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.

Copyright© 1984 by Hayes Publishing Ltd. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without permission from the publisher.

ISBN 0-88625-081-1

TRS-80 is a registered trademark of Radio Shack, Division, Tandy Corporation.
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>MOONSHUTTLE</td>
<td>6</td>
</tr>
<tr>
<td>SPACE DUEL</td>
<td>12</td>
</tr>
<tr>
<td>About DATA Statements</td>
<td>17</td>
</tr>
<tr>
<td>PROTECTOR</td>
<td>18</td>
</tr>
<tr>
<td>ALIEN ATTACK</td>
<td>22</td>
</tr>
<tr>
<td>About READ Statements</td>
<td>27</td>
</tr>
<tr>
<td>RED ALERT</td>
<td>28</td>
</tr>
<tr>
<td>INVASION!</td>
<td>34</td>
</tr>
<tr>
<td>Design Your Own Game</td>
<td>40</td>
</tr>
</tbody>
</table>
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;

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 'T' (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. FOR I=0 TO 2: POKE &H3EBD+I, 18: NEXT: I=&H3F1E
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.

If something goes wrong, you can either turn the TRS-80 off and on to begin again, or press the RESET button.

And remember, there is nothing that you can enter into the TRS-80 by
MOON SHUTTLE
Adapted by Derek Simonson
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 FOR GP=0 TO 35: READB(GP): NEXT
60 DATA 28, 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 FORM C=1 TO 11
90 READ CH
100 FOR GP=22 TO 10 STEP -1
110 DATA 77, 79, 79, 78, 83, 72, 85, 84, 84, 76, 69
120 POKE GP+1054+MC, CH
130 IF MC=1 THEN POKE GP+1055+MC, 96
140 NEXT: NEXT
150 PRINT: PRINT: PRINT" BY P. A. ROBERTS"
160 PRINT" CONVERSION BY DEREK SIMONSON"
170 PRINT: PRINT TAB(6) "INSTRUCTIONS(Y/N)"
180 RES=INKEY$: IF RES="" THEN 180 ELSE IF RES="Y" THEN GOSUB 1030
190 PMODE 3, 1: PCLS
200 DRAW"BM20, 20C6R2F2D4G2BR2BD1U1BL2L2BL2BD1U1BR2H2U4E2
210 PAINT(22, 22), 6, 6
220 GET(17, 20)-(26, 29), SH, G
230 PUT(50, 50)-(59, 59), SH, PSET
240 PCLS: GET(16, 20)-(25, 29), BL, G
250 COLOR 8, 5: FOR GP=0 TO 35 STEP 4: LINE (B(GP), B(GP+1))-(B(GP+2), B
(GP+1)), PSET: NEXT
260 DRAW"BM0,2S4C6R4L4D2R2L2D2BR9BD1R60U6L60D6"
270 DRAW"BM0,100C7R6F8D12F12D16R20U28H8U12E16R3E12F3D4F8D4G1
U8G4U6E8R
280 PAINT(1, 101),7,7
290 IF AF=1 THEN 320
300 PRINTTAB(10)"LEVEL (1-9)";
310 RES=INKEY$:IF RES="" THEN 310 ELSE L=VAL(RES)
320 PMODE0,5:PCLS:SCREEN1,1:FOR GP=5 TO 1 STEP-1
330 IF GP=5 THEN AS="BR2L2D2R2D2L2"
340 IF GP=4 THEN AS="D2R2U2D4"
350 IF GP=3 THEN AS="R2D2L1R1D2L2"
360 IF GP=2 THEN AS="R2D2L2D2R2"
370 IF GP=1 THEN AS="BR2D4"
380 PCLS:DRAW"BM127,95S12;XAS;"
390 SOUND20*GP,5
400 NEXT
410 GP=RND(230)+10
420 LX=GP:LY=SY
430 PMODE3,1:SCREEN1,1
440 FOR SX=240 TO 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 FOR GP=1 TO 2*L:CIRCLE(RND(255),RND(70)),4,7:NEXT
490 RES=INKEYS
500 IF RES$="U" THEN 560
510 IF RES$="J" THEN 620
520 IF RES$="H" THEN 680
530 IF RES$="N" THEN 740
540 SY=SY+1: GOSUB 800
550 GOTO 490
560 GOSUB 860: IF V=2 THEN 540
570 SY=SY-3
580 IF SY<9 THEN SY=9
590 GOSUB 800
600 IF V=2 THEN 930
