
500 POKE36879,27:PRINT"▒"
510 IFBL+1>8161THEN560
520 POKEBL,32:POKEBL+1,32:POKEBL+2,32
530 BL=BL+1:IFBL>8161THEN560
540 POKEBL,Y1:POKEBL+1,Y2:POKEBL+2,Y3
550 RETURN
560 T2=T2+0.5:RETURN
570 REM LEFT
580 IFBL-1<8142THENRETURN
590 POKEBL,32:POKEBL+1,32:POKEBL+2,32
600 BL=BL-1:IFBL<8142THEN560
610 POKEBL,Y1:POKEBL+1,Y2:POKEBL+2,Y3
620 T2=T2+.5:RETURN
630 POKE36878,15:T3$=TI$
640 FORW=BL+1TO7724STEP-22
650 XX=PEEK(W):YY=PEEK(W+30720)
660 IFPEEK(W)=83THEN710
670 POKEW,AR:POKEW+30720,0:POKEK,0
680 FORU=255TO195STEP-10:POKE36876,U:NEXT
690 POKEW,XX:POKEW+30720,YY:NEXT
700 POKE36876,0:TI$=T3$:RETURN
710 POKE36876,0:POKE36877,240
720 FORDE=1TO20:POKE36877,0:POKEW+30720,4
730 POKEW+30720,1:POKEK,0:POKE36877,199:NEXT
740 TI$=T3$:POKE36877,0
750 POKEW+30720,YY:POKEW,MM:POKEK,0
760 S=S+1
770 PRINT"▒▒▒SCORE "S:T2=VAL(RIGHT$(TI$,2))
780 IF S>5 THEN 800
790 ONSGOTO820,980,820,980,820
800 S=S-5
810 ONSGOTO980,820,980,820,840
820 IFT2>100THEN1320
830 POKEK,0:GOTO400
840 GOSUB1110
844 REM
845 REM CONGRATULATIONS, FINISHED
846 REM
```

```
850 PRINT"■■YOU HAVE VANQUISHED"
860 PRINT"■■THE EVIL KNIGHTS-";
870 PRINT"■■■■■■■■WELL DONE!"
880 PRINT"■■YOUR TIME="T2
890 PRINT"■■■(SKJLL LEVEL"PEEK(825)")"
900 DF=PEEK(825):T1=PEEK(825+DF)
910 IFT2<T1THENT1=T2
920 PRINT"■■■LOWEST TIME="T1
930 POKE825+DF,T1
940 PRINT"■■■ANOTHER QUEST?"
950 GETA$:IFA$<>"Y"ANDA$<>"N"THEN950
960 IFA$="N"THEN1450
970 CLR:GOTO100
980 POKE8185,S:POKE38905,5
990 IFVAL(TI$)>100THEN1320
1000 BL=BL-8142:POKE8183,BL:POKE38903,5
1010 CLR
1020 DEFFNA(DF)=INT(RND(8)*90+TV)
1030 DEFFNB(DF)=INT(RND(4)*DF*10)
1040 K=197:Z=PEEK(K)
1050 TL=7742:TD=7724:Y1=106:Y2=113:Y3=117
1060 KN=83:AR=30:BL=PEEK(8183)+8142:TV=7758
1070 T2=PEEK(8184):S=PEEK(8185)
1080 PRINT"■■■SCORE "S
1090 DF=PEEK(825)
1100 POKE36869,240:TU=2:GOTO400
1104 REM
1105 REM PLAY TUNE AT BEGINNING OF GAME
1106 REM
1110 RESTORE:S2=36876:V=36878
1120 POKEV,15
1130 READP
1140 IFP=-1THENRETURN
1150 POKES2,P
1160 READD
1170 FORDE=1TOD:NEXTDE
1180 POKES2,0
1190 GOTO1130
1194 REM
1195 REM DATA FOR MUSIC
1196 REM
1200 DATA217,200,213,200,223,400
1210 DATA227,100,234,100,230,400
1220 DATA227,100,234,100,230,400
1230 DATA223,200,227,200,217,200
1240 DATA-1
1244 REM
1245 REM INPUT FROM KEYBOARD
1246 REM
1250 FORT=1TO1
1260 Z=PEEK(K)
```

```
1270 IFZ=14THENGOSUB630
1280 IFZ=18THENGOSUB510
1290 IFZ=17THENGOSUB570
1300 NEXT
1310 RETURN
1314 REM
1315 REM RUN OUT OF TIME
1316 REM
1320 FORDE=1TO1500:NEXT
1330 PRINT"▒▒▒OUT OF TIME!"
1340 PRINT"▒▒▒ANOTHER QUEST?"
1350 GETA$:IFA$<>"Y"ANDA$<>"N"THEN1350
1360 IFA$="N"THEN1450
1370 CLR:GOTO100
1380 IFTU>1THENRETURN
1384 REM
1385 REM DISPLAY COMMANDS
1386 REM
1390 PRINT"▒▒▒COMMANDS:":PRINT"──────────"
1400 PRINT"A-CANNON LEFT"
1410 PRINT"D-CANNON RIGHT"
1420 PRINT"*-CANNON FIRE"
1430 FORI=1TO3:GOSUB1110:NEXT
1440 RETURN
1450 POKE36879,27:PRINT"▒▒"
READY.
```

HI-RES AID

DESCRIPTION

This program is a useful aid to producing graphics characters. With this program the user may design any characters he wishes (subject to the limitations of the grid). Using the data he will be provided with, the user has a ready-made data statement for his own program.

EQUIPMENT REQUIRED

Basic VIC-20 with 3K expansion.

RUNNING THE PROGRAM

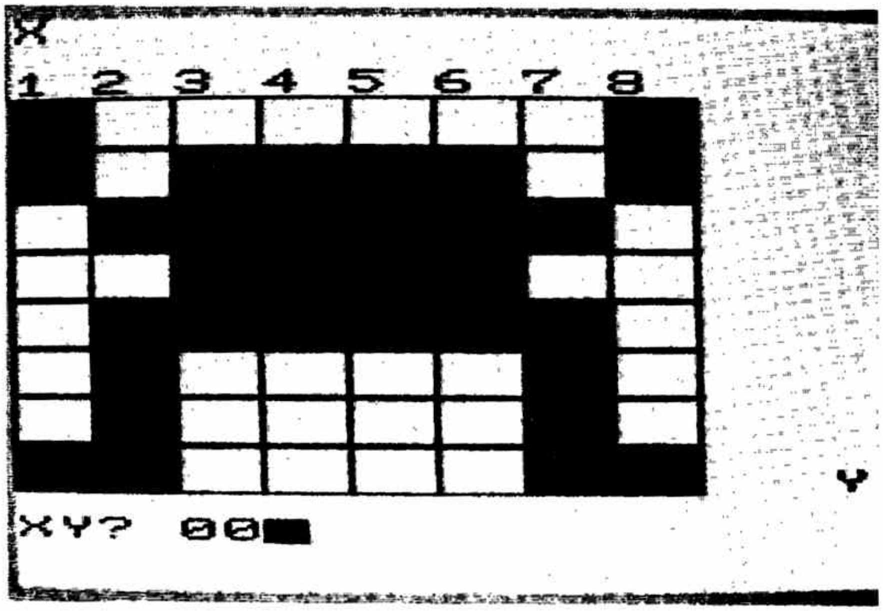
This program will run without expansion if all the REM statements are removed.
After loading and typing RUN, the user will initially see the words 'Moving chr generator' appear at the top of the screen. After that the screen will clear and an 8 × 8 grid is drawn out. The user then inputs the X and Y co-ordinates of the block he wishes to fill, without separating by commas. The block is then coloured in and the user carries on drawing in his character. When the user wishes to see what his design looks like he must type in co-ordinates 00; the screen clears to display hundreds of little aliens (or whatever).
Hitting any key then prints out the values to be entered as data for that particular character; then the user may continue to design another character.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows:

80-170	Print out for grid.
190-200	Input co-ordinates.
220-230	Calculate X and Y from previous routine.
250	Check if in range.
270	If 00 entered, end of design.
290	Bleep to show input accepted.
310	Calculate top left hand corner of block.

330	Check for double entry.
350	Print out blocks.
370-380	Calculate value of block.
400	Turn bleep off.
420	Return for next input.
450-480	Store the character.
500-510	Change the registers.
530-600	Set screen to black and display hi-res character on the screen.
620-710	Restore the registers and display the values of the character on the screen.
730-770	Move the character generator.



