62-8001

TRS-80™ 16K

COLOUR COMPUTER

SPACE ADVENTURES

Radio Shack®

SPACE ADVENTURES

FOR THE TRS-80

Programs Adapted by Scott McCann
and Derek Simonson
Programs Edited by Peter Lear

Editor:	Cyril Hayes
Design:	Karen McArdle
	Margaret Horvath
Illustration:	Richard Livingston

Radio Shack, 279 Bayview Drive, Barrie, Ontario. L4M 4W5

NOTE

The publisher and authors have made every effort to assure that the computer programs are accurate and complete. However, this publication is prepared for general readership, and neither the publisher nor the authors have any knowledge about or ability to control any third party's use of the programs and programming information. There is no warranty or representation by the publisher or the authors that the programs or programming information in this book will enable the reader or user to achieve any particular result.

ISBN 0-88625-081-1

CONTENTS

INTRODUCTION

This book contains listings for games written in BASIC for the TRS-80 COLOR COMPUTER made by Radio Shack.

You probably know from your owner's manual that the question mark (?) is a short way of typing in PRINT. You can save time by using it when you are typing in these programs.

Remember, accidents do happen; you could type in many lines of a program only to have a power failure, or you could unknowingly RUN a program with mistakes that destroy the program. So SAVE copies of the game on tape or floppy disk as you enter it. Once you have a final copy of the game you can record over the other versions on tape or delete the disk versions.

DEBUGGING

After typing in a program and SAVEing it, the TRS-80 may have trouble RUNning it. This happens when copying a program from paper to computer. Check that you entered all the program lines completely. A common problem is the famous SYNTAX ERROR. If you get one, list the program line on the screen. Compare it to the book listing. You will probably see one of these problems:

1. Spelling error, or
2. Punctuation error (brackets, commas, colons or semi-colons missing), or
3. The number zero confused with the letter 'O' (or vice versa), or
4. The number one confused with the letter 'I' (or vice versa).

Fix it and try RUNning the program again. It may take several attempts to get all the errors out, but the work will be worth it. Remember to SAVE a final copy that has all the corrections made.

FASTER PROGRAMS

The games in this book are pretty fast, but as you master them you may find them more challenging if they were faster. There are two ways to make the games faster.

The first method is to add these lines to any of the programs:

```
1 IF PEEK(&H3EB9) <> &H32 THEN
   CLEAR200,&H3EB0:FOR
   I=&H82B9 TO &H831E:POKE I-
  &H4400,PEEK(I):NEXT ELSE 5
2 FORI=0TO2:POKE &H3EBD+I,
  18:NEXT:I=&H3F1E
3 POKE I,&H26:POKEI+1,3:
  POKEI+2,&H7E:POKEI+3,&H83:
  POKEI+4,&H22:POKEI+5,&H7E
4 POKE I+6,&HA4:POKEI+7,&H4C
```

Adding the above lines will disable the break key, which speeds the games up. To break out now you must use the reset button.

Another way to speed things up is to enter this line before RUNning a program:

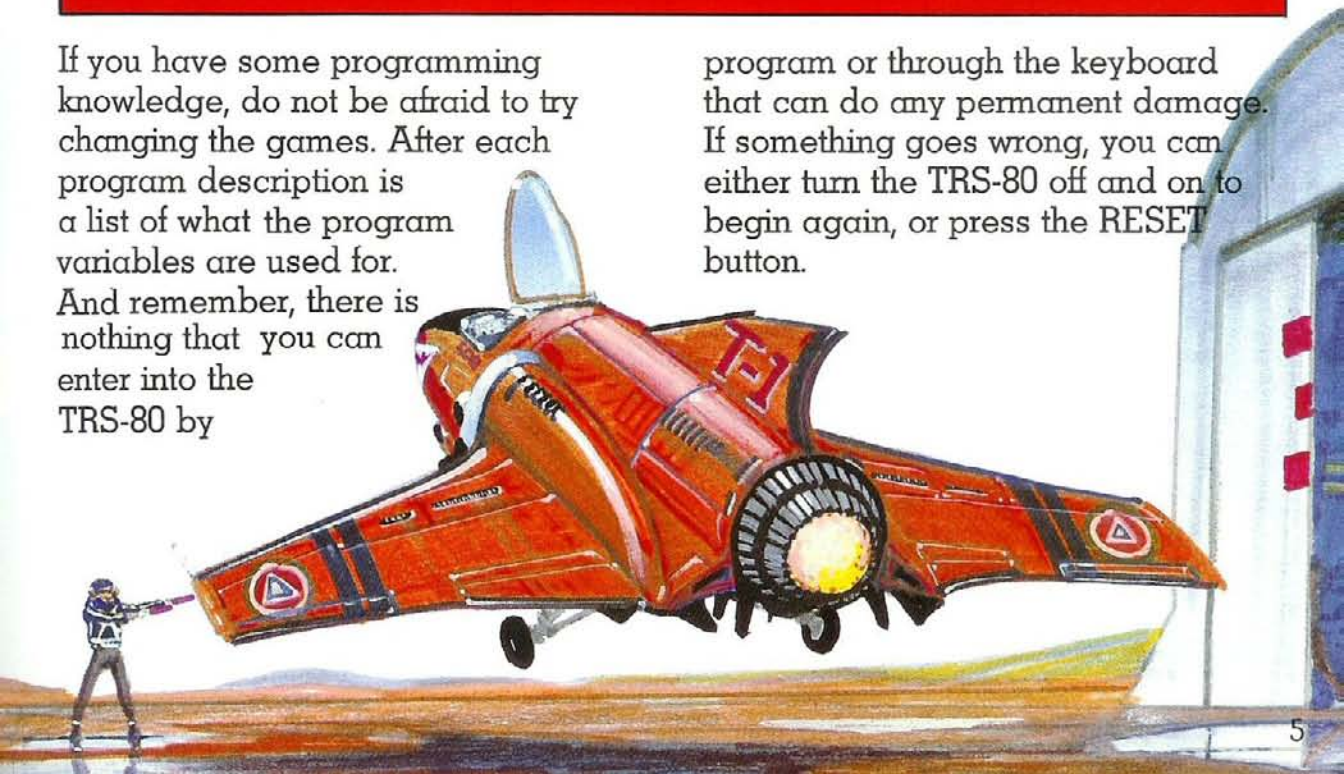
```
POKE 65495,0
```

This line nearly doubles the speed of the Color Computer. You will notice even the sound speeds up. Before doing any disk or tape access you must press reset or enter this line:

```
POKE 65494,0
```

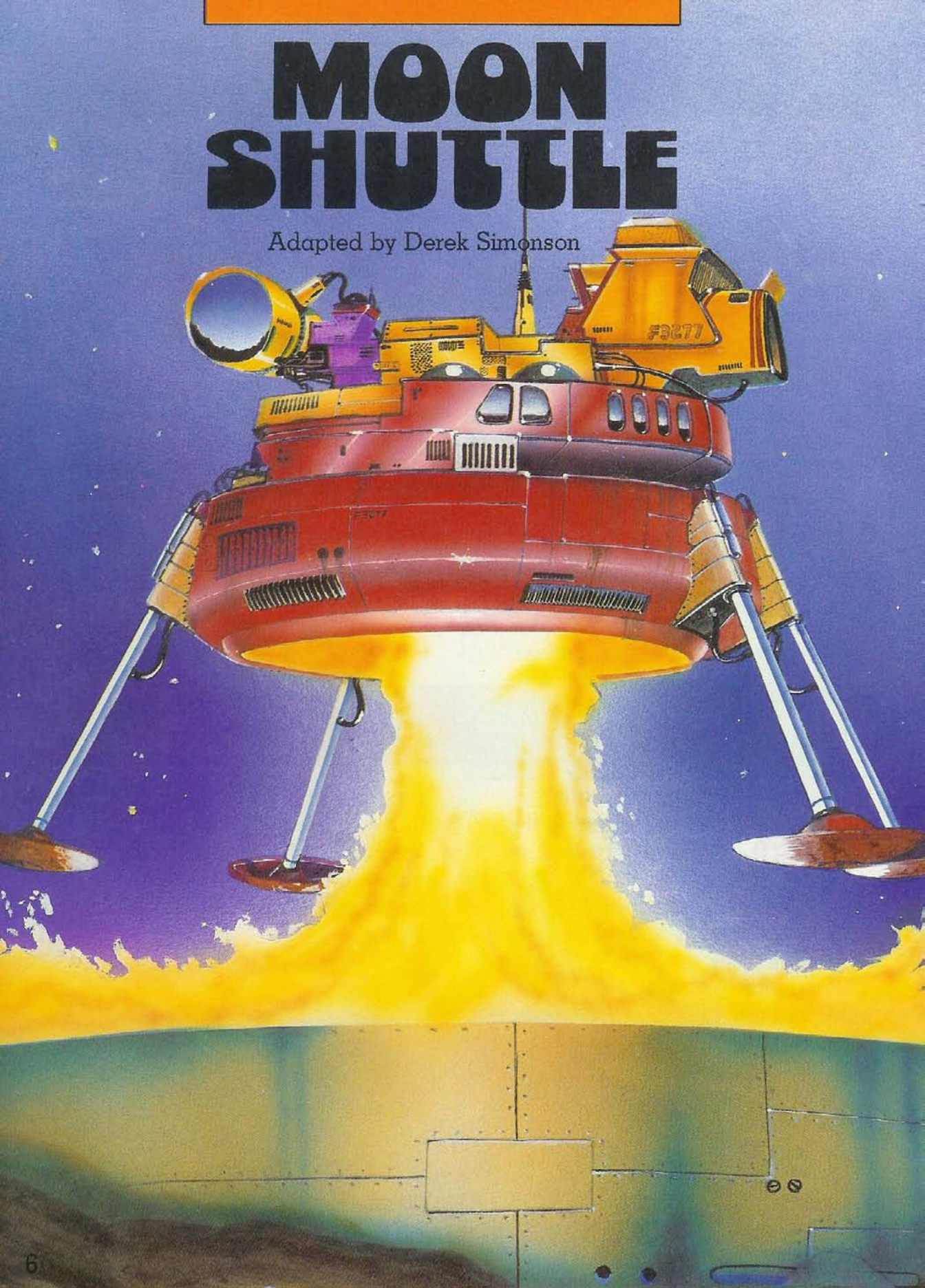
EXPERIMENT

If you have some programming knowledge, do not be afraid to try changing the games. After each program description is a list of what the program variables are used for. And remember, there is nothing that you can enter into the TRS-80 by program or through the keyboard that can do any permanent damage. If something goes wrong, you can either turn the TRS-80 off and on to begin again, or press the RESET button.