```
XY? 00
```

```
10 REM****HI-RES AID****
20 REM******************
30 PRINT"]MOVING CHR GENERATOR"
40 GOSUB730
50 PRINT:PRINT
60 REM**SET VARIABLES**
70 Q=160:S3=36876:POKES3+2,15
80 REM**PRINT OUT GRID**
90 PRINT"SX":PRINT
100 PRINT"1 2 3 4 5 6 7 8"
110 FORI=1TO8
120 PRINT"┌─┬─┬─┬─┬─┬─┬─┬─┐ "
130 PRINT"└─┴─┴─┴─┴─┴─┴─┴─┘ "
140 NEXTI
150 PRINT"]",TAB(19)"Y"
160 Z=7700
170 PRINT
180 REM**ENTER CO-ORDINATES**
190 INPUT"XY";A$
200 IFLEN(A$)<>2THEN420
210 REM**PICK OFF X,Y**
220 X=VAL(LEFT$(A$,1))
230 Y=VAL(RIGHT$(A$,1))
240 REM**CHECK TO SEE IF IN RANGE**
250 IFX>8ORY>8THEN420
260 REM**END OF DESIGN**
270 IFX=0ANDY=0THEN430
280 REM**BLEEP ON**
290 POKES3,200
300 REM**CALCULATE TOP LEFT HAND CORNER OF BLOCK**
310 P=Z+X*2+(Y*44)
320 REM**CHECK FOR DOUBLE ENTRY**
330 IFPEEK(P)=QTHEN420
340 REM**PRINT OUT BLOCKS**
350 POKEP,Q:POKEP+1,Q:POKEP+22,Q:POKEP+23,Q
360 REM**CALCULATE VALUE OF BLOCK**
370 U=ABS(X-8)
380 A(Y)=A(Y)+(2↑U)
390 REM**BLEEP OFF**
400 POKES3,0
410 REM**BACK TO INPUT**
420 PRINT"T]":GOTO190
430 PRINT"]"
440 REM**STORE PATTERN'S VALUE IN NEW GENERATOR**
450 POKES3,0
460 FORI=1TO8
470 POKE6143+I,A(I)
480 NEXTI
490 REM**CHANGE REGISTERS**
500 POKE36869,253
510 POKE36866,PEEK(36866)OR128
```

```
520 REM**CHANGE SCREEN COL. TO BLACK**
530 POKE36879,8
540 PRINT"♥"
550 REM**PRINT HI-RES CHR. ON SCREEN**
560 FORI=7702TO8185STEP4
570 POKEI,128
580 NEXTI
590 PRINT"■■      HIT ANY KEY       "
600 GETA$:IFA$=""THEN600
610 REM**RESTORE REGISTERS**
620 POKE36869,240
630 POKE36866,150
640 PRINT"♥■"
650 POKE36879,27
660 PRINT"THESE ARE THE VALUES"
670 PRINT"■TO ENTER AS DATA":PRINT:PRINT
680 FORI=1TO8
690 PRINTA(I)
700 NEXTI
710 END
720 REM**MOVE CHR. GENERATOR**
730 FORI=0TO1024
740 POKE5120+I,PEEK(32768+I)
750 NEXTI
760 PRINT"♥"
770 RETURN
READY.
```

TINYMON

DESCRIPTION

One of the functions missing on the standard Vic is the ability to enter and explore the inner workings of the machine. The program Tinymon allows access to the Vic's internal memory, on either the standard Vic computer or one with any size expansion added, and provides six commands for manipulation and working of that memory.
Once the program has been entered and RUN (see next section), the following six commands are available :

1) All monitor instructions start with the prompt '.'. To display memory, enter :
.M 123E 12A7 (for example)
And the Vic will respond with something like :
.123E AA 39 17 FF 04 etc.

To change memory, display the required range of memory first, and then just type over the locations to be changed. Don't forget to press RETURN first!

2) To display registers, enter :
.R
and the Vic will display (for example) :

```
   PC       SR       AC       XR       YR       SP
 .;1234     30       00       22       33       F8
```

These are the internal registers of the 6502 processor. To change them, just type over the ones you wish to alter, and then press RETURN.

3) To return to Basic, enter :
.X

4) To run a machine language program, enter :
.G 1400 (for example)
and the program will execute from that address.

5) To save specific memory locations, enter :

```
.S 'NAME',01,1250,13A0
```

Where 01 is the device number (cassette deck in this instance), 1250 is the start address of the memory block you wish to save, and 13A0 is the end address PLUS ONE !! Always add one to the end of the memory block you want to save, owing to a peculiarity in the Vic's operating system.

6) To load programs into memory, enter :

```
.L 'NAME',01
```

Programs always load to where they were saved from.

ENTERING AND RUNNING THE PROGRAM

To enter the program, a monitor is required, you therefore need access to a Commodore PET computer before going any further. The reason for this? This program will give your Vic a monitor, but to enter the program you need a monitor.

Using a PET, enter its monitor by entering SYS 1024 (RETURN), and display the first block of numbers and letters by entering .M 0400 0438. This will bring up a display of numbers, and you have to type in the ones shown in the listing at the end. Don't forget to press RETURN at the end of each line.

When you've finished the first block, enter .M 0440 0478 and type in the second group of numbers.

Carry on in this manner until the final block i.e. .M 0800 0818.

Before doing anything else, save this by typing .S 'TINYMON',01,0400,0819, assuming you are using a cassette deck.

With something of this length it is more than likely that you'll have made a mistake or two along the way, the following 'checksum' has been provided to a) see if you've typed it all in correctly, and b) find out where the errors lie.

Type in the following short program, IN DIRECT MODE ONLY, if the number that is printed up at the end equals 117952, CONGRATULATIONS!! You should have a working copy of Tinymon.

```
A=0:FORI=1024 TO 2079 : A=A+PEEK(I):NEXT:PRINTA
```

Take your tape over to your Vic, load it in in the normal way, and RUN it. You now have access to the Vic monitor, which can be re-accessed by a SYS 13 call. The program itself occupies about 800 bytes, that should leave plenty of space for you to explore the inner workings of the Vic.

If the number displayed is not 117952, then we have problems. Take the program mentioned earlier :

```
A=0:FORI=X TO Y : A=A+PEEK(I):NEXT:PRINTA
```

The following table shows the values of X and Y to be entered to check each block in turn, and the total that should be displayed. If your total differs from the one printed here, you've found a block with a mistake (or two) in it. Re-enter the monitor, check your listing against the original, and correct any mistakes you find. After checking each block, re-save the whole program as described earlier.

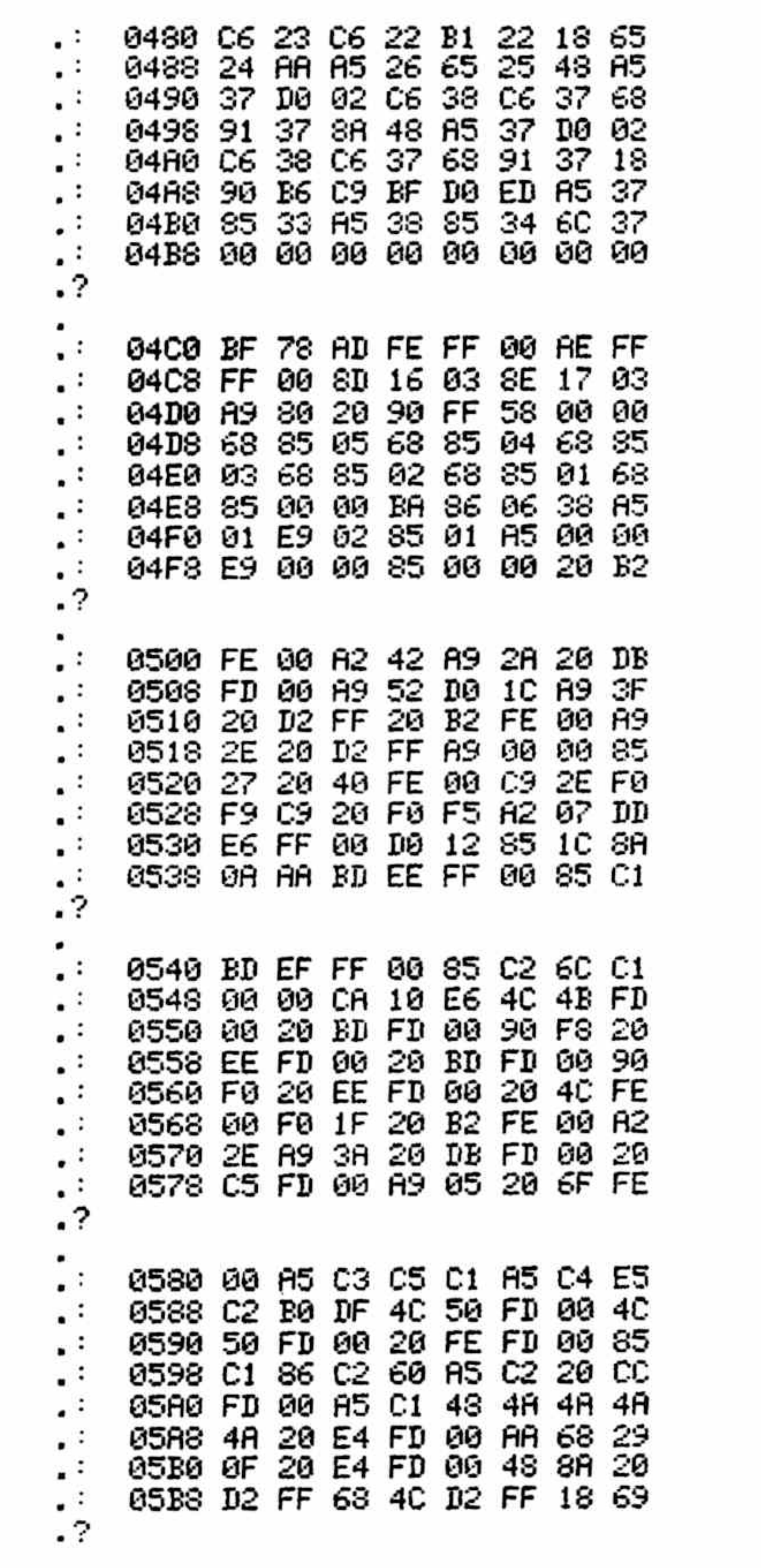
BLOCK NO.	X	Y	TOTAL
1	1024	1087	3841
2	1088	1151	6376
3	1152	1215	6272
4	1216	1279	5922
5	1280	1343	8322
6	1344	1407	8022
7	1408	1471	8451
8	1472	1535	7098
9	1536	1599	5853
10	1600	1663	7007
11	1664	1727	8287
12	1728	1791	7240
13	1792	1855	6786
14	1856	1919	9362
15	1920	1983	9475
16	1984	2047	6551
17	2048	2079	3130

The effort involved in typing this in should be amply rewarded by the results obtained.

```
10 PRINT"■VMON 10 XP"
20 PRINT"■.R
30 PRINT" REGISTER DISPLAY
40 PRINT"■.M FFFF TTTT
50 PRINT" MEMORY DISPLAY
60 PRINT"■.X
70 PRINT" EXIT TO BASIC"
80 PRINT"■.G AAAA
90 PRINT" GOTO (EX) ADDRESS
100 PRINT"■.S"CHR$(34)"PPPP"CHR$(34)",01,FFFF,TTTT
110 PRINT" SAVE PROGRAM
120 PRINT"■.L"CHR$(34)"PPPF"CHR$(34)
130 PRINT" LOAD PROGRAM
READY.
100 PRINT"■■■■■■■ TINYMON
110 PRINT"■ JIM BUTTERFIELD"
120 SYS(PEEK(43)+256*PEEK(44)+078)
READY.


READY.

B*
     PC  IRQ  SR AC XR YR SP
.;  0005 E62E 30 00 5E 04 F8
.
.:  0400 00 18 04 64 00 99 22 93
.:  0408 11 11 12 1D 1D 1D 20 54
.:  0410 49 4E 59 4D 4F 4E 20 00
.:  0418 31 04 6E 00 99 22 11 20
.:  0420 4A 49 4D 20 42 55 54 54
.:  0428 45 52 46 49 45 4C 44 22
.:  0430 00 4C 04 78 00 9E 28 C2
.:  0438 28 34 33 29 AA 32 35 36
.?
.
.:  0440 AC C2 28 34 34 29 AA 30
.:  0448 37 38 29 00 00 00 EA EA
.:  0450 A5 2D 85 22 A5 2E 85 23
.:  0458 A5 37 85 24 A5 38 85 25
.:  0460 A0 00 A5 22 D0 02 C6 23
.:  0468 C6 22 B1 22 D0 3C A5 22
.:  0470 D0 02 C6 23 C6 22 B1 22
.:  0478 F0 21 85 26 A5 22 D0 02
.?
```



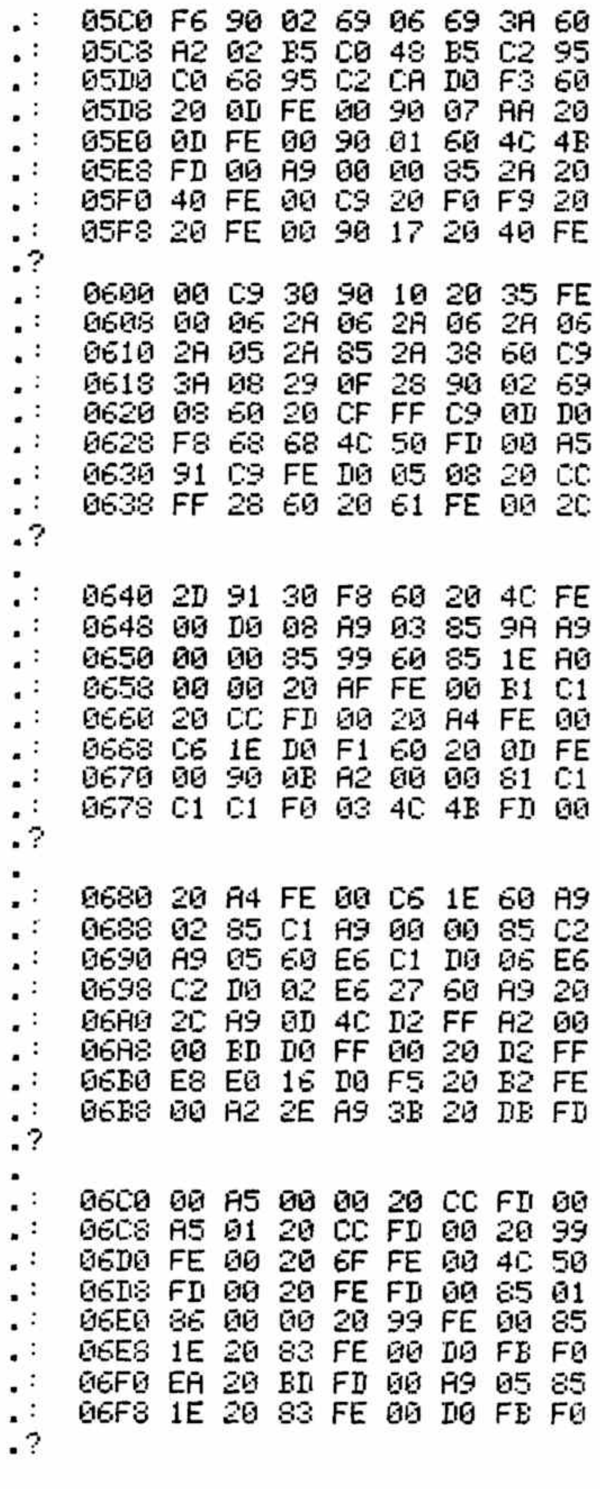
```
.:   0480 C6 23 C6 22 B1 22 18 65
.:   0488 24 AA A5 26 65 25 48 A5
.:   0490 37 D0 02 C6 38 C6 37 68
.:   0498 91 37 8A 48 A5 37 D0 02
.:   04A0 C6 38 C6 37 68 91 37 18
.:   04A8 90 B6 C9 BF D0 ED A5 37
.:   04B0 85 33 A5 38 85 34 6C 37
.:   04B8 00 00 00 00 00 00 00 00
.?

.
.:   04C0 BF 78 AD FE FF 00 AE FF
.:   04C8 FF 00 8D 16 03 8E 17 03
.:   04D0 A9 80 20 90 FF 58 00 00
.:   04D8 68 85 05 68 85 04 68 85
.:   04E0 03 68 85 02 68 85 01 68
.:   04E8 85 00 00 BA 86 06 38 A5
.:   04F0 01 E9 02 85 01 A5 00 00
.:   04F8 E9 00 00 85 00 00 20 B2
.?

.
.:   0500 FE 00 A2 42 A9 2A 20 DB
.:   0508 FD 00 A9 52 D0 1C A9 3F
.:   0510 20 D2 FF 20 B2 FE 00 A9
.:   0518 2E 20 D2 FF A9 00 00 85
.:   0520 27 20 40 FE 00 C9 2E F0
.:   0528 F9 C9 20 F0 F5 A2 07 DD
.:   0530 E6 FF 00 D0 12 85 1C 8A
.:   0538 0A AA BD EE FF 00 85 C1
.?

.
.:   0540 BD EF FF 00 85 C2 6C C1
.:   0548 00 00 CA 10 E6 4C 4B FD
.:   0550 00 20 BD FD 00 90 F8 20
.:   0558 EE FD 00 20 BD FD 00 90
.:   0560 F0 20 EE FD 00 20 4C FE
.:   0568 00 F0 1F 20 B2 FE 00 A2
.:   0570 2E A9 3A 20 DB FD 00 20
.:   0578 C5 FD 00 A9 05 20 6F FE
.?