MOON SHUTTLE

Adapted by Derek Simonson

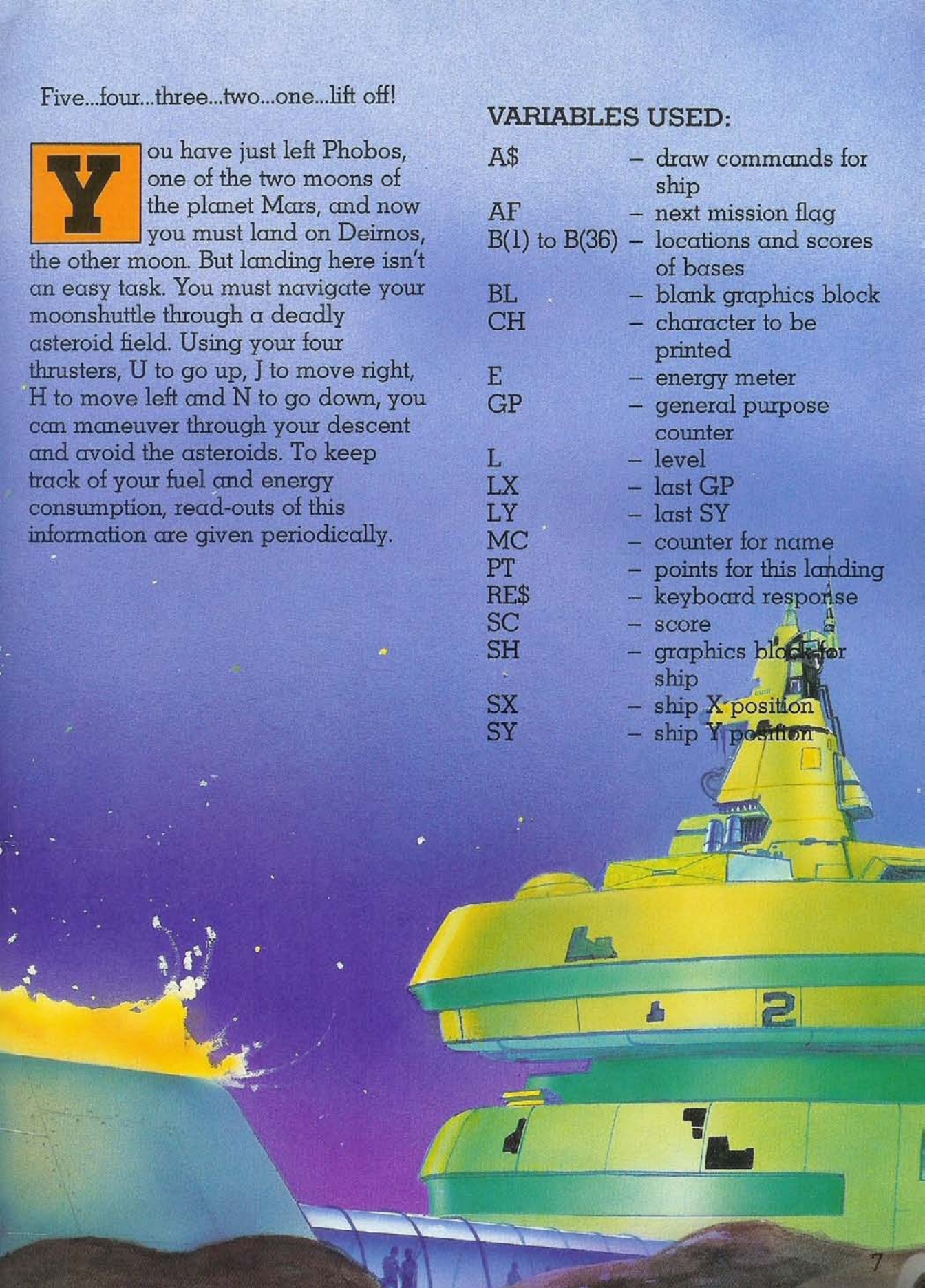


Five...four...three...two...one...lift off!

You have just left Phobos, one of the two moons of the planet Mars, and now you must land on Deimos, the other moon. But landing here isn't an easy task. You must navigate your moonshuttle through a deadly asteroid field. Using your four thrusters, U to go up, J to move right, H to move left and N to go down, you can maneuver through your descent and avoid the asteroids. To keep track of your fuel and energy consumption, read-outs of this information are given periodically.

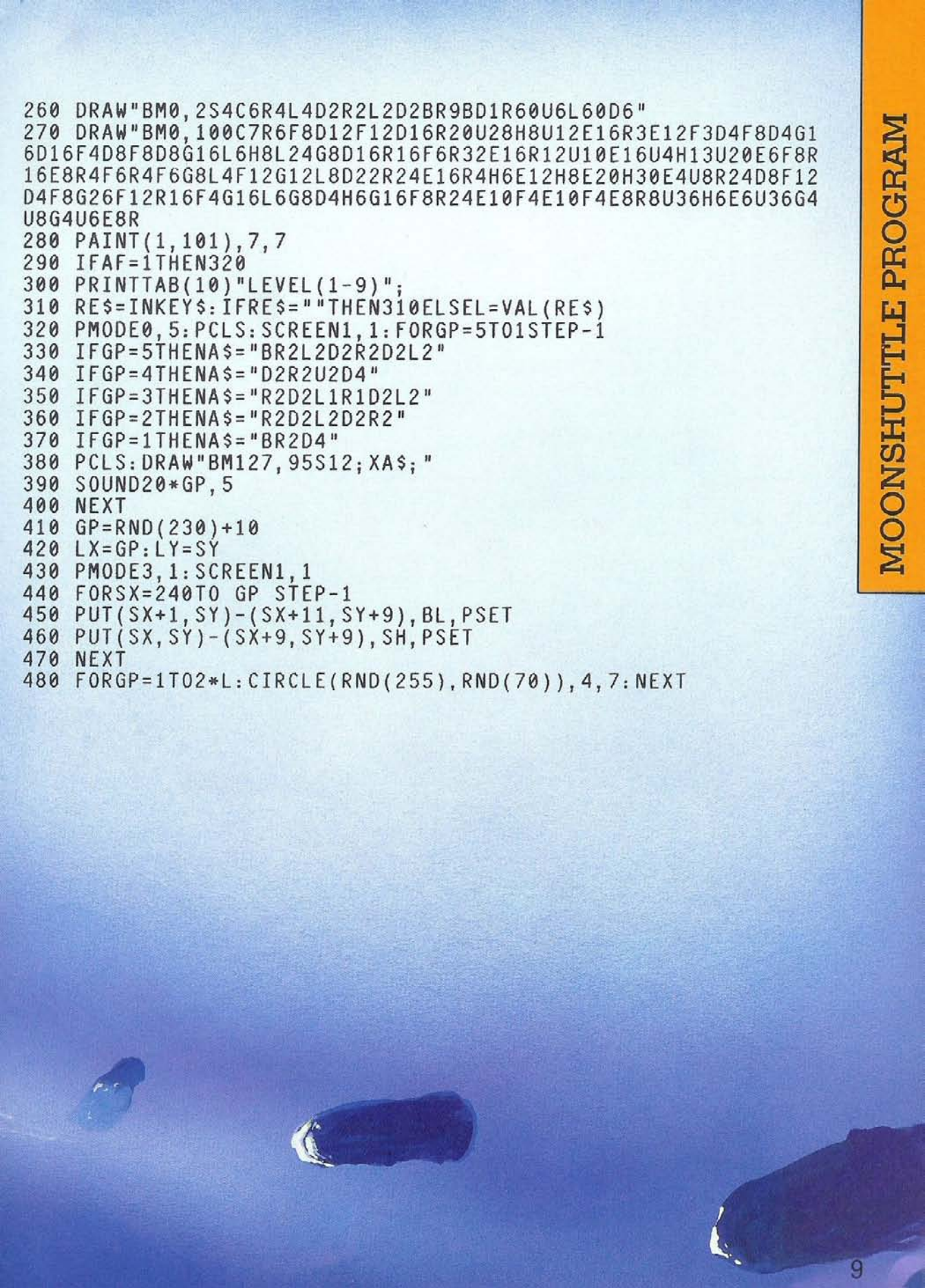
VARIABLES USED:

A$	– draw commands for ship
AF	– next mission flag
B(1) to B(36)	– locations and scores of bases
BL	– blank graphics block
CH	– character to be printed
E	– energy meter
GP	– general purpose counter
L	– level
LX	– last GP
LY	– last SY
MC	– counter for name
PT	– points for this landing
RE$	– keyboard response
SC	– score
SH	– graphics block for ship
SX	– ship X position
SY	– ship Y position



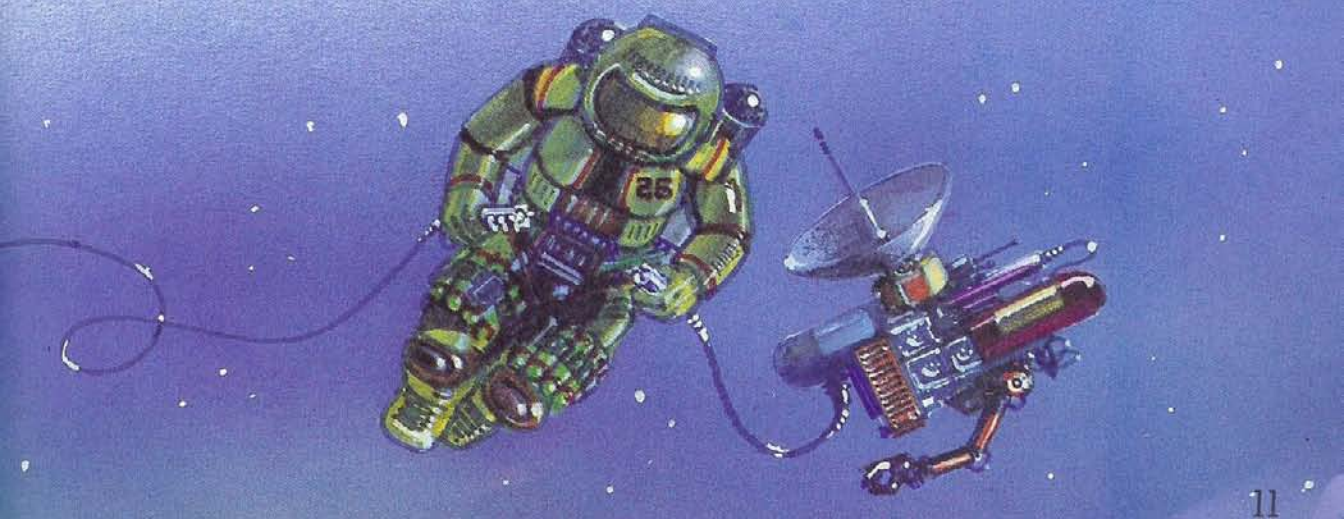
```
10 PCLEAR5
20 DIM SH(20),BL(20),B(36)
30 SY=9
40 E=30
50 FORGP=0TO35:READB(GP):NEXT
60 DATA28,147,50,30,36,188,72,90,16,182,30,120,112,111,130,2
0,140,169,164,50,174,65,196,20,204,135,219,40,180,185,203,60
,84,172,97,60
70 CLS:PRINT:PRINT
80 FORMC=1TO11
90 READCH
100 FORGP=22TO10STEP-1
110 DATA77,79,79,78,83,72,85,84,84,76,69
120 POKEGP+1054+MC,CH
130 IFMC=11THENPOKEGP+1055+MC,96
140 NEXT:NEXT
150 PRINT:PRINT:PRINT"        BY P.A. ROBERTS"
160 PRINT"  CONVERSION BY DEREK SIMONSON
170 PRINT:PRINTTAB(6)"INSTRUCTIONS(Y/N)"
180 RE$=INKEY$:IFRE$=""THEN180ELSEIFRE$="Y"THENGOSUB1030
190 PMODE3,1:PCLS
200 DRAW"BM20,20C6R2F2D4G2BR2BD1U1BL2L2BL2BD1U1BR2H2U4E2
210 PAINT(22,22),6,6
220 GET(17,20)-(26,29),SH,G
230 PUT(50,50)-(59,59),SH,PSET
240 PCLS:GET(16,20)-(25,29),BL,G
250 COLOR8,5:FORGP=0TO35STEP4:LINE(B(GP),B(GP+1))-(B(GP+2),B
(GP+1)),PSET:NEXT
```

```
260 DRAW"BM0,2S4C6R4L4D2R2L2D2BR9BD1R60U6L60D6"
270 DRAW"BM0,100C7R6F8D12F12D16R20U28H8U12E16R3E12F3D4F8D4G1
6D16F4D8F8D8G16L6H8L24G8D16R16F6R32E16R12U10E16U4H13U20E6F8R
16E8R4F6R4F6G8L4F12G12L8D22R24E16R4H6E12H8E20H30E4U8R24D8F12
D4F8G26F12R16F4G16L6G8D4H6G16F8R24E10F4E10F4E8R8U36H6E6U36G4
U8G4U6E8R
280 PAINT(1,101),7,7
290 IFAF=1THEN320
300 PRINTTAB(10)"LEVEL(1-9)";
310 RE$=INKEY$:IFRE$=""THEN310ELSEL=VAL(RE$)
320 PMODE0,5:PCLS:SCREEN1,1:FORGP=5TO1STEP-1
330 IFGP=5THENA$="BR2L2D2R2D2L2"
340 IFGP=4THENA$="D2R2U2D4"
350 IFGP=3THENA$="R2D2L1R1D2L2"
360 IFGP=2THENA$="R2D2L2D2R2"
370 IFGP=1THENA$="BR2D4"
380 PCLS:DRAW"BM127,95S12;XA$;"
390 SOUND20*GP,5
400 NEXT
410 GP=RND(230)+10
420 LX=GP:LY=SY
430 PMODE3,1:SCREEN1,1
440 FORSX=240TO GP STEP-1
450 PUT(SX+1,SY)-(SX+11,SY+9),BL,PSET
460 PUT(SX,SY)-(SX+9,SY+9),SH,PSET
470 NEXT
480 FORGP=1TO2*L:CIRCLE(RND(255),RND(70)),4,7:NEXT
```