.
.:   0580 00 A5 C3 C5 C1 A5 C4 E5
.:   0588 C2 B0 DF 4C 50 FD 00 4C
.:   0590 50 FD 00 20 FE FD 00 85
.:   0598 C1 86 C2 60 A5 C2 20 CC
.:   05A0 FD 00 A5 C1 48 4A 4A 4A
.:   05A8 4A 20 E4 FD 00 AA 68 29
.:   05B0 0F 20 E4 FD 00 48 8A 20
.:   05B8 D2 FF 68 4C D2 FF 18 69
.?
```

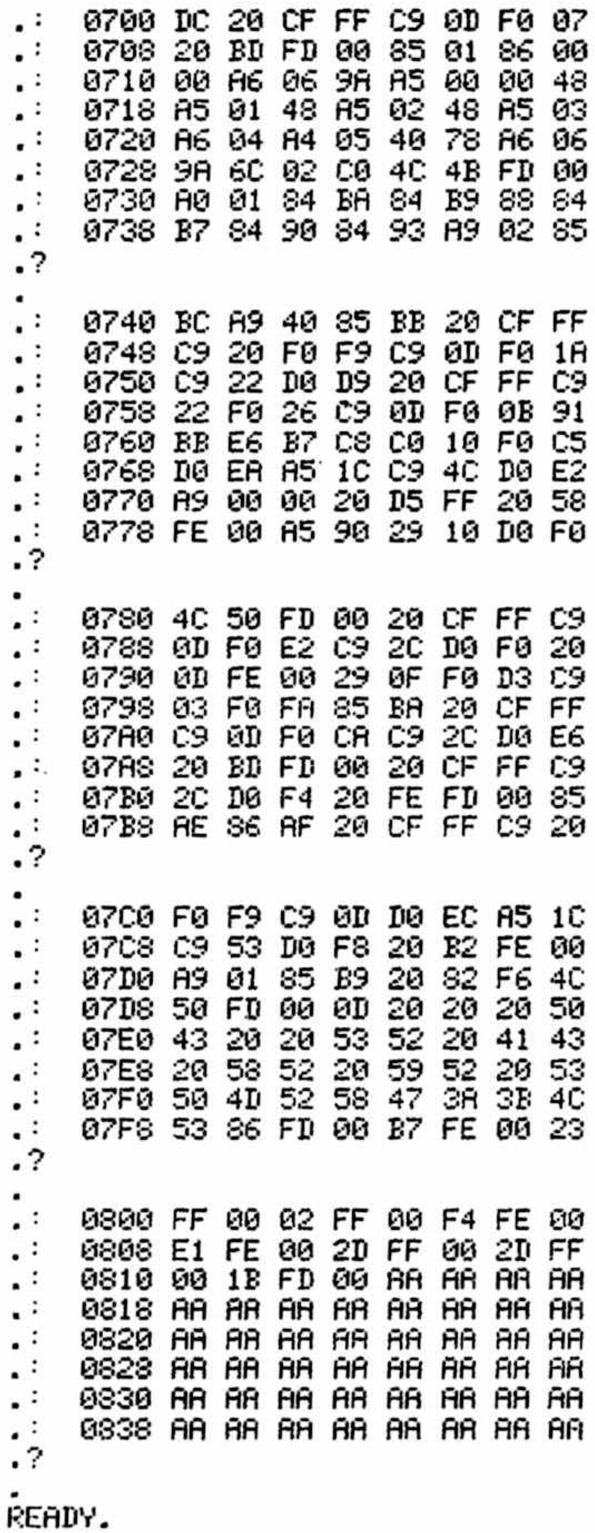


```
.:   05C0 F6 90 02 69 06 69 3A 60
.:   05C8 A2 02 B5 C0 48 B5 C2 95
.:   05D0 C0 68 95 C2 CA D0 F3 60
.:   05D8 20 0D FE 00 90 07 AA 20
.:   05E0 0D FE 00 90 01 60 4C 4B
.:   05E8 FD 00 A9 00 00 85 2A 20
.:   05F0 40 FE 00 C9 20 F0 F9 20
.:   05F8 20 FE 00 90 17 20 40 FE
.?
.:   0600 00 C9 30 90 10 20 35 FE
.:   0608 00 06 2A 06 2A 06 2A 06
.:   0610 2A 05 2A 85 2A 38 60 C9
.:   0618 3A 08 29 0F 28 90 02 69
.:   0620 08 60 20 CF FF C9 0D D0
.:   0628 F8 68 68 4C 50 FD 00 A5
.:   0630 91 C9 FE D0 05 08 20 CC
.:   0638 FF 28 60 20 61 FE 00 2C
.?

.
.:   0640 2D 91 30 F8 60 20 4C FE
.:   0648 00 D0 08 A9 03 85 9A A9
.:   0650 00 00 85 99 60 85 1E A0
.:   0658 00 00 20 AF FE 00 B1 C1
.:   0660 20 CC FD 00 20 A4 FE 00
.:   0668 C6 1E D0 F1 60 20 0D FE
.:   0670 00 90 0B A2 00 00 81 C1
.:   0678 C1 C1 F0 03 4C 4B FD 00
.?

.
.:   0680 20 A4 FE 00 C6 1E 60 A9
.:   0688 02 85 C1 A9 00 00 85 C2
.:   0690 A9 05 60 E6 C1 D0 06 E6
.:   0698 C2 D0 02 E6 27 60 A9 20
.:   06A0 2C A9 0D 4C D2 FF A2 00
.:   06A8 00 BD D0 FF 00 20 D2 FF
.:   06B0 E8 E0 16 D0 F5 20 B2 FE
.:   06B8 00 A2 2E A9 3B 20 DB FD
.?

.
.:   06C0 00 A5 00 00 20 CC FD 00
.:   06C8 A5 01 20 CC FD 00 20 99
.:   06D0 FE 00 20 6F FE 00 4C 50
.:   06D8 FD 00 20 FE FD 00 85 01
.:   06E0 86 00 00 20 99 FE 00 85
.:   06E8 1E 20 83 FE 00 D0 FB F0
.:   06F0 EA 20 BD FD 00 A9 05 85
.:   06F8 1E 20 83 FE 00 D0 FB F0
.?
```



```
.:   0700 DC 20 CF FF C9 0D F0 07
.:   0708 20 BD FD 00 85 01 86 00
.:   0710 00 A6 06 9A A5 00 00 48
.:   0718 A5 01 48 A5 02 48 A5 03
.:   0720 A6 04 A4 05 40 78 A6 06
.:   0728 9A 6C 02 C0 4C 4B FD 00
.:   0730 A0 01 84 BA 84 B9 88 84
.:   0738 B7 84 90 84 93 A9 02 85
.?
.
.:   0740 BC A9 40 85 BB 20 CF FF
.:   0748 C9 20 F0 F9 C9 0D F0 1A
.:   0750 C9 22 D0 D9 20 CF FF C9
.:   0758 22 F0 26 C9 0D F0 0B 91
.:   0760 BB E6 B7 C8 C0 10 F0 C5
.:   0768 D0 EA A5 1C C9 4C D0 E2
.:   0770 A9 00 00 20 D5 FF 20 58
.:   0778 FE 00 A5 90 29 10 D0 F0
.?
.
.:   0780 4C 50 FD 00 20 CF FF C9
.:   0788 0D F0 E2 C9 2C D0 F0 20
.:   0790 0D FE 00 29 0F F0 D3 C9
.:   0798 03 F0 FA 85 BA 20 CF FF
.:   07A0 C9 0D F0 CA C9 2C D0 E6
.:   07A8 20 BD FD 00 20 CF FF C9
.:   07B0 2C D0 F4 20 FE FD 00 85
.:   07B8 AE 86 AF 20 CF FF C9 20
.?
.
.:   07C0 F0 F9 C9 0D D0 EC A5 1C
.:   07C8 C9 53 D0 F8 20 B2 FE 00
.:   07D0 A9 01 85 B9 20 82 F6 4C
.:   07D8 50 FD 00 0D 20 20 20 50
.:   07E0 43 20 20 53 52 20 41 43
.:   07E8 20 58 52 20 59 52 20 53
.:   07F0 50 4D 52 58 47 3A 3B 4C
.:   07F8 53 86 FD 00 B7 FE 00 23
.?
.
.:   0800 FF 00 02 FF 00 F4 FE 00
.:   0808 E1 FE 00 2D FF 00 2D FF
.:   0810 00 1B FD 00 AA AA AA AA
.:   0818 AA AA AA AA AA AA AA AA
.:   0820 AA AA AA AA AA AA AA AA
.:   0828 AA AA AA AA AA AA AA AA
.:   0830 AA AA AA AA AA AA AA AA
.:   0838 AA AA AA AA AA AA AA AA
.?
.
READY.
```

RACER

DESCRIPTION

This game has two programs making a whole. The object of the game is to steer the racing car through the course against such hazards as oncoming racing cars and oil slicks. The pace of the game is very fast to begin with and it becomes progressively harder. To score points the racer must pass through certain points on the course, a tricky procedure, as the oil slicks cause the racer to slide one point either side. The game is over when either the oncoming cars are hit or the racer is steered off the edge of the course.

EQUIPMENT REQUIRED

Basic VIC-20 with no expansion.

RUNNING THE PROGRAM

Load and run part one and follow instructions .
Steering is obtained using key L to go left and ';' to go right.
Avoid all oil slicks and oncoming vehicles.

PROGRAM STRUCTURE

The lines of most interest are as follows:

Part One:

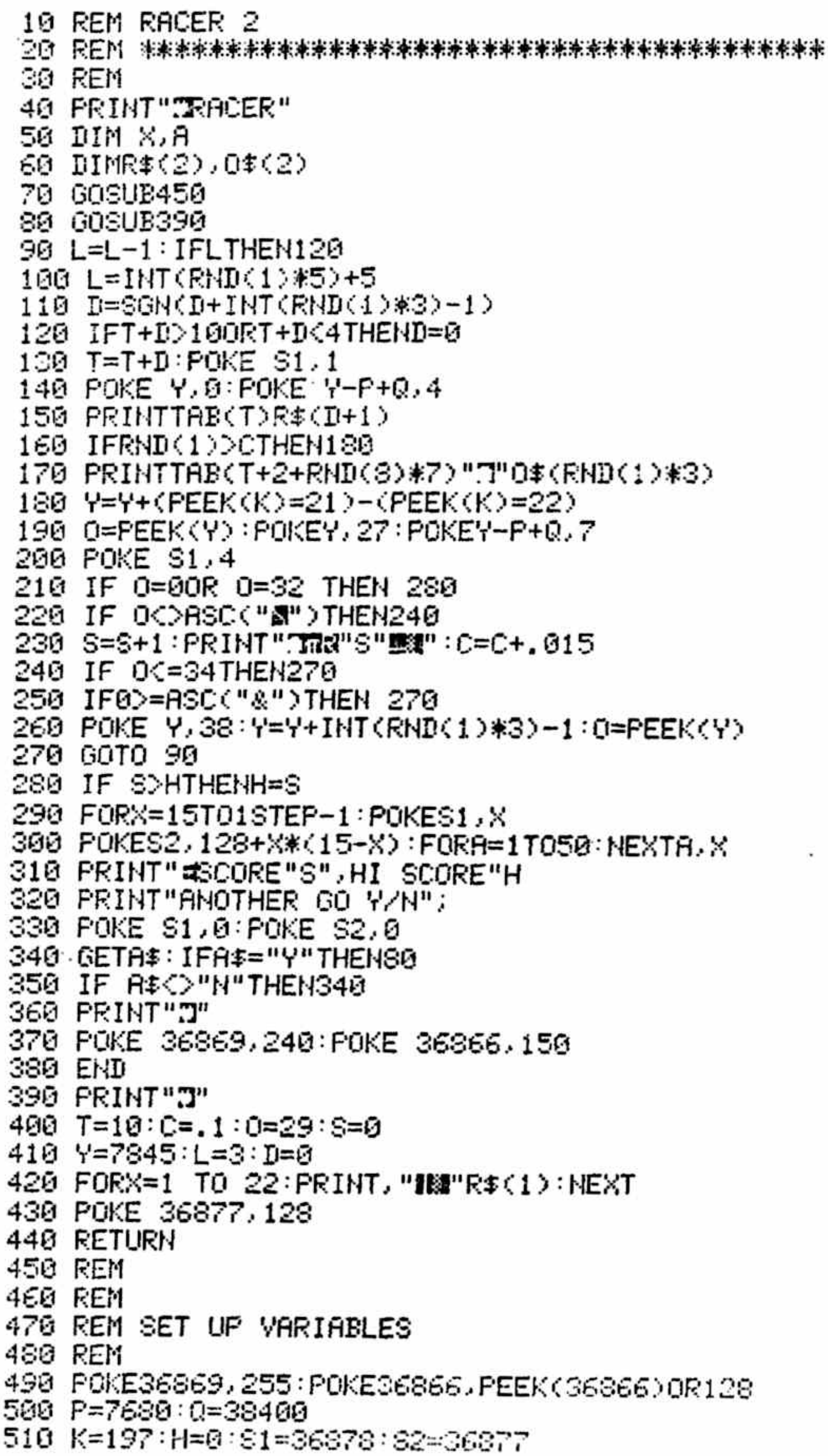
40	Limit top of string space to protect characters
50	Black screen
70-90	Move character generator
100	Set graphics mode
120-140	Pause until instructions are read
180-340	Set up graphics characters from data
350-550	Character data
560-640	Instructions

Part Two:

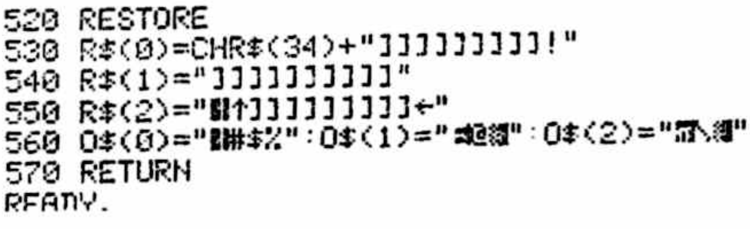
40	Clear the screen and display title
50-60	Arrays for track, cars and obstacles
70-380	Main game routine
390-440	Set up initial screen display
490-570	Set up track and obstacles into arrays r$ and o$

```
10 REM RACER
20 REM ****************************************
30 REM
40 POKE 51,0:POKE 52,28
50 POKE 36879,8
60 GOSUB 560
70 FOR X=0TO511
80 POKE 7168+X,PEEK(32768+X)
90 NEXT
100 POKE36869,255:POKE 36866,PEEK(36866)OR 128
110 GOSUB 170
120 PRINT"■■■HIT SPACE TO CONTINUE"
130 GETA$:IFA$<>" "THEN130
140 POKE 36869,240:POKE36866,150
150 PRINT"■NOW LOAD & RUN THE     ■NEXT PROGRAM"
160 END
169 REM
170 REM CHARACTER DATA ENTERED
171 REM
180 READ X:X=7168+X*8
190 IF X<7000 THEN RETURN
200 READ A$:IFLEN(A$)=8THEN260
210 POKE X,VAL(A$)
220 FORY=1TO7
230 READ A:POKE X+Y,A
240 NEXT
250 GOTO 170
260 GOSUB 310
270 FORX=X+1TOX+6
280 GOSUB 300:NEXT
290 GOTO 170
300 READ A$
310 A=0:FORY=1TO8
320 A=A*2-(MID$(A$,Y,1)="*"):NEXT
330 POKEX,A
340 RETURN
344 REM
345 REM CHARACTER DATA
346 REM
350 DATA 0,153,255,165,60,60,189,255,129
360 DATA 27,129,255,189,60,60,165,255,153
370 DATA 28,255,129,189,165,165,189,129,255
380 DATA 29,255,255,255,255,255,255,255,255
390 DATA 30,255,127,63,31,15,7,3,1
400 DATA 31,128,192,224,240,248,252,254,255
410 DATA 33,255,254,252,248,240,224,192,128
420 DATA 34,1,3,7,15,31,63,127,255
430 DATA 35,240,192,131,2,2,131,192,240
440 DATA 36,0,0,212,84,84,215,0,0
450 DATA 37,15,3,1,0,0,129,3,15
460 DATA 38
```

```
470 DATA...**...
480 DATA..****..
490 DATA.*.**.*.
500 DATA*.****.*
510 DATA.*.**.*.
520 DATA..****..
530 DATA...**...
540 DATA...**...
550 DATA -99
554 REM
555 REM INSTRUCTIONS
556 REM
560 PRINT"▒▒RACER"
570 PRINT"-----"
580 PRINT"▒▒USE THE 'L' AND ';'"
590 PRINT"▒KEYS TO STEER YOUR CAR"
600 PRINT"'[' ALONG A ROAD. HIT"
610 PRINT"▒'\' TARGETS FOR POINTS"
620 PRINT"▒BUT AVOID HEAD ON"
630 PRINT"▒COLLISIONS WITH '@'"
640 RETURN
READY.
```