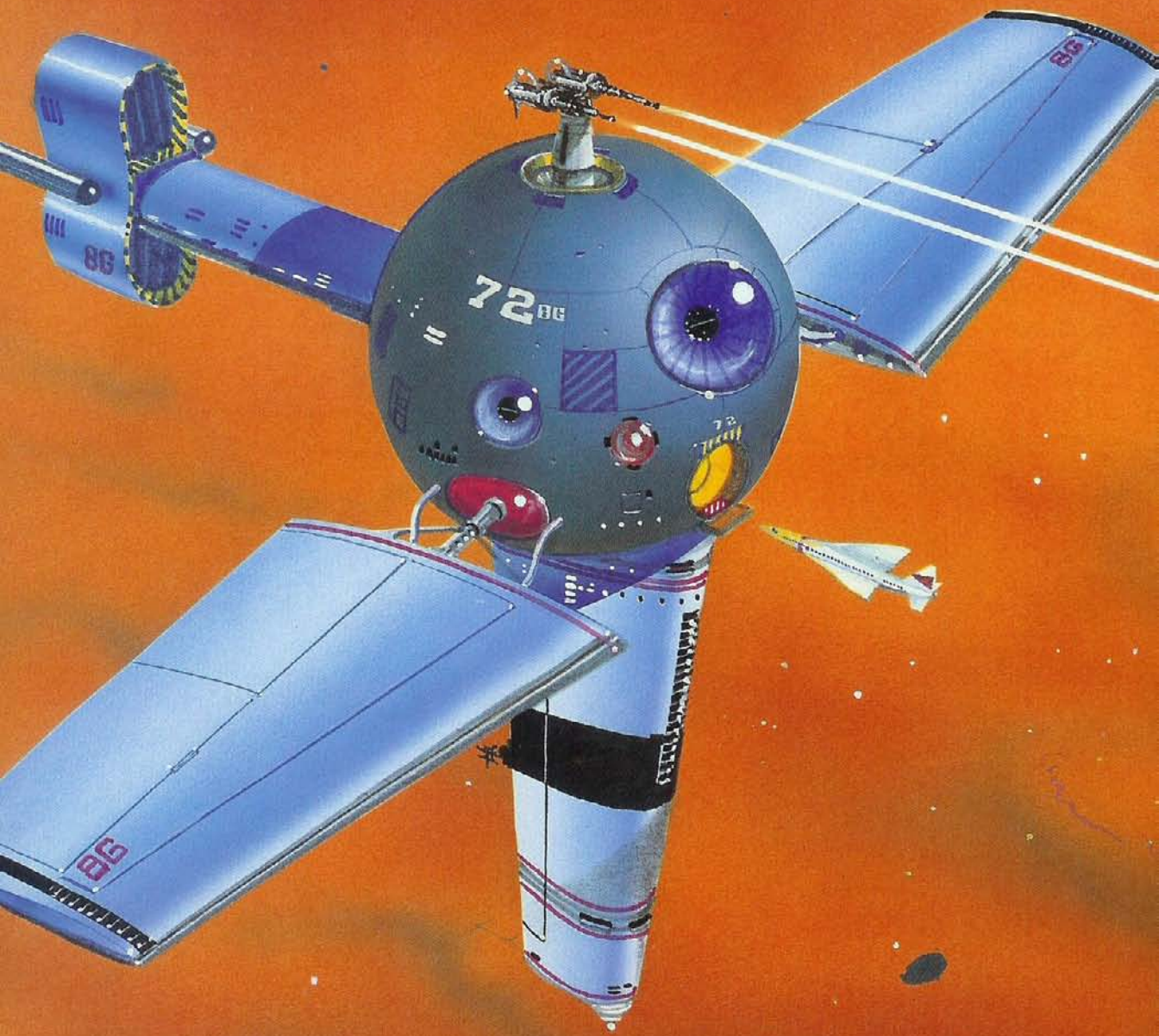
```
490 RE$=INKEY$
500 IFRE$="U"THEN560
510 IFRE$="J"THEN620
520 IFRE$="H"THEN680
530 IFRE$="N"THEN740
540 SY=SY+1:GOSUB800
550 GOTO490
560 GOSUB860:IFV=2THEN540
570 SY=SY-3
580 IFSY<9THENSY=9
590 GOSUB800
600 IFV=2THEN930
610 IFINKEY$<>"U"THEN490ELSE560
620 GOSUB 860:IFV=2THEN540
630 SX=SX+3
640 IFSX>240THENSX=240
650 GOSUB800
660 IFV=2THEN930
670 IFINKEY$<>"J"THEN490ELSE620
680 GOSUB860:IFV=2THEN540
690 SX=SX-3
700 IFSX<1THENSX=1
710 GOSUB800
720 IFV=2THEN930
730 IFINKEY$<>"H"THEN490ELSE680
740 GOSUB860:IFV=2THEN540
750 SY=SY+2
760 IFSY>240THENSY=240
770 GOSUB800
780 IFV=2THEN930
790 IFINKEY$<>"N"THEN490ELSE740
800 IF PPOINT(SX+9,SY+9)=7ORPPOINT
(SX,SY+9)=7ORPPOINT(SX,SY)=7ORPPOINT(SX+9,SY)=7THEN910
810 IFPPOINT(SX+9,SY+9)=8 OR PPOINT(SX,SY+9)=8THEN930
```

```
820 PUT(LX,LY)-(LX+9,LY+9),BL,PSET
830 PUT(SX,SY)-(SX+9,SY+9),SH,PSET
840 LX=SX:LY=SY
850 RETURN
860 E=E-1:IFE<=0THENV=2:E=0
870 IFE>57THENE=57
880 LINE(11,4)-(67,4),PRESET
890 LINE(11,4)-(11+E,4),PSET
900 RETURN
910 FORX=1TO60STEP4:CIRCLE(SX+4,SY+4),X,RND(3)+5:SOUNDX,1:NE
XT
920 CLS:PRINT"    YOU CRASHED YOUR SHUTTLE":PRINT"FINAL SCOR
E->"SC:GOTO1010
930 PLAY"T10DEFFED"
940 FORGP=0TO35STEP4:IFSY+9=B(GP+1)THENPT=B(GP+3):SC=SC+PT:E
=E+10:ELSENEXT
950 CLS:PRINT"AMOUNT OF FUEL REMANING->"E
960 PRINT"POINTS FOR THIS LANDING->"PT
970 PRINT"CURRENT SCORE->"SC
980 PRINT"HIT <ENTER> WHEN READY FOR                NEXT MISS
ION"
990 SY=9
1000 IFINKEY$=CHR$(13)THENAF=1:GOTO240ELSE1000
1010 PRINT"WOULD YOU LIKE TO PLAY AGAIN Y/N"
1020 RE$=INKEY$:IFRE$=""THEN1020ELSEIFRE$="Y"THENRUNELSEEND
1030 CLS:PRINTTAB(9)"INSTRUCTIONS"
1040 PRINT:PRINTTAB(7)"DEPARTURE-PHOBOS"
1050 PRINTTAB(6)"DESTINATION-DEIMOS"
1060 PRINT:PRINTTAB(11)"ASSIGNMENT":PRINT"NAVIGATE YOUR  SHU
TTLE THROUGH ASTERIOD FIELD WATCH YOUR ENERGY  CONSUMPTION W
HILE ATTEMPTING        TO LAND ON FLAT AREAS
1070 PRINT"  USE THESE KEYS FOR THRUSTING":PRINTTAB(15)"U":P
RINTTAB(14)"H J":PRINTTAB(15)"N"
1080 RETURN
```



SPACE DUEL

By Scott McCann

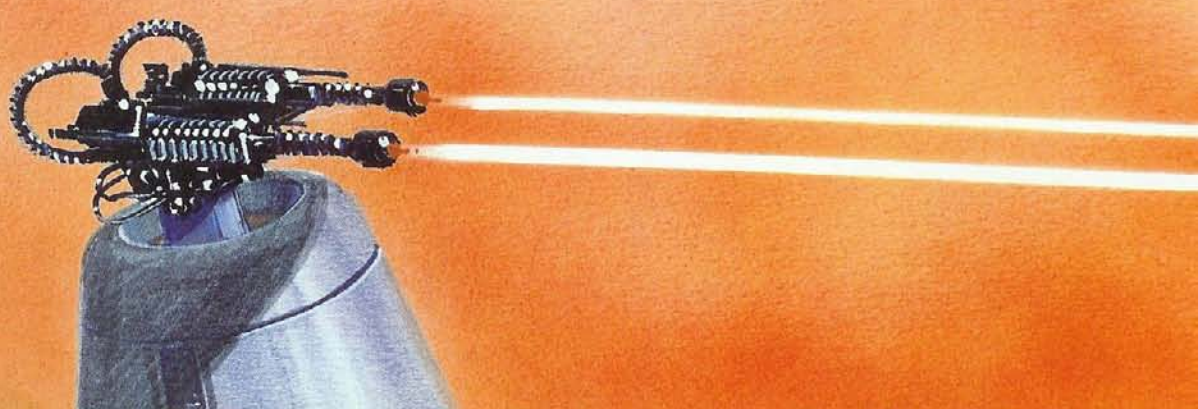


You are trapped outside your space capsule and someone is firing at you. The situation is desperate; you have only seven shots. This could be your last stand. To move up and down use the joystick or keys Q and A respectively; fire with W. If you are playing against a second player, they use the keys P and ; to move up and down; O fires. If both players run out of shots, the game is declared a draw.

VARIABLES USED:

A,B,C	– arrays to hold shapes
A,B	– left and right joystick values
A$,W$	– temporary strings
C	– computer play flag
D	– difficulty
LC,LZ	– left player laser coordinates
LU,RU,SU	– capsule, left and right player direction vectors
LX,RX,SX	– capsule, left and right player coordinates
P2	– check joystick buttons
RQ,RZ	– right player laser coordinates
SL,SR	– laser fire flags
SS	– ship shot flag

```
10 DIMA(38),B(38),C(46)
20 CLS:PRINT@10,"SPACE DUEL"
30 PRINT:PRINT" -LEFT PLAYER SHOOTS WITH 'W'   AND MOVES U/D
 WITH 'Q'&'A'."
40 PRINT:PRINT" -RIGHT PLAYER SHOOTS WITH 'O'  AND MOVES U/D
 WITH 'P'&';'."
50 PRINT:PRINT" -JOYSTICKS MOVE PLAYER U/D AND FIRE WITH BUT
TONS."
60 PRINT:PRINT"EACH PLAYER GETS 7 SHOTS. HOLD  FIRE DOWN UNT
IL LASER HEATS UP, THEN LET GO TO FIRE.":PRINT:PRINT"<HIT AN
Y KEY TO CONTINUE>";
70 IF INKEY$="" THEN 70
80 CLS:PRINT" -WHEN PLAYING AGAINST THE      COMPUTER, IT GE
TS AN UNLIMITED  NUMBER OF SHOTS."
90 PRINT:PRINT" - A DRAW IS AWARDED IN THE     EVENT THAT BO
TH PLAYERS RUN OUT OF LASER SHOTS."
100 PRINT:PRINT" -HOLDING FIRE BUTTON DOWN WILL EXPELL LASER
 FIRE A FEW  STEPS  BUT WILL NOT BE COUNTED AS A    SHOT FIR
ED.":PRINT:PRINT"<HIT ANY KEY>";
110 IF INKEY$=""THEN110
120 PLAY"T19FE04FED03FEDG02FEDGD01FEDGDC03FEDCGF02DE01DE03DE
04DE02DEFFD01C03DT255"
130 PMODE4,1:PCLS
140 DRAW"S4BM34,100;G4D3F1R1D1R3U1H2U2E1D3F1U5F1D4E1R1H1U1D3
G2R2"
150 DRAW"D1L5G2D5F2G1D1R9U1L8E1R5U1"
160 DRAW"E2U1H1R3BR2U1NR2BL2L3E1H2U1E2U3H4NE4L2H4BD16BR3D5R4
"
170 DRAW"BM222,100;F4D3G1L1D1L3U1E2U2H1D3G1U5G1D4H1L1E1U1D3F
2L2"
180 DRAW"D1R5F2D5G2F1D1L9U1R8H1L5U1"
190 DRAW"H2U1E1L3BL2U1NL2BR2R3H1E2U1H2U3E4NH4R2E4BD16BL3ND2L
4D5BR4H3R3"
200 DRAW"S8BM126,93;D5G8F2R1NF1BE1NR2U3R2BR2NR2D3E1F1U3BR2ND
3R2D3BD2BL10R1F1R6E1R1E1R1E2H8U5L2"
210 GET(30,88)-(46,134),A,G
220 GET(210,88)-(226,134),B,G
```



```
230 GET(110,88)-(148,134),C,G
240 CIRCLE(70,161),8:PAINT(70,161),5,5
250 CIRCLE(186,33),8:PAINT(186,33),5,5
260 CLS:INPUT"1 OR 2 PLAYERS";C:IF C=1 THEN INPUT"DIFFICULTY
(1-4)";D:IFD<1ORD>4THEN260
270 IFC>2ORC<1THEN 260
280 CLS:INPUT"JOYSTICKS (Y/N)";A$
290 LX=88:RX=88:SX=88:B=22:SL=0:SR=0:LC=0:RC=0
300 SCREEN1,1:IF A$<>"Y" THEN 400
310 A=JOYSTK(0):A=JOYSTK(3)
320 IF C=2 THEN B=JOYSTK(1)
330 IF A<21 THEN LU=-8
340 IF A>42 THEN LU=8
350 IF B<21 THENRU=-8
360 IF B>42THENRU=8
370 P2=PEEK(65280):IFP2=125ORP2=253THENSL=1:LQ=47:LZ=LX+25
380 IFC=2ANDP2=254ORP2=126THENSR=1:RQ=209:RZ=RX+25
390 GOSUB470:GOTO310
400 IF PEEK(339)=254THENLU=8
410 IF PEEK(339)=251THENLU=-8
420 IFC=2AND PEEK(338)=251THENRU=-8
430 IFC=2AND PEEK(341)=223THENRU=8
440 IF PEEK(345)=251 THENSL=1:LQ=47:LZ=LX+25
450 IF PEEK(345)=253THENSR=1:RQ=209:RZ=RX+25
460 GOSUB 470:GOTO400
470 IFC=2 AND RU GOSUB570
480 IF LU GOSUB550
490 IF C=1GOSUB 710
500 IF RND(10)=5 GOSUB680
510 IF LC>6AND RC>6 THEN CLS:PRINT"'DRAW.'":FORI=1TO500:NEXT
:GOTO790
520 IF SL AND LC<7 GOSUB590:GOSUB590:GOSUB590
530 IF C=2 AND  SR AND RC<7GOSUB630:GOSUB630:GOSUB630
540 RETURN
550 IF LX+LU<0 OR LX+LU>148 THEN RETURN
```

```
560 LX=LX+LU:PUT(30,LX)-(46,LX+46),A,PSET:LU=0:RETURN
570 IFRX+RU<0ORRX+RU>148THENRETURN
580 RX=RX+RU:PUT(210,RX)-(226,RX+46),B,PSET:RU=0:RETURN
590 IFSL=0 THEN RETURN ELSEIF LQ+20>256ORPPOINT(LQ+20,LZ)OR
LQ+20>110AND LZ>SX+4 AND LZ<SX+41 THEN PLAY"D":SL=0:LC=LC+1:
RETURN
600 LINE(LQ,LZ)-(LQ+20,LZ),PSET:LINE(LQ,LZ)-(LQ+20,LZ),PRESE
T
610 LQ=LQ+20:IF LQ>209AND LZ>RX+12 AND LZ<RX+34THENPLAY"AEGC
FFED":FORI=1TO10:PUT(210,RX+I)-(227,RX+46+I),B,PSET:PLAY"CFD
E":NEXT:W$="LEFT":GOTO780
620 RETURN
630 IFSR=0 THEN RETURN ELSEIFRQ-20<0OR RQ-20<148AND RZ<SX+41
 AND RZ>SX+4THEN PLAY"F":SR=0:RC=RC+1:RETURN
640 IF PPOINT(RQ-20,RZ)THEN PLAY"F":SR=0:RC=RC+1:RETURN
650 LINE(RQ,RZ)-(RQ-20,RZ),PSET:LINE(RQ,RZ)-(RQ-20,RZ),PRESE
T
660 RQ=RQ-20:IFRQ<47AND RZ>LX+12 AND RZ<LX+34 THENPLAY"DEFFC
GEA":FORI=1TO10:PUT(30,LX+I)-(47,LX+46+I),A,PSET:PLAY"FCD":N
EXT:W$="RIGHT":GOTO780
670 RETURN
680 SU=5*(2-RND(3)):IFSU=0GOTO680
690 IF SX+SU<30OR SX+SU>133 THENRETURN
700 SX=SX+SU:PUT(110,SX)-(148,SX+46),C,PSET:RETURN
710 IF LX<RX THEN RX=RX-D*2 ELSE IF LX>RX THEN RX=RX+D*2
720 IF RX<0 THEN RX=0 ELSE IF RX>148 THEN RX=148
730 GOSUB 580
740 IFSS AND RX+25>LX+8 AND RX+25<LX+26 THEN SS=0:SR=1:RQ=20
9:RZ=RX+25
750 IF SS=0  THEN FORI=1TO D:GOSUB630:NEXT
760 IF SR=0 THEN SS=1
770 RETURN
780 CLS:PRINTW$"HAND PLAYER WON.":FORI=1TO1000:NEXT
790 CLS:PRINT"PLAY AGAIN (Y/N)"
800 A$=INKEY$:IF A$="Y" THEN 130 ELSE IF A$="N"THEN CLS:PRIN
T"BYE.":END
810 GOTO800
```

ABOUT DATA STATEMENTS

Alien Attack and the other games in this book make use of DATA statements. Why DATA statements instead of variables? Because DATA statements can hold information that is not needed all at once or information that is needed only once.

The directions that the ships must move are stored in DATA statements at the end of the program. The program variables are used to hold information that is required very often; for example, the score and score counter (SC and CS).

To use DATA statements in your own programs type "DATA" followed by the information you want stored. Separate each piece by commas.

For example:
100 DATA 8,10,20,-17

Strings may be stored this way too. If a string has punctuation or spaces in it, the string must be enclosed in quotes; otherwise, the computer will not know where it ends.

For example:
105 DATA FRED,SCORE,BOX,
"CATS AND MICE",LASER

Numerical and string information may be stored in the same DATA statement (up to a total of 88 characters).

For example:
110 DATA 255,210,-22,"FUEL
SUPPLY IS ",ELEPHANT,89

If you have more information to store, simply start a new DATA statement.

To retrieve the information stored in a DATA statement, the BASIC command READ is used. To find out more about READ statements, turn to page 27 in this book.

PROTECTOR

by L. Braine

You are on an intercept mission near the surface of Neptune. Only you and your ship stand between the space colony and the attacking aliens. They come in swarms of four at a time. Can even your valiant ship keep up this deadly pace? If you can ward off the aliens for two minutes, the colony will be saved. Good luck!

VARIABLES USED:

A,E,N	– general purpose
A(1) to A(4)	– alien ship X position
A$	– general keyboard inputs
AF	– ship left or right from keyboard
AL	– alien graphics
B(1) to B(4)	– alien ship Y position
B$	– general purpose
BL, BY	– last laser position
H	– horizontal joystick position
JK	– joystick position or keyboard flag
L,Q	– laser X,Y position
LA, LB	– last
LX, LY	– last
PR, PL	– player left and right graphics
PX, PY	– player's ship X and Y positions
S	– which side is player on flag
SB	– blank graphics block
SC	– score
V	– vertical joystick position
X	– counter

```
10 CLS:PRINT@194,"PROTECTOR"
20 PRINTTAB(7)"BY L. BRAINE"
30 PRINTTAB(11)"CONVERSION BY"TAB(48)"DEREK SIMONSON
40 FORX=5TO20STEP5
50 B$=STR$(X)
60 A$="T"+B$+"03E04CECD03B04C03ABAG
70 IFPEEK(16380)=0THENPLAYA$
80 NEXT
90 PRINT:PRINT"MISSION-> INTERCEPT                WHERE-> ABOVE
 NEPTUNE          ALIENS-> COME IN SWARMS OF FOUR TIME-> YO
U HAVE TWO MINUTES     10 PTS-> HIT ALIENS IN CENTRE"
100 PRINT"CONTROLS-> KEYS(ARROWS, SPACE)              JOYSTIC
K(RIGHT)"
110 CLEAR200:PCLEAR4
120 PRINT:PRINT"PRESS FIRE BUTTON FOR JOYSTICK  HIT <ENTER>
FOR KEYBOARD"
130 E=PEEK(65280):IFE=254ORE=126THENJK=1:GOTO150ELSEIFINKEY$
=CHR$(13)THENJK=2:H=31:AF=1:GOTO150ELSE130
140 CLEAR200
150 DIM AL(5),PR(20),PL(16),BL(16),SB(5)
160 PX=20:PY=75
170 FORA=1TO4:A(A)=240:B(A)=RND(160):NEXT
180 PMODE3,1:PCLS5
190 DRAW"C7R6U2R6D2L4D6R4D2L6U2L6E3H3
200 PAINT(130,97),7,7
210 GET(128,94)-(140,104),AL,G
220 PCLS:GET(128,94)-(140,104),SB,G
230 PCLS:DRAW"C6E8F12BU4U4L4D4R4BD8D4L4U4R4BU4G12H8L4U8R4"
240 GET(125,88)-(150,113),PL,G
250 PCLS:DRAW"R4U8L4H8G12BU4R4U4L4D4BD8R4D4L4U4BU4F12E8
260 GET(108,80)-(133,103),PR,G
270 PCLS:GET(108,80)-(133,103),BL,G
280 PCLS:DRAW"BM0,191C6E5R4E4F6E6F4E9R10F9R5E6R8E3F8R5E7R3F3
E4F8E7R6F8R4F3R7E10R3F4E8R3F7E6R6F3R10E4R4F4R4E4F11E4F10
290 PAINT(20,191),6,6
300 SCREEN1,1
310 TIMER=0
320 IF TIMER>7200THEN 710
330 ON JK GOTO340,380
340 H=JOYSTK(0):V=JOYSTK(1)
350 A=PEEK(65280):IFA=254 OR A=126THENGOSUB560
360 PY=V*3
370 GOTO390
380 A$=INKEY$:IFA$=CHR$(94)THENPY=PY-10ELSEIFA$=CHR$(10)THEN
PY=PY+10ELSEIFA$=CHR$(9)THENAF=1:GOTO410:ELSE IFA$=CHR$(8)TH
ENAF=2:GOTO410ELSEIFA$=CHR$(32)THENGOSUB560
387 IF PY<1THENPY=1
```

```
390 IFPY>150THENPY=150
400 IFPY=LY THEN450
410 PUT(LX,LY)-(LX+25,LY+23),BL,PSET
420 IFH>31 ORAF=1THENPX=230:PUT(PX,PY)-(PX+25,PY+23),PL,PSET
:S=-1
430 IFH<310RAF=2THENPX=20:PUT(PX,PY)-(PX+25,PY+23),PR,PSET:S
=1
440 LX=PX:LY=PY
450 N=N+1
460 IFN>4THENN=1
470 IFS=-1THENA(N)=A(N)+(RND(15)-3)
480 IFS=1THENA(N)=A(N)-(RND(15)-3)
490 PUT(LA(N),LB(N))-(LA(N)+12,LB(N)+10),SB,PSET
500 B(N)=B(N)+(RND(14)-7)
510 IF A(N)<1 THENA(N)=240ELSEIFA(N)>240THENA(N)=1
520 IFB(N)<1 THENB(N)=1ELSEIFB(N)>160THENB(N)=160
530 PUT(A(N),B(N))-(A(N)+12,B(N)+10),AL,PSET
540 LA(N)=A(N):LB(N)=B(N)
550 GOTO320
560 Q=PY+12
570 IF S=1THENFOR L=PX+26 TO 270STEP3ELSE FOR L=PX-1 TO 1 ST
EP-3
580 IFPPOINT(L,Q)=7THEN650
590 PSET(L,Q,6)
600 PSET(BL,BY,5)
610 BL=L:BY=Q
620 NEXT
630 PSET(BL,BY,5)
640 RETURN
650 SC=SC+5
660 FORN=1TO4
665 IFL<13THENL=13
670 IF A(N)<L AND A(N)+12>L THEN B(N)=RND(160):A(N)=240:SC=S
C+10:PUT(L-12,Q-11)-(L+13,Q+12),BL,PSET:PLAY"T15CG#C#F":RETU
RN
680 NEXT
690 SOUND200,2
700 RETURN
710 FORA=1TO200STEP5
720 A$="T"+STR$(A)+"CG#C#F"
730 PLAYA$
740 NEXT
750 CLS:PRINTSTRING$(32,"=")
760 PRINT"YOUR 2 MINUTES HAS EXPIRED                    YOU SC
ORED";SC
770 PRINT"PLAY AGAIN  (Y/N)
780 A$=INKEY$:IFA$=""THEN780ELSEIFA$="Y"THENPOKE16380,
    1:RUNELSEPOKE16380,0
```

ALIEN ATTACK

Adapted by Scott McCann
by Peter Lear

You await your doom inside the crippled space station. The aliens will be attacking in clusters. There are eight different ways that they can get in. When they do come, the station's force shields will protect you from at least three blasts on each side. Then, if you're lucky, you will figure out how to operate the malfunctioning laser cannon and fight back. There are several controls that could do it – your joystick and some keys. Try T, Y, U, G, J, B, N and M. Maybe by destroying enough attackers you will be able to recharge your shield and save the space station and your life.