```
 10 REM RACER 2
 20 REM ****************************************
 30 REM
 40 PRINT"▛RACER"
 50 DIM X,A
 60 DIMR$(2),O$(2)
 70 GOSUB450
 80 GOSUB390
 90 L=L-1:IFLTHEN120
 100 L=INT(RND(1)*5)+5
 110 D=SGN(D+INT(RND(1)*3)-1)
 120 IFT+D>10ORT+D<4THEND=0
 130 T=T+D:POKE S1,1
 140 POKE Y,0:POKE Y-P+Q,4
 150 PRINTTAB(T)R$(D+1)
 160 IFRND(1)>CTHEN180
 170 PRINTTAB(T+2+RND(8)*7)"▜"O$(RND(1)*3)
 180 Y=Y+(PEEK(K)=21)-(PEEK(K)=22)
 190 O=PEEK(Y):POKEY,27:POKEY-P+Q,7
 200 POKE S1,4
 210 IF O=0OR O=32 THEN 280
 220 IF O<>ASC("▞")THEN240
 230 S=S+1:PRINT"▜▛▞"S"▓▓":C=C+.015
 240 IF O<=34THEN270
 250 IF0>=ASC("&")THEN 270
 260 POKE Y,38:Y=Y+INT(RND(1)*3)-1:O=PEEK(Y)
 270 GOTO 90
 280 IF S>HTHENH=S
 290 FORX=15TO1STEP-1:POKES1,X
 300 POKES2,128+X*(15-X):FORA=1TO50:NEXTA,X
 310 PRINT"≢SCORE"S",HI SCORE"H
 320 PRINT"ANOTHER GO Y/N";
 330 POKE S1,0:POKE S2,0
 340 GETA$:IFA$="Y"THEN80
 350 IF A$<>"N"THEN340
 360 PRINT"▛"
 370 POKE 36869,240:POKE 36866,150
 380 END
 390 PRINT"▛"
 400 T=10:C=.1:O=29:S=0
 410 Y=7845:L=3:D=0
 420 FORX=1 TO 22:PRINT,"▮▓"R$(1):NEXT
 430 POKE 36877,128
 440 RETURN
 450 REM
 460 REM
 470 REM SET UP VARIABLES
 480 REM
 490 POKE36869,255:POKE36866,PEEK(36866)OR128
 500 P=7680:Q=38400
 510 K=197:H=0:S1=36878:S2=36877
```



```
520 RESTORE
530 R$(0)=CHR$(34)+"]]]]]]]]]]!"
540 R$(1)="]]]]]]]]]]"
550 R$(2)="▓↑]]]]]]]]]]←"
560 O$(0)="▓#$%":O$(1)="▓▓▓":O$(2)="▓\▓"
570 RETURN
REA0Y.
```

CAR RACE

DESCRIPTION

This game is a computerised version of the Grand Prix. The player has five choices of internationally famous racing drivers to emulate. Having chosen your driver the players—you and the computer—line up at the starting point. The display is an attractive picture of a tree-lined track. The course runs over five laps and should the player steer off the course he hits the trees but is still able to continue because the player is playing against time; the VIC player is programmed to complete the course precisely.

EQUIPMENT REQUIRED

Basic VIC-20 with 3K expansion.

RUNNING THE PROGRAM

Load the Game and type run. Steering is obtained using a group of keys on the left hand side of the keyboard. These are as follows:

Q W E

A S D

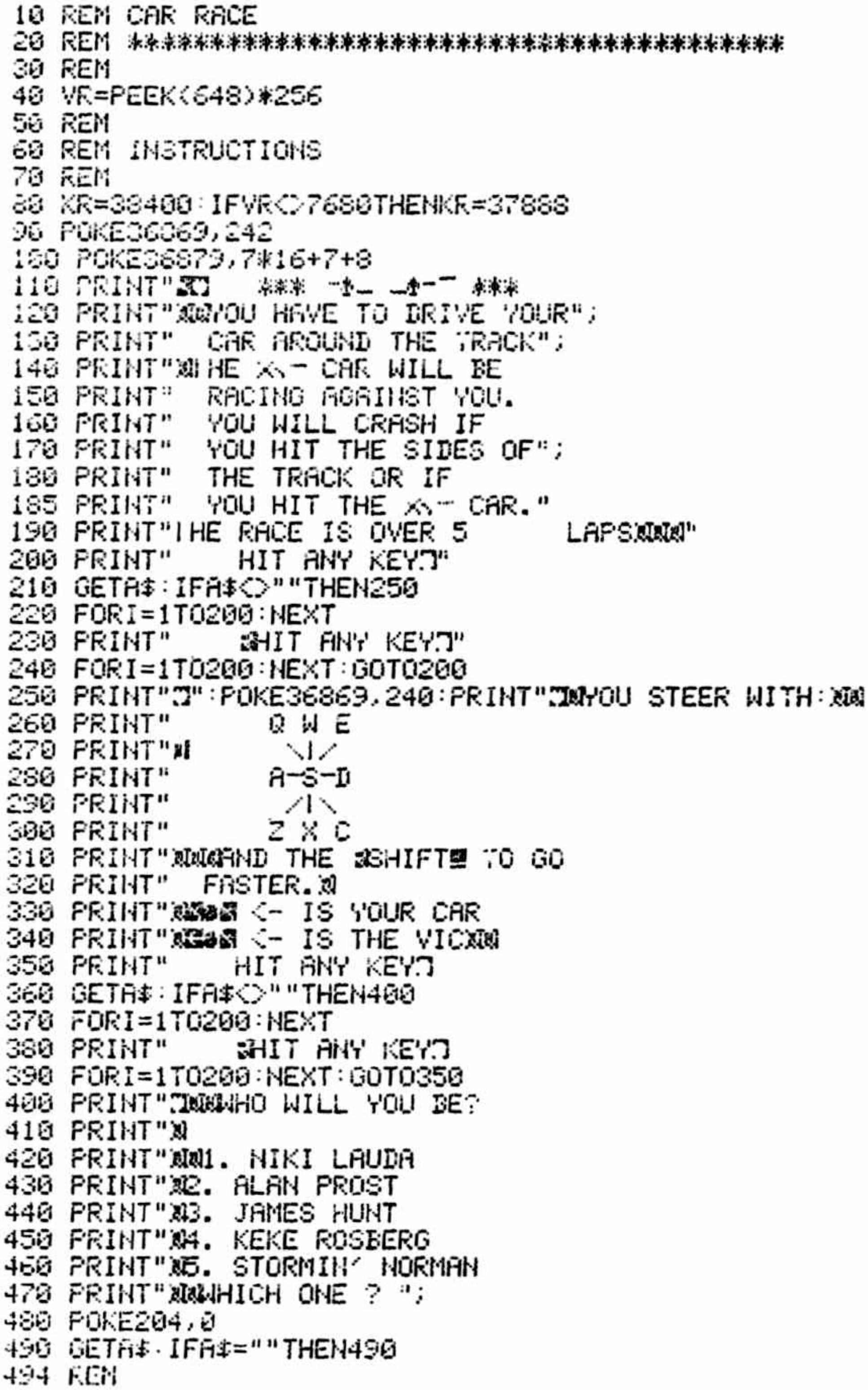
Z X C

These directions are based around the central position of 'S', corresponding to the points compass.

PROGRAM STRUCTURE

The lines of most interest in this program are as follows:

40-80	Decide on video map
100-490	Instructions
500-1000	Main game routine
1020-1250	Display the track



```
10 REM CAR RACE
20 REM ****************************************
30 REM
40 VR=PEEK(648)*256
50 REM
60 REM INSTRUCTIONS
70 REM
80 XR=38400:IFVR<>7680THENKR=37888
90 POKE36869,242
100 POKE36879,7*16+7+8
110 PRINT"▒▒    *** -↑_ _↑-‾ ***
120 PRINT"▒▒YOU HAVE TO DRIVE YOUR";
130 PRINT"  CAR AROUND THE TRACK";
140 PRINT"▒IHE ✕\- CAR WILL BE
150 PRINT"  RACING AGAINST YOU.
160 PRINT"  YOU WILL CRASH IF
170 PRINT"  YOU HIT THE SIDES OF";
180 PRINT"  THE TRACK OR IF
185 PRINT"  YOU HIT THE ✕\- CAR."
190 PRINT"IHE RACE IS OVER 5       LAPS▒▒▒"
200 PRINT"     HIT ANY KEY▒"
210 GETA$:IFA$<>""THEN250
220 FORI=1TO200:NEXT
230 PRINT"     ▒HIT ANY KEY▒"
240 FORI=1TO200:NEXT:GOTO200
250 PRINT"▒":POKE36869.240:PRINT"▒▒YOU STEER WITH:▒▒
260 PRINT"       Q W E
270 PRINT"▒       \|/
280 PRINT"       A-S-D
290 PRINT"        /|\
300 PRINT"       Z X C
310 PRINT"▒▒▒AND THE ▒SHIFT▒ TO GO
320 PRINT"  FASTER.▒
330 PRINT"▒▒▒▒ <- IS YOUR CAR
340 PRINT"▒▒▒▒ <- IS THE VIC▒▒
350 PRINT"     HIT ANY KEY▒
360 GETA$:IFA$<>""THEN400
370 FORI=1TO200:NEXT
380 PRINT"     ▒HIT ANY KEY▒
390 FORI=1TO200:NEXT:GOTO350
400 PRINT"▒▒▒WHO WILL YOU BE?
410 PRINT"▒
420 PRINT"▒▒1. NIKI LAUDA
430 PRINT"▒2. ALAN PROST
440 PRINT"▒3. JAMES HUNT
450 PRINT"▒4. KEKE ROSBERG
460 PRINT"▒5. STORMIN' NORMAN
470 PRINT"▒▒WHICH ONE ? ";
480 POKE204,0
490 GETA$:IFA$=""THEN490
494 REM
```