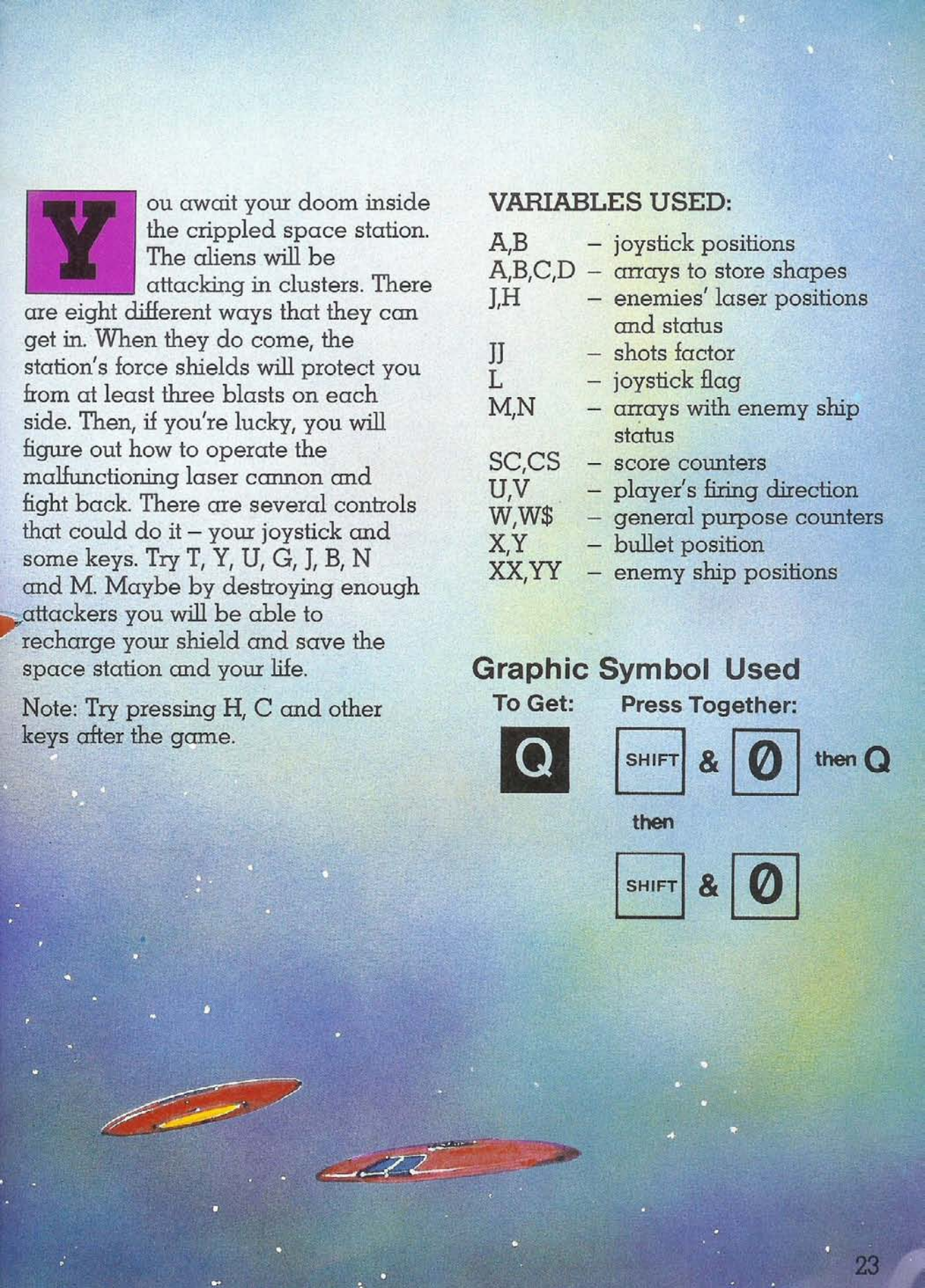
Note: Try pressing H, C and other keys after the game.

VARIABLES USED:

A,B	– joystick positions
A,B,C,D	– arrays to store shapes
J,H	– enemies' laser positions and status
JJ	– shots factor
L	– joystick flag
M,N	– arrays with enemy ship status
SC,CS	– score counters
U,V	– player's firing direction
W,W$	– general purpose counters
X,Y	– bullet position
XX,YY	– enemy ship positions

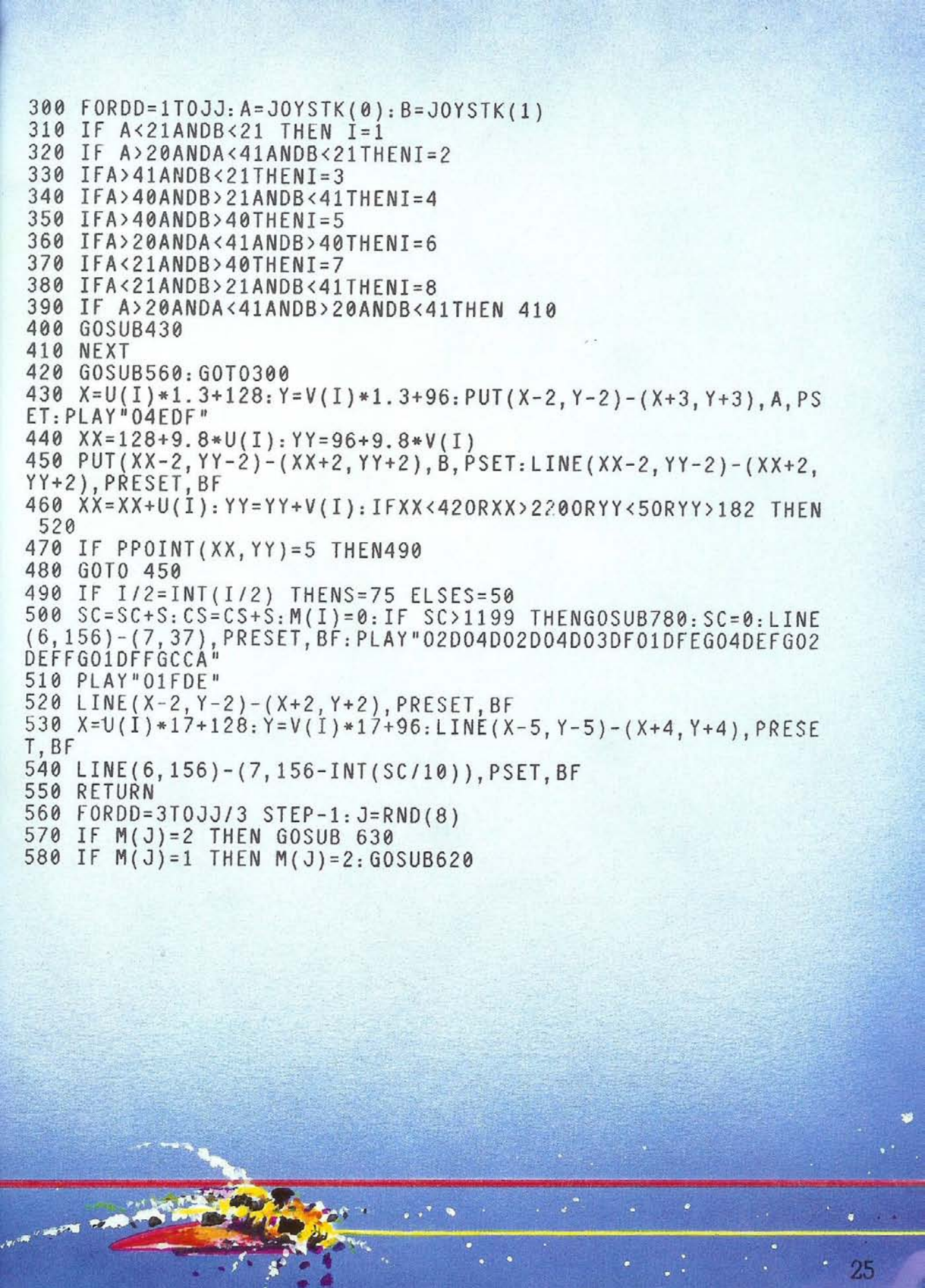
Graphic Symbol Used

To Get:	Press Together:
Q	SHIFT & 0 then Q
	then
	SHIFT & 0



```
10 'ALIEN ATTACK': BY PETER LEAR   ADAPTED FOR 16K TRS-80 EX
TENDED COLOR COMPUTER BY SCOTT MCCANN
20 DIMA(5,5),B(9,9),C(10,10),M(8),D(9,9)
30 CLS:PRINT@10,"ALIEN ATTACK"
40 PLAY"T12DEFDEFP5DEFP5DEFGGP2DEFFEEFFGFDFGEDT255T255"
50 PRINT:PRINT" - SHIELDS REPLACED WHEN        INDICATOR REA
CHES TOP.":PRINT
60 INPUT"SHOTS FACTOR (1-9)";JJ:IF JJ>9 ORJJ<1THEN60
70 RESTORE
80 XX=0:YY=0:X=0:Y=0
90 FOR J=1TO8:READU(J),V(J):NEXT
100 FOR I=1 TO 8:READN(I):NEXT
110 A$="■TYUJMNBG"
120 INPUT"(1) JOYSTICKS OR (2) KEYS";A:IF A<>1ANDA<>2THEN120
ELSECLS
130 IF A=2 THEN PRINT@0,"TO MOVE USE:":PRINT"   TYU":PRINT"
  G J":PRINT"   BNM":IF INKEY$=""THEN130
140 CLS:PRINT@233,"GOOD LUCK";
150 SC=0:CS=0:POKE65494,0:FORW=1TO8:M(W)=0:NEXT
160 PMODE 4,1:PCLS
170 DRAW"S4BM12,10":GOSUB180:GOTO190
180 DRAW"G2R4H2D2G2E2F2": GET(10,10)-(15,15),A,G:RETURN
190 DRAW"BM40,40NF9BD5NR9BD4NE9BR4NU9BR4NH9":LINE(42,42)-(47
,47),PSET,BF:GET(40,40)-(49,49),D,G
200 DRAW"BM21,20R3F1L4D1R4D1L4F1R3":GET(21,20)-(26,25),B,G
210 DRAW"BM43,40ND10R1ND10BR2ND10R1D10":GET(40,40)-(50,50),C
,G
220 PCLS:GOSUB780
230 DRAW"BM4,36S4NR5D120R5U120BR2NR7NE2F2"
240 SCREEN1,1:IFA=1THEN300
250 FORDD=1TOJJ:I=INSTR(A$,INKEY$)
260 IF I>1 THEN I=I-1:GOSUB 430
270 NEXT
280 GOSUB 560
290 GOTO250
```

```
300 FORDD=1TOJJ:A=JOYSTK(0):B=JOYSTK(1)
310 IF A<21ANDB<21 THEN I=1
320 IF A>20ANDA<41ANDB<21THENI=2
330 IFA>41ANDB<21THENI=3
340 IFA>40ANDB>21ANDB<41THENI=4
350 IFA>40ANDB>40THENI=5
360 IFA>20ANDA<41ANDB>40THENI=6
370 IFA<21ANDB>40THENI=7
380 IFA<21ANDB>21ANDB<41THENI=8
390 IF A>20ANDA<41ANDB>20ANDB<41THEN 410
400 GOSUB430
410 NEXT
420 GOSUB560:GOTO300
430 X=U(I)*1.3+128:Y=V(I)*1.3+96:PUT(X-2,Y-2)-(X+3,Y+3),A,PS
ET:PLAY"O4EDF"
440 XX=128+9.8*U(I):YY=96+9.8*V(I)
450 PUT(XX-2,YY-2)-(XX+2,YY+2),B,PSET:LINE(XX-2,YY-2)-(XX+2,
YY+2),PRESET,BF
460 XX=XX+U(I):YY=YY+V(I):IFXX<42ORXX>220ORYY<5ORYY>182 THEN
 520
470 IF PPOINT(XX,YY)=5 THEN490
480 GOTO 450
490 IF I/2=INT(I/2) THENS=75 ELSES=50
500 SC=SC+S:CS=CS+S:M(I)=0:IF SC>1199 THENGOSUB780:SC=0:LINE
(6,156)-(7,37),PRESET,BF:PLAY"O2D04D02D04D03DF01DFEG04DEFG02
DEFFG01DFFGCCA"
510 PLAY"01FDE"
520 LINE(X-2,Y-2)-(X+2,Y+2),PRESET,BF
530 X=U(I)*17+128:Y=V(I)*17+96:LINE(X-5,Y-5)-(X+4,Y+4),PRESE
T,BF
540 LINE(6,156)-(7,156-INT(SC/10)),PSET,BF
550 RETURN
560 FORDD=3TOJJ/3 STEP-1:J=RND(8)
570 IF M(J)=2 THEN GOSUB 630
580 IF M(J)=1 THEN M(J)=2:GOSUB620
```



```
590 IF M(J)=0 THEN M(J)=1
600 NEXT
610 RETURN
620 X=U(J)*17+128:Y=V(J)*17+96:PUT(X-5,Y-5)-(X+4,Y+4),D,PSET
:RETURN
630 XX=U(J)*14+128:YY=V(J)*14+96
640 I=N(J)
650 LINE(XX,YY)-(XX+U(I)/5,YY+V(I)/5),PSET:LINE(XX,YY)-(XX+U
(I)/5,YY+V(I)/5),PRESET
660 IF H THEN H=0:LINE(XX-6,YY-6)-(XX+6,YY+6),PSET,BF:LINE(X
X-6,YY-6)-(XX+6,YY+6),PRESET,BF:PLAY"01FF":RETURN
670 IF XX=128ANDYY=96THEN700
680 XX=XX+U(I)*2:YY=YY+V(I)*2:IFPPOINT(XX,YY)ORPPOINT(XX+U(I
),YY+V(I))THEN H=1
690 GOTO650
700 FORW=1TO44:SCREEN1,RND(2)-1:NEXT:SCREEN1,1
710 FORW=1TO18:PLAY"V"+STR$(W+12)+"O1C":CIRCLE(128,96),W,5:N
EXT:FORW=18TO1STEP-1:PLAY"V"+STR$(W+12)+"DC":CIRCLE(128,96),
W,0:NEXT
720 DRAW"BM125,93S4":GOSUB180:FORW=1TO100:NEXT:FORW=1TO36STE
P4:DRAW"BM125,93;S"+STR$(W):GOSUB180:NEXT
730 PCLS:CLS:FORW=1TO50:NEXT:SCREEN1,0:PRINT@0,"TIME-";INT(T
IMER/33):PRINT:PRINT"SCORE-";CS
740 IFCS>S(JJ)THENS(JJ)=CS
750 A$=INKEY$:IFA$="H"THENCLS:PRINT@10,"high scores":PRINT:F
ORW=1TO9:PRINT"LEVEL"W;TAB((16-LEN(STR$(S(W)))/2));S(W):NEXT
:FORW=1TO1000:NEXT:GOTO30
760 IF A$="C" THEN30ELSE IFA$<>"" THEN CLS:GOTO140
770 GOTO750
780 FORW=4TO12STEP4
790 DRAW"BM128,96;S"+STR$(W)+";BU15R5F10D10G10L10H10U10E10R5
"
800 NEXT:RETURN
810 DATA -5,-5,0,-5,5,-5,5,0,5,5,0,5,-5,5,-5,0
820 DATA 5,6,7,8,1,2,3,4
```

ABOUT READ STATEMENTS

To retrieve the information stored in a DATA statement, the BASIC command READ is used. It is followed by one or more variables separated by commas. READ takes the first piece of information from the first DATA statement and puts it into the variable that follows the READ statement. If there is a second variable, the next piece of information is taken from the same DATA statement and put into the second variable. This continues until all the variables following the READ statement are filled. If everything from the first DATA statement is read, the computer automatically jumps to the next one.

There must be as many pieces of information in your DATA statements as are to be READ. The variable type (string or number) must match the type of information being READ. Here is an example using the DATA statements from page 17:

```
120 READ N
```

This line reads the number 8 from the DATA statement in line 100 and puts it in the variable N.

```
130 READ A,B,C,D$,E$,F$,G$
```

This line reads all the information in line 100 and some from 105. Notice how the string variable reads string information and the number variables read numbers.

```
140 READ H$,I,J,K,L$,M$,O
```

This line reads all the remaining information in lines 105 and 110.

If information in a DATA statement does not match the type being READ, you will get a "TYPE MISMATCH" error.

When two commas follow one another without any information between them, a READ statement will take this as either an empty string or the number zero. Here is an example:

```
150 DATA 10,20,,15,GREG,,6,7
160 READ G,H,I,J,K$,L$,M,N,
```

This program gives the variables the following information:

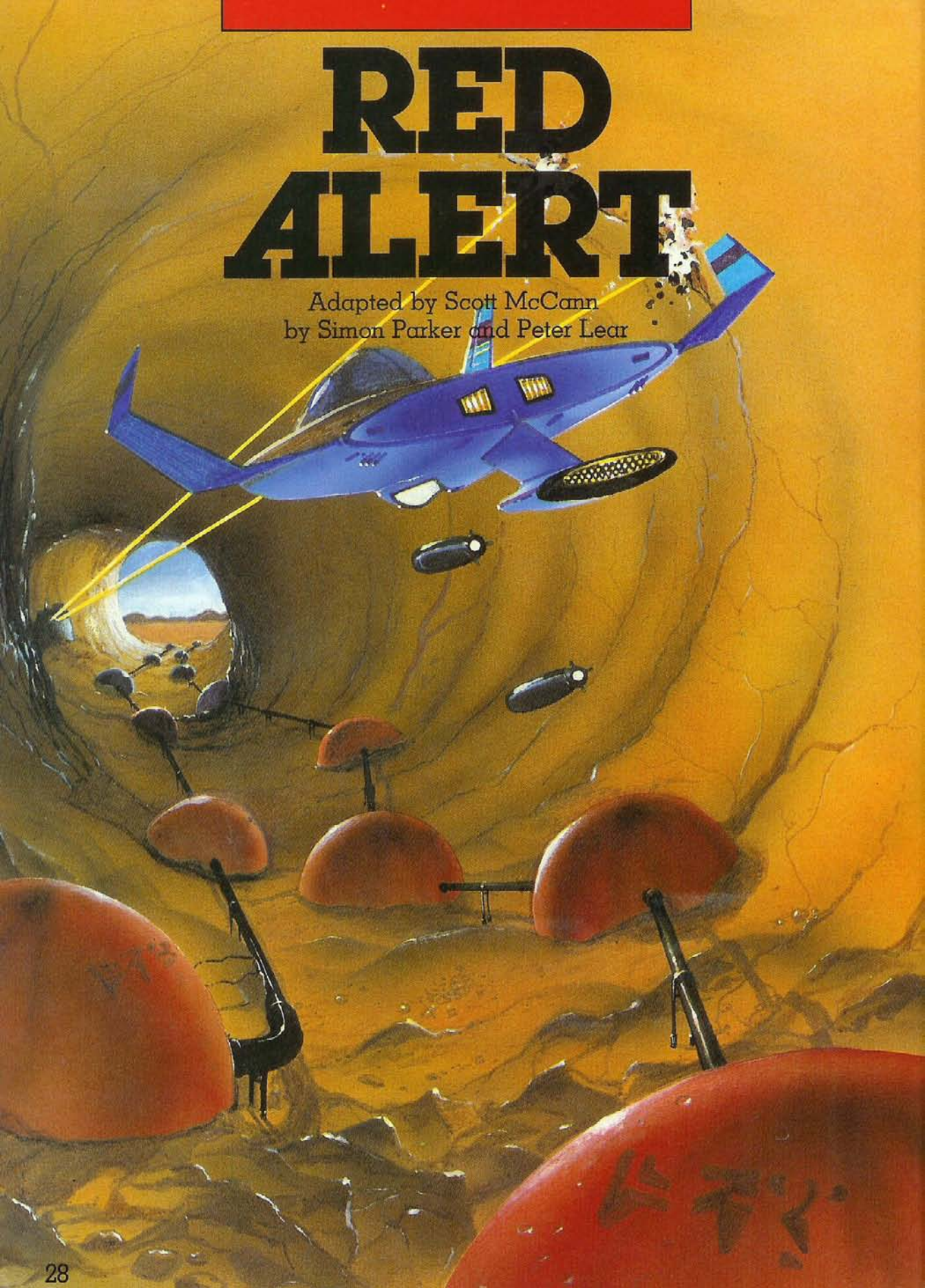
```
G=10
H=20
I=0
J=15
K$="GREG"
L$=""  This is an empty string.
M=6
N=7
```

When a DATA statement has been totally READ, another READ command will result in an "OUT OF DATA ERROR." To avoid this, always match up the number of pieces of information with the number of variables that will READ them.

The RESTORE command will allow you to use the DATA statements again.

RED ALERT

Adapted by Scott McCann
by Simon Parker and Peter Lear

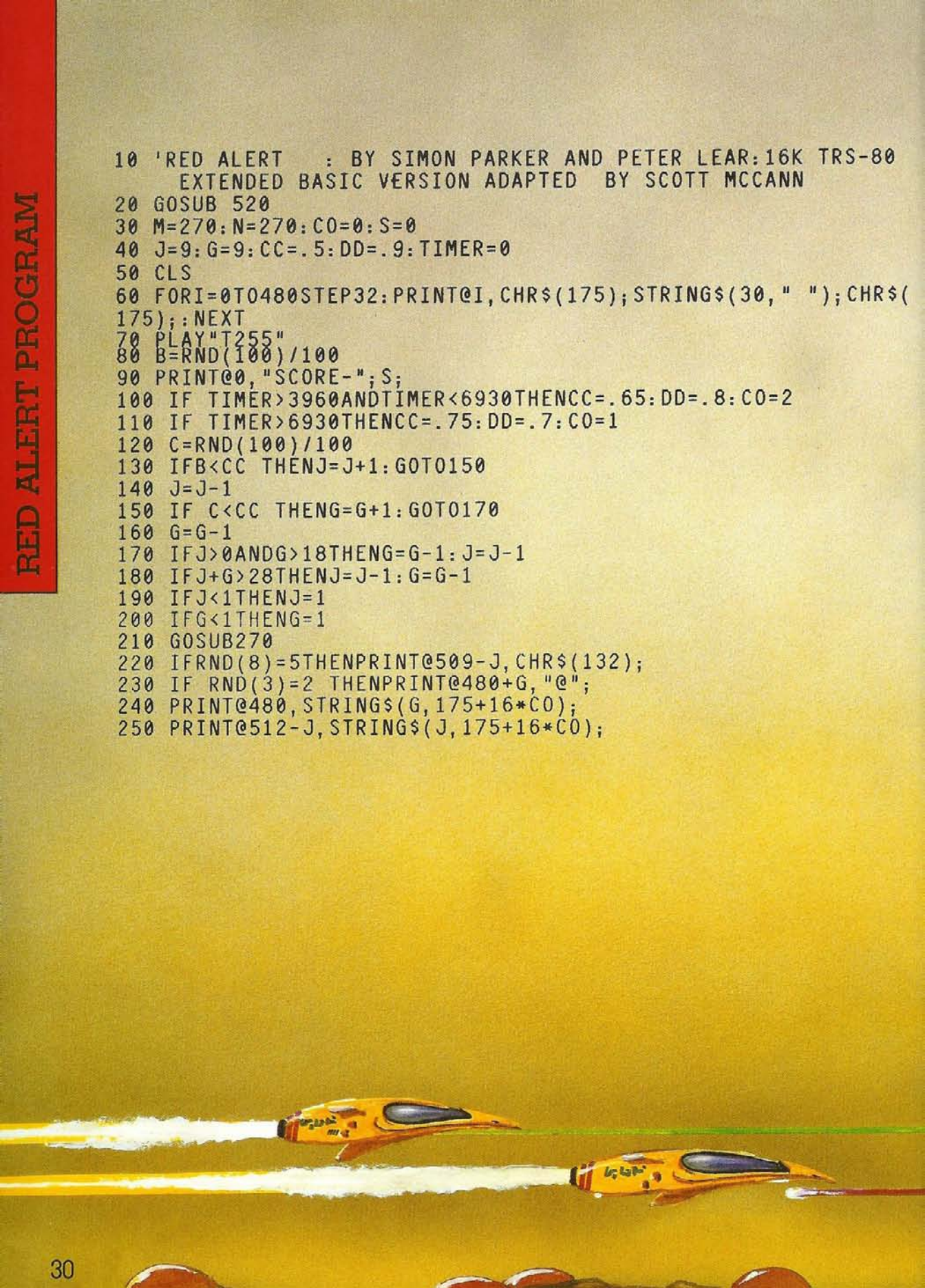


You have penetrated the enemy planet's refueling station. You must pilot your spacecraft through the cavern that houses the fuel dump. The terrain is rough and changes radically. Be very careful; the survival of your planet depends on you. Use the Arrow keys to move your ship left, right, up and down; the greater and less than signs fire left and right and the space bar fires down. Try not to hit the walls as that costs points. Good hunting!