```
495 REM PLAY THE GAME
496 REM
500 POKE204,1:PRINT" ▮";
510 SL=VAL(A$):IFSL>5ORSL<1THEN480
520 DIMN(28),D(28)
530 DATA3,2,3,1,2,4,3,7,3,8,1,7,2,4,1,1,5,2,3
540 DATA1,1,4,1,7,12,8,1,9,3,8,1,9,2,6
550 DATA1,3,3,2,1,3,3,6,2,9,1,8,2,9,2,6,2,3,7,2
560 FORI=1TO27:READN(I),D(I):NEXT
570 DATA21,22,23,-1,0,1,-23,-22,-21
580 FORI=1TO9:READM(I):NEXT
590 DATA"Z","X","C","A","S","D","Q","W","E"
600 FORI=1TO9:READT$(I):NEXT
610 YP=VR+259:CP=YP+2:GOSUB1020
620 GETB$:IFB$=""THEN620
630 GETA$:IFA$=""THENA$=B$
640 B$=A$:POKEYK,0:POKEYP,32
650 IFA$=""THEN700
660 FORI=1TO9
670 IF(ASC(A$)AND127)=ASC(T$(I))THENA$=STR$(I):I=10
680 NEXT
690 IFASC(A$)>128THENA$=CHR$(ASC(A$)-128)
700 YP=YP+M(VAL(A$))
710 IFPEEK(YP)=32ORPEEK(YP)=96THEN760
720 IFNOT(PEEK(YP)=61ANDM(VAL(A$))>0)THEN730
725 YL=YL+1:YP=YP+22:GOTO760
730 IFPEEK(YP)=83THENGOSUB950:GOTO760
740 GOSUB950
750 B$="":YP=YP-M(VAL(A$)):A$=""
760 POKEYP,81
770 YK=YP-VR+KR:POKEYK,2
780 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";
785 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"YL
790 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"CL
800 IFTG=1THENTG=0:GOTO630
810 IFRND(TI)<SL*.09THENGOSUB890
820 IFCS=1ANDSL=5ANDRND(TI)>.2THENGOSUB890
830 IFPEEK(654)=1THENTG=1:GOTO850
840 FORU=1TO100:NEXT
850 GOSUB890
860 IFYL=5THEN1260
870 IFCL=5THEN1260
880 GOTO630
890 CS=0:IFC=0THENG=G+1:C=N(G):D=M(D(G))
900 IFN(G)>5THENCS=1
910 IFG<>28THEN930
920 G=0:CL=CL+1:CP=7941:C=0:POKECP-22,61:GOTO890
930 C=C-1:POKECK,0:POKECP,32:CP=CP+D:POKECP,81
940 CK=CP-VR+KR:POKECK,6:RETURN
950 FORI=1TO9:R=PEEK(YP+M(I))
960 IFR<128THENR=R+128:POKEYP+M(I),R
970 NEXT
```

```
980 FORI=9TO1STEP-1:R=PEEK(YP+M(I))
990 IFR>128THENR=R-128:POKEYP+M(I),R
1000 NEXT:RETURN
1010 END
1014 REM
1015 REM DISPLAY THE TRACK
1016 REM
1020 POKE36879,16+8+1
1030 PRINT"□ ▓▶◆■_____▓▶◆◆◆◆◆◆◆■_____▓▶◆◆
1040 PRINT" ▓▶■/     \▓▶◆◆◆◆◆■/     \▓▶◆"
1050 PRINT" ■ |      \▓▶◆◆◆■/      \▓▶"
1060 PRINT" ■ |  /‾\  |▓▶■ |  /‾\   | "
1070 PRINT" ■ |  |▓▶■ |  \_/  |▓▶■ |   | "
1080 PRINT" ■ |  |▓▶■ |      /▓▶◆■ |   | "
1090 PRINT" ■ |  |▓▶◆■\     /▓▶◆◆■ |   | "
1100 PRINT" ■ ;  |▓▶◆◆■‾‾‾‾‾▓▶◆◆◆■ |   | "
1110 PRINT" ■ !  |▓▶◆■_____▓▶◆◆◆■ |   | "
1120 PRINT" ■ |  |▓▶■/     \▓▶◆◆■ |   | "
1130 PRINT" ■ | /▓▶■/      \▓▶◆■ ■===■| "
1140 PRINT" ■ || |   /‾\   |▓▶■ |   | "
1150 PRINT" ■ || |   |▓▶■ |  |▓▶■ |   | "
1160 PRINT" ■ || |   |▓▶■ |  |▓▶■ |   | "
1170 PRINT" ■ || |   \▓▶■ |  |▓▶■ |   | "
1180 PRINT" ■ ||▓▶■\   | |  \_/    | "
1190 PRINT" ■ ||▓▶■ |   |▓▶■\       /▓▶"
1200 PRINT" ■/ \_/    |▓▶◆■\     /▓▶◆"
1210 PRINT" ■|       /▓▶◆◆◆■\  /▓▶◆◆"
1220 PRINT" ■\      /▓▶◆◆◆◆◆■‾‾‾▓▶◆◆◆"
1230 PRINT" ▓▶■\    /▓▶■ ● _APS 0 ▓▶◆"
1240 PRINT" ▓▶◆■‾‾‾‾▓▶◆■ ● _APS 0 ▓▶◆■"
1250 RETURN
1260 POKE36879,27:PRINT"□■
READY.
```

TAPE SEARCH

DESCRIPTION

This program is a utility to enable a fast forward search of a program tape. It will not recognise sequential files. The program is set up for ten programs but a larger number will do.

EQUIPMENT REQUIRED

Basic VIC-20.

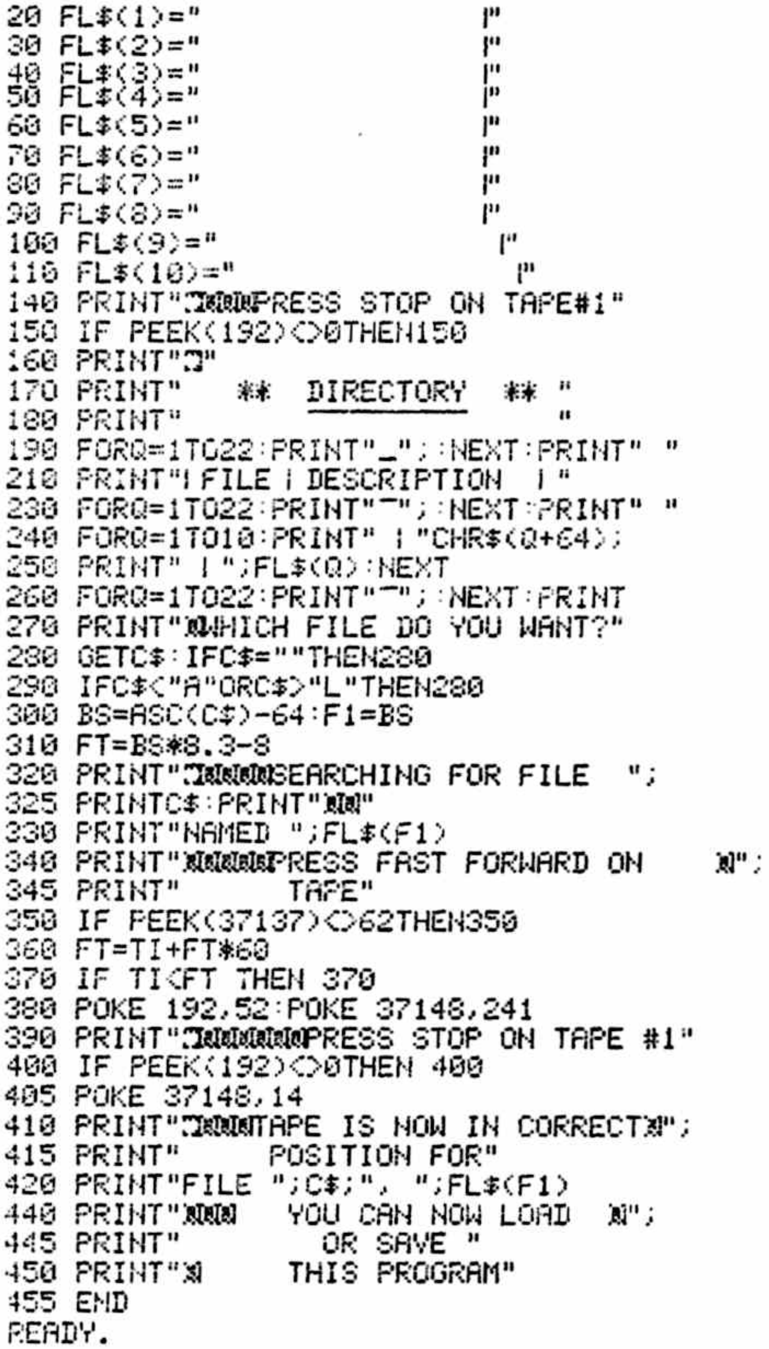
RUNNING THE PROGRAM

Once typed in, the array FL$ is set up with the names of the programs on the tape. Type run and a directory of the tape is displayed on the screen from the array FL$. Hitting the correspondence key will tell the computer which program is required. The program will then ensure that the stop key has been pressed before requesting that the fast forward key be pressed. After that, all you have to do is wait until you are told to press the stop key and then the tape is in the correct position to load the requested file. Note that the tape must be rewound or set to the position of the first file in FL$.

PROGRAM STRUCTURE

The lines of interest in this program are as follows.

20-110	Set up the directory.
140-150	Check that no keys are depressed on the tape deck.
160-310	Print directory and choose program.
320-380	Searching for the file.
390-455	Tape is now ready to load from.



```
20 FL$(1)="                  |"
30 FL$(2)="                  |"
40 FL$(3)="                  |"
50 FL$(4)="                  |"
60 FL$(5)="                  |"
70 FL$(6)="                  |"
80 FL$(7)="                  |"
90 FL$(8)="                  |"
100 FL$(9)="                  |"
110 FL$(10)="                  |"
140 PRINT"♥ⓆⓆⓆPRESS STOP ON TAPE#1"
150 IF PEEK(192)<>0THEN150
160 PRINT"♥"
170 PRINT"    **  DIRECTORY  ** "
180 PRINT"                     "
190 FORQ=1TO22:PRINT"_";:NEXT:PRINT"  "
210 PRINT"| FILE | DESCRIPTION  | "
230 FORQ=1TO22:PRINT"¯";:NEXT:PRINT"  "
240 FORQ=1TO10:PRINT" | "CHR$(Q+64);
250 PRINT" | ";FL$(Q):NEXT
260 FORQ=1TO22:PRINT"¯";:NEXT:PRINT
270 PRINT"ⓆWHICH FILE DO YOU WANT?"
280 GETC$:IFC$=""THEN280
290 IFC$<"A"ORC$>"L"THEN280
300 BS=ASC(C$)-64:F1=BS
310 FT=BS*8.3-8
320 PRINT"♥ⓆⓆⓆⓆSEARCHING FOR FILE  ";
325 PRINTC$:PRINT"ⓆⓆ"
330 PRINT"NAMED ";FL$(F1)
340 PRINT"ⓆⓆⓆⓆⓆPRESS FAST FORWARD ON     Ⓠ";
345 PRINT"       TAPE"
350 IF PEEK(37137)<>62THEN350
360 FT=TI+FT*60
370 IF TI<FT THEN 370
380 POKE 192,52:POKE 37148,241
390 PRINT"♥ⓆⓆⓆⓆⓆⓆPRESS STOP ON TAPE #1"
400 IF PEEK(192)<>0THEN 400
405 POKE 37148,14
410 PRINT"♥ⓆⓆⓆTAPE IS NOW IN CORRECTⓆ";
415 PRINT"      POSITION FOR"
420 PRINT"FILE ";C$;", ";FL$(F1)
440 PRINT"ⓆⓆⓆ   YOU CAN NOW LOAD  Ⓠ";
445 PRINT"         OR SAVE "
450 PRINT"Ⓠ     THIS PROGRAM"
455 END
READY.
```

APPENDIX TO SPECIAL PRINT CHARACTERS

KEYS TO PRESS		*HOW IT APPEARS IN PRINT*	*DEFINITION*
CTRL	! 1	■	BLACK
CTRL	" 2	E	WHITE
CTRL	# 3	£	RED
CTRL	$ 4	◣	CYAN
CTRL	% 5	▒	PURPLE
CTRL	& 6	↑	GREEN
CTRL	' 7	←	BLUE
CTRL	(8	π	YELLOW
CTRL	) 9	R	REVERSE ON
CTRL	0	_	REVERSE OFF
SHIFT	CLR HOME	♥	CLEAR THE SCREEN AND HOME
	↑ CRSR ↓	Q	CURSOR DOWN
SHIFT	↑ CRSR ↓	●	CURSOR UP
	← CRSR →	]	CURSOR RIGHT
SHIFT	← CRSR →	|	CURSOR LEFT
	CLR HOME	S	HOME

INDEX

BREAKOUT 6

This program is a well adapted version of the popular arcade game of the same name.

FIND THE WORD 11

A recognition test—useful for teaching and learning.

SPACE PIRATES 17

An amusing game in a race against time.

VIC VIC 22

A program demonstrating the use of the control registers.

BIRDS DEMO 25

An attractive demo for user defined characters.

RHINO 30

An interesting game requiring the user to use logic to escape through the jungle.

DO-RAY-ME 39

A program to enable the user to use the VIC-20 keyboard as a musical keyboard.

SOUND EFFECTS 43

A program for use on its own or in a program written by the user.

ARROW 47

An amusing game which tests the user's co-ordination.

TANK-V-UFO 52

A simple game with the user taking the part of the tank.

LANDMINE 58

A maze game with increasing difficulty for the user.

SPACEWAR 65

Two programs making an interesting and intriguing game for the user.

JOYSTICK TEST 72

A useful routine for the user to incorporate in his own programs.

DEFINE KEYS 74

Enables the user to define the eight functions keys on the keyboard.

U.S.A. SONG 77
A demonstration program using the VIC-20 sound output to play a tune.

DIGICLOCK 82
A program demonstrating the time keeping facility of the VIC-20.

LEAP FROG 85
A puzzle for the mathematically minded.

RUBIK'S CUBE 92
An absorbing version of Mr. Rubik's addictive game.

BOSS PUZZLE 97
Tests the mental agility of the user to re-arrange a mixed up block of numbers into a sequential pattern.

SKETCHING 1 102
This program enables the user to produce displays using the Commodore graphics symbols.

SKETCHING 2 106
This program provides the user with an 8 × 8 grid to sketch in high resolution using the joystick.

KALEIDOSCOPE 110
Creates a colourful and attractive changing pattern on the screen.

HI-RES DEMO 113
A program to demonstrate the high resolution capabilities of the VIC-20 when using the Super Expander Cartridge.

BANDIT 116
This game is a highly addictive version of the one armed bandit pub game.

MOON LANDER 122
A realistic game/test for the would be astronauts.

CIRCLE DEMO 127
A program to demonstrate the use of the character generator when displaying circles in high resolution.

HI-RES PLOT 130
Enables the user to plot points in high resolution.

HANGMAN 133
An intelligent version of the childrens game, would be useful as a teaching/learning aid.

GOMOKO 141
A test of skill playing the old oriental game of Gomoko (or Go as it is sometimes known).

SUPERMIND 147
A version of Mastermind adapted for the VIC-20.

CONQUEST 153
A tale of Do and Dare as the Knights defend their castle.

HI-RES AID 159
A useful program to design graphics characters.

TINYMON 163
A machine code monitor for the VIC-20.

RACER 170
An exciting and intricate racing game on a vertical plane.

CAR RACE 176
The Grand Prix of video racing.

TAPE SEARCH 180
Fast forward tape search program.

APPENDIX TO SPECIAL PRINT CHARACTERS 183
Useful for recognizing the special characters found in PRINT statements.

1060-5

VIC™ GAMES

NICK HAMPSHIRE

A collection of 36 exciting and useful programs for your Commodore VIC-20 microcomputer. Features arcade-style games, strategy games, and educational word games. Challenge your driving skills on your own Grand Prix race track; fight a war with space pirates; maneuver your way through a maze of landmines; solve the Rubik's cube (if you can!); create your own colorful images and music to go with them; find your way out of the jungle; improve your children's (or your own!) spelling and vocabulary skills; plus much, much more!

Sound, music, utilities, and graphics programs that are fun, easy to play, and instructional for every VIC-20 user.

HAYDEN BOOK COMPANY, INC.
Rochelle Park, New Jersey

ISBN 0-8104-1060-5