VARIABLES USED:

B,C	– random number
C$	– keyboard input
CC,DD	– cave width multiplying factors
CO	– wall color
G,J	– right and left walls
I,IV	– vertical shot counter and direction orientation
JO	– joystick flag
K	– counter
M,N	– ship position
P,PP	– stores status of cell in front of ship
Q,Z	– joystick position
S	– score
U	– ship direction orientation

```
10 'RED ALERT    : BY SIMON PARKER AND PETER LEAR:16K TRS-80
     EXTENDED BASIC VERSION ADAPTED  BY SCOTT MCCANN
20 GOSUB 520
30 M=270:N=270:CO=0:S=0
40 J=9:G=9:CC=.5:DD=.9:TIMER=0
50 CLS
60 FORI=0TO480STEP32:PRINT@I,CHR$(175);STRING$(30," ");CHR$(
175);:NEXT
70 PLAY"T255"
80 B=RND(100)/100
90 PRINT@0,"SCORE-";S;
100 IF TIMER>3960ANDTIMER<6930THENCC=.65:DD=.8:CO=2
110 IF TIMER>6930THENCC=.75:DD=.7:CO=1
120 C=RND(100)/100
130 IFB<CC THENJ=J+1:GOTO150
140 J=J-1
150 IF C<CC THENG=G+1:GOTO170
160 G=G-1
170 IFJ>0ANDG>18THENG=G-1:J=J-1
180 IFJ+G>28THENJ=J-1:G=G-1
190 IFJ<1THENJ=1
200 IFG<1THENG=1
210 GOSUB270
220 IFRND(8)=5THENPRINT@509-J,CHR$(132);
230 IF RND(3)=2 THENPRINT@480+G,"@";
240 PRINT@480,STRING$(G,175+16*CO);
250 PRINT@512-J,STRING$(J,175+16*CO);
```



```
260 GOTO80
270 IF JO=0THENC$=INKEY$:IFC$=CHR$(94)THENU=-32ELSEIFC$=CHR$
(10)THENU=32ELSEIFC$=CHR$(8)THENU=-1ELSEIFC$=CHR$(9)THENU=1E
LSEU=0
280 IFJO=0 THENIF C$="@" THEN GOSUB370 ELSEIF C$=" "THENGOSU
B420
290 IFJO=1 THENQ=JOYSTK(0):Z=JOYSTK(1):IF(Z<21ANDQ>21ANDQ<42
)THENU=-32ELSEIF(Z>42ANDQ>21ANDQ<42)THENU=32ELSEIFQ<21THENU=
-1ELSEIFQ>42THENU=1ELSEU=0
300 IFJO=1THENIF PEEK(65280)=126ORPEEK(65280)=254THENGOSUB42
0 ELSE IF INKEY$=" "THEN GOSUB370
310 IFM+U<64ORM+U>511THENM=M-U
320 M=M+U:P=PEEK(1024+M):PP=PEEK(1025+M)
330 PRINT@N-32,"  ";:PRINT@N-64,"  ";:PRINT@M-32,CHR$(132)CH
R$(136);:PRINT@M,CHR$(139)CHR$(135);
340 IFP=139ORPP=135 THEN 360
350 IFP<>96ORPP<>96 THEN PLAY"O3":FORK=1TO10:PRINT@M-32,CHR$
(136+K)CHR$(132+K);:PLAYSTR$(K):NEXT:GOTO490
360 N=M:RETURN
370 I=M-33
380 P=PEEK(I+1024):IFP=94THENS=S+250
390 IF P=64THENS=S+200:PRINT@I,CHR$(35)" ";:PLAY"O3DEFGG":RE
TURN
400 IF P=175ORP=191ORP=207 THEN S=S-50:PLAY"O4DO1DO3F":PRINT
@I,CHR$(134)" ";:PRINT@I,CHR$(137);:RETURN
410 PRINT@I,CHR$(60)" ";:I=I-1:GOTO380
420 I=M
```

```
430 P=PEEK(I+1024):PP=PEEK(I+1025):IFP=132ORPP=132THENS=S+50
:PLAY"O1DFDDG"
440 IFP=64ORPP=64THENS=S+200:PLAY"ABCDE"
450 IFP=175+16*CO ORPP=175+16*CO THENRETURN
460 PRINT@I,CHR$(133)CHR$(138);:PRINT@I,"  ";
470 I=I+32:IFI>M+150ORI>511THEN RETURN
480 GOTO430
490 FORK=1TO90:NEXT:CLS:PRINT@0,"YOUR SCORE WAS:";S
500 PRINT:INPUT"TRY AGAIN (Y/N)";AA$:IF AA$="Y"THENGOTO30
510 CLS:PRINT"BYE.":END
520 CLS:PRINT@12,"RED ALERT"
530 PRINT@192,"";:INPUT"(1) JOYSTICK OR (2) KEYBOARD";A:IF A
=1THENJO=1:GOTO630
540 IF A=2 THEN JO=0:GOTO560
550 GOTO530
560 CLS:PRINT"TO MOVE USE:":PRINT
570 PRINT"      UP ARROW"
580 PRINT"LEFT ARROW  RIGHT ARROW"
590 PRINT"     DOWN ARROW"
600 PRINT:PRINT"    @       FIRES TO THE LEFT":PRINT"<SPACE
BAR> FIRES DOWNWARD":
610 IF INKEY$<>""THENRETURN ELSE GOTO610
620 SOUND44,44
630 CLS:PRINT"   USE JOYSTICK TO MOVE UP,DOWN,RIGHT AND LEFT
."
640 PRINT"   USE JOYSTICK BUTTON TO FIRE  DOWNWARD AND <SPAC
E BAR> TO FIRELEFT."
650 GOTO610
```

INVASION

Adapted by Derek Simonson
by Alex Kiernan and Peter Lear

The future of your planet is in your hands. Yours is the last alpha base left; everything else has been destroyed by the Astrol Fleet. You can still defeat the warring aliens, but it will take skill and cunning on your part. Only one multi-directional laser is still operational, but due to a computer malfunction, you cannot move the laser and fire it at the same time. Z moves your base left, X moves it right; , moves your turret left and . moves it right. Fire your laser with the space bar. Remember, existence as you know it depends on you. Good luck!

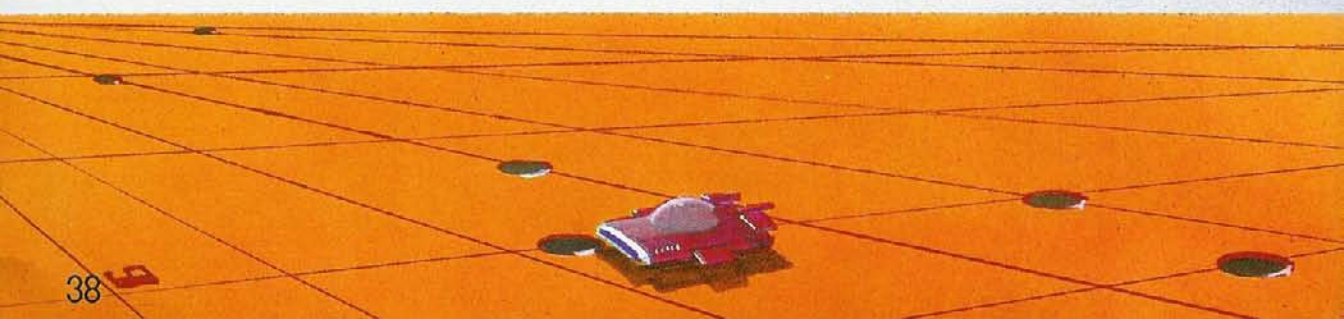
VARIABLES USED:

A$	– keyboard inputs
AL	– alien ship graphics
BX,BY	– last missile positions
CH	– character being printed
EB	– erase block graphics
GP,NM,A,E	– general purpose counters
I	– check fire button
IX,IY	– invaders' positions
JK	– joystick or keyboard flag
LB	– last player position
LE	– joystick 0
LX,LY	– last invader positions
MC	– name counter
MX,MY	– missile Y,Y position
PB	– player X position
S1,S2,S3	– ship with turret graphics
SX,SY	– get positions
TP	– turret position
V	– difficulty level
VO	– joystick 1
X	– marksmanship level

```
10 DIM S2(5),S1(5),S3(5),AL(5),EB(6),NN(5)
20 CLS:PRINT:PRINT
30 FORMC=0TO9
40 READCH
50 FORGP=22TO12STEP-1
60 DATA33,73,78,86,65,83,73,79,78,33
70 POKEGP+1054+MC,CH
80 IFMC=9THENPOKEGP+1055+MC,96
90 NEXT:NEXT
100 PRINT:PRINTTAB(8)"BY ALEX KIERNAN"
110 PRINT"  CONVERSION BY DEREK SIMONSON
120 PRINT:PRINT"ENTER:   1-->FOR KEYBOARD                    2--
>FOR RIGHT JOYSTICK"
130 GOSUB910:JK=VAL(A$):IFJK<1ORJK>2THEN130
140 PRINT:PRINT"      INSTRUCTIONS(Y/N)"
150 GOSUB910:IFA$="Y"THEN160ELSE300
160 CLS:PRINTTAB(12)"!INVASION!"
170 PRINT:PRINT" YOU ARE IN CONTROL OF THE SOLE   REMAINING
ALPHA BASE  ON THE        TERRAN PLANET EARTH        IT IS Y
OUR JOB TO PREVENT THE     WARRING BATTLE FLEET FROM
  INVADING THE PLANET
180 PRINT"  DUE TO PREVIOUS ATTACKS THE     COMPUTER HAS MAL
FUNCTIONED     DISABLING MOVEMENT OF THE BASE     AND GUN AT
 THE SAME TIME      *HIT ENTER  TO SEE CONTROLS*
190 GOSUB 910
200 CLS
210 ON JK GOTO220,250
220 PRINTTAB(7)"<<<<<CONTROLS>>>>>"
```

```
230 PRINTTAB(6)"BASE          TURRET          LEFT RIGHT
LEFT RIGHT        Z     X          ,                    SPACE
-FIRES
240 GOTO300
250 PRINT@43,"JOYSTICK
260 PRINT:PRINTTAB(12)"TURRET
270 PRINTTAB(44)" LEFT
280 PRINT"MOVE   LEFT MIDDLE RIGHT
290 PRINTTAB(12)" RIGHT
300 PRINT:PRINT"          DIFFICULTY                    1:HA
RD - 9:EASY          "
310 GOSUB910
320 V=VAL(A$):IFV<1 OR V>9THEN310
330 PRINT"WHAT'S YOUR MARKSMANSHIP RATING       1:EXCELLENT
- 5:POOR
340 GOSUB910
350 X=VAL(A$):IFX<1ORX>5THEN340
360 IX=RND(230)+10:IY=11:LX=IX:LY=IY:PB=RND(230)+10:LB=PB:NN
=9:TP=2
370 PMODE3,1:PCLS5
380 SX=21:SY=20
390 DRAW"BM20,20E5R3U5D5R2F5"
400 GET(SX,SY)-(SX+14,SY-10),S2,G
410 PCLS:DRAW"BM20,20E5R3H5F5R2F5"
420 GET(SX,SY)-(SX+14,SY-10),S1,G
430 PCLS:DRAW"BM20,20E5R2E5G5R3F5":GET(SX,SY)-(SX+14,SY-10),
S3,G:PCLS
440 PCLS:DRAW"BM20,20C6E2R9L9E3R5F5"
```

```
450 PAINT(26,16),6,6:GET(SX,SY)-(SX+15,SY-10),AL,G
460 PCLS:GET(SX,SY)-(SX+15,SY-10),EB
470 PCLS:SCREEN1,1
480 FORA=1TO255STEP9
490 CIRCLE(A,192),15,8
500 NEXT
510 FORA=120TO1+(V*X)STEP-1:PSET(RND(240)+10,RND(165),7):NEX
T
520 IX=IX+RND(20)-10
530 IY=IY+RND(9)-INT(V/2)
540 IFIX<10THENIX=15ELSEIFIX>240THENIX=238
550 IFIY<10THENIY=10
560 PUT(LX,LY)-(LX+15,LY+10),EB,PSET
570 PUT(IX,IY)-(IX+15,IY+10),AL,PSET
580 LX=IX:LY=IY
590 IFIY>162THEN870
600 ON JK GOTO610,650
610 A$=INKEY$
620 IFA$=""THEN520
630 IFA$="Z"THENPB=PB-V ELSEIFA$="X"THENPB=PB+V ELSEIFA$=","
THENTP=TP-1ELSEIFA$="."THENTP=TP+1ELSEIFA$=CHR$(32)THENLB=PB
:GOSUB750
640 GOTO680
650 I=PEEK(65280):IFI=126ORI=254THENLB=PB:GOSUB750
660 LE=JOYSTK(0):VO=JOYSTK(1):IFLE<22 THEN PB=PB-V ELSE IF L
E>41 THENPB=PB+V
670 IF VO<22THENTP=1ELSEIFVO<44THENTP=2ELSETP=3
680 IFTP=4THENTP=3ELSEIFTP=0THENTP=1
```



```
690 IFPB<10THENPB=10ELSEIFPB>240THENPB=240
700 PUT(LB,165)-(LB+14,175),EB,PSET
710 ON TP GOSUB820,830,840
720 LB=PB
730 IF NN=0THEN850
740 GOTO520
750 IFTP=1THENMX=PB+2ELSEIFTP=2THENMX=PB+8ELSEMX=PB+12
760 MY=165
770 IFMX<10RMY<10RMX>255THENRETURN
780 IF MY<IY THENRETURN
790 IF MY<IY+10 AND MY>IY THENA=PPOINT(MX-(TP*4-8),MY+4):MC=
PPOINT(MX-(TP*2-4),MY+2):E=PPOINT(MX,MY):IFE=7ORMC=7ORA=7THE
NRETURNELSEIFE=6ORMC=6ORA=6THEN810
800 PSET(MX,MY,8):PSET(BX,BY,5):BX=MX:BY=MY:MX=MX+(TP*6-12):
MY=MY-6:GOTO770
810 NN=NN-1:IX=RND(230)+10:IY=15:SCREEN1,0:PLAY"T30V10BV20CV
30D":SCREEN1,1:RETURN
820 PUT(PB,165)-(PB+14,175),S1,PSET:RETURN
830 PUT(PB,165)-(PB+14,175),S2,PSET:RETURN
840 PUT(PB,165)-(PB+14,175),S3,PSET:RETURN
850 PLAY"ABCDEF"
860 CLS:PRINT"   YOU HAVE DESTROYED ALL THE      ALIENS AND
HAVE SAVED EARTH":GOTO890
870 FORA=1TO10:SCREEN1,RND(2)-1:PLAY"T50ABCDEFG":NEXT
880 CLS:PRINTTAB(7);"!!!!YOUR DEAD!!!!          -NUMBER OF IN
VADERS DESTROYED-":PRINTTAB(14)9-NN
890 PRINT"    ------>PLAY AGAIN<------"
900 GOSUB910:IFA$="Y"THEN300ELSEEND
910 A$=INKEY$:IFA$=""THEN910ELSERETURN
```

DESIGN YOUR OWN GAME

Every game starts in the same place... in someone's head. The idea is then put down on paper. All the features the game will have are written down. Pictures of the various characters and backgrounds are drawn. Every rule and aspect of the game is included in this paper plan.

The next step is to put all this information into an order of events. On another piece of paper shapes are drawn and each event of the game is put in a box, circle or diamond. Then, with each figure a brief note of purpose is made. The name of this series of shapes and notes is a flowchart.

Every event in the flowchart is a small program in itself. These small programs are commonly called subroutines. Breaking all the events into subroutines makes the task of programming the game much easier. Tracing a flaw in a particular subroutine is easier than tracking one down in a long, unco-ordinated program.

Quite often subroutines can be used more than once. They can even be transferred from one game to another. There is no point in designing a new subroutine to examine which way a player has moved the joystick for every game using a joystick. By using some of the same subroutines from game to game, a programmer will save himself or herself a lot of time.

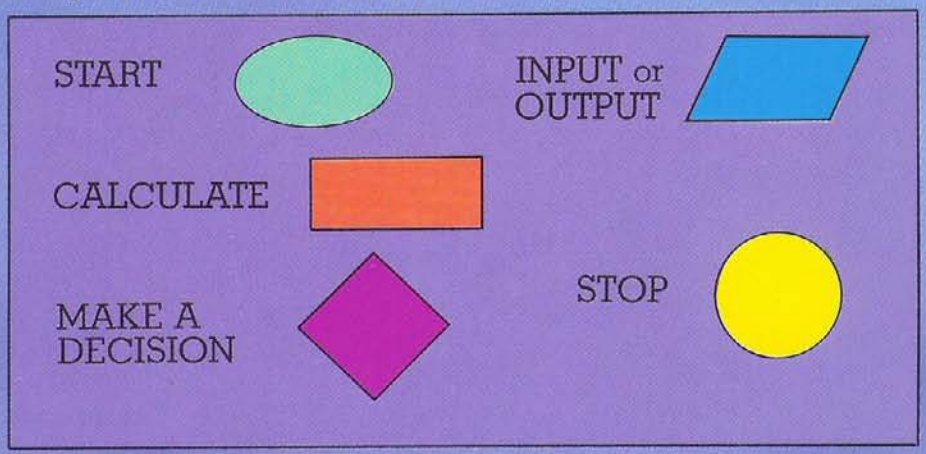
Here are some guidelines to follow when designing a game:

1. Write down your ideas.	2. Draw up a flowchart.
3. Use subroutines.	4. Use the same subroutines where possible.

The game SPACE DUEL started with these ideas:

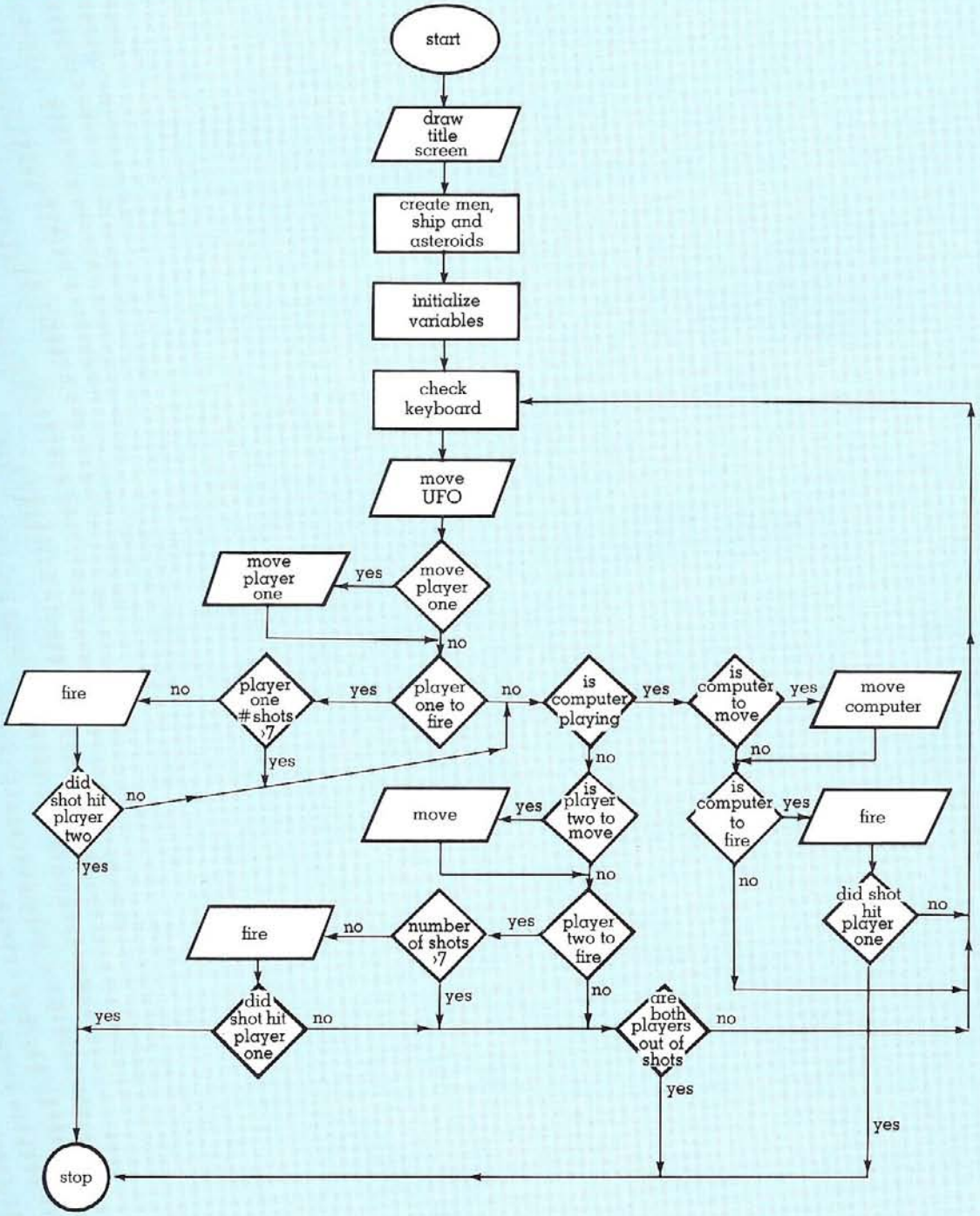
— two players will be dueling in space
— each player can move up and down
— each player has seven shots
— some objects will be placed on the screen to provide temporary protection.
— there will be a one-player option to play the computer
— the computer will have unlimited shots

These ideas were then summarized in a flowchart. The shapes in the flowchart mean:



Here is a flowchart for

SPACE DUEL



Here is a verbal description of how the game SPACE DUEL works:

Line numbers	Operation
10	build arrays
20 - 110	instructions
120	play music
130 - 250	set up and draw graphics
260 - 280	input game variables
290	set up some variables
300	branch to line 400 if using keyboard
310 - 390	joystick check
400 - 460	keyboard check
470 - 540	main loop
550 - 560	move left man
570 - 580	move right man
590 - 620	left man shoots
630 - 670	right man shoots
680 - 700	move capsule
710 - 770	computer control loop
780 - 810	end of game

By examining the ideas, the flowchart, the listing description and the program listing itself, you will be able to see how Space Duel fits together. You may follow these same steps when creating games of your own. **HAVE FUN!**

Another title from this series:

FANTASTIC GAMES FOR THE TRS-80

will give you hours of computer fun and entertainment.

RADIO SHACK DIVISION, TANDY ELECTRONICS LIMITED

CANADA: BARRIE, ONTARIO L4M 4W5
U.S.A.: FORT WORTH, TEXAS 76102

TANDY CORPORATION

AUSTRALIA	U.K.	BELGIUM
91 KURRAJONG AVENUE MOUNT DRUITT, N.S.W. 2770	BILSTON ROAD, WEDNESBURY WEST MIDLANDS WS10 7JN	PARC INDUSTRIEL DE NANINNE 5140 NANINNE

1054 ISBN 0-88625-081-1 Printed in Hong Kong