610 IF INKEY$<>"U" THEN 490 ELSE 560
620 GOSUB 860: IF V=2 THEN 540
630 SX=SX+3
640 IF SX>240 THEN SX=240
650 GOSUB 800
660 IF V=2 THEN 930
670 IF INKEY$<>"J" THEN 490 ELSE 620
680 GOSUB 860: IF V=2 THEN 540
690 SX=SX-3
700 IF SX<1 THEN SX=1
710 GOSUB 800
720 IF V=2 THEN 930
730 IF INKEYS$<>"H" THEN 490 ELSE 680
740 GOSUB 860: IF V=2 THEN 540
750 SY=SY+2
760 IF SY>240 THEN SY=240
770 GOSUB 800
780 IF V=2 THEN 930
790 IF INKEYS$<>"N" THEN 490 ELSE 740
800 IF PPOINT(SX+9, SY+9) = 7 OR PPOINT(SX, SY+9) = 7 OR PPOINT(SX+9, SY) = 7 THEN 910
810 IF PPOINT(SX+9, SY+9) = 8 OR PPOINT(SX, SY+9) = 8 THEN 930
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:NEXT
920  E=E-0:IFE<=0THENV=2:E=0
930  IFN>57THENF=57
940  LINE(11,4)-(67,4),PRESET
950  LINE(11,4)-(11+E,4),PSET
960  RETURN
970  FORX=1TO60STEP4:CIRCLE(SX+4,SY+4),X,RND(3)+5:SOUNDX,1:NEXT
980  E=E-0:IFE<=0THENV=2:E=0
990  IFN>57THENF=57
1000  RETURN
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 INKEYS="" 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 INKEYS="" THEN 110 
120 PLAY"T19FE04FEDO3FEDG02FEDGD01FEDGDC03FEDCGF02DE01DEO3DE
04DE02DEFFD01C03DT255" 
130 PMODE4,1:PCLS 
140 DRAW"S4BM34,100;G4D3F1R1D1R3U1H2U2E1D3F1U5F1D4E1R1H1U1D3
G2R2" 
150 DRAW"D1L5G2D5F2G1D1R9U1L8E1R5U1" 
160 DRAW"E2U1H1R3BR2U1NR2BL2L3E1H2U1E2U3H4NE4L2H4BD16BR3D5R4
E3" 
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:IF D<10RD>4THEN260
270 IF C>20RC<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 SCREEN 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 THEN RU=-8
360 IF B>42 THEN RU=8
370 P2=PEEK(65280):IF P2=125OR P2=253 THEN SL=1:LQ=47:LZ=LX+25
380 IF C=2 AND P2=2540 THEN SR=1:RQ=209:RZ=RX+25
390 GOSUB 470:GOTO 310
400 IF PEEK(339)=254 THEN LU=8
410 IF PEEK(339)=251 THEN LU=-8
420 IF C=2 AND PEEK(341)=223 THEN RU=-8
430 IF C=2 AND PEEK(345)=253 THEN SL=1:LQ=47:LZ=LX+25
440 IF PEEK(345)=251 THEN SR=1:RQ=209:RZ=RX+25
450 GOSUB 470:GOTO 400
460 IF C=2 AND RU GOSUB 570
470 IF LU GOSUB 550
480 IF SL AND LC<7 GOSUB 590:GOSUB 590:GOSUB 590
490 IF C=1 GOSUB 710
500 IF RND(10)=5 GOSUB 680
510 IF LC>6 AND RC>6 THEN CLS:PRINT "DRAW.";FOR I=1 TO 500:NEXT
520 IF SL AND RC<7 GOSUB 590:GOSUB 590:GOSUB 590
530 IF C=2 AND SR AND RC<7 GOSUB 630:GOSUB 630:GOSUB 630
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<00RRX+RU>148THENRETURN
580 RX=RX+RU,PUT(210,RX)-(226,RX+46),B,PSET:RU=0:RETURN
590 IFSL=0 THEN RETURN ELSE IF LQ+20>256ORPOINTER(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),PRESER
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:WS="LEFT":GOT0780
620 RETURN
630 IFSR=0 THEN RETURN ELSEIFRQ-20<00R RQ-20<148AND RZ<SX+41
AND RZ>SX+4 THEN 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),PRESER
660 RQ=RQ-20:IFRQ<47AND RZ<LX+12 AND RZ<LX+34 THENPLAY"DFG
E"A":FORI=1TO10:PUT(30,LX+I)-(47,LX+46+I),A,PSET:PLAY"FCR":N
EXT:WS="RIGHT":GOT0780
670 RETURN
680 SU=5*(2-RND(3))):IFSU=0GOT0680
690 IF SX+SU<300OR 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:PRINTWS$"HAND PLAYER WON. ":FORI=1TO1000:NEXT
790 CLS:PRINT"PLAY AGAIN (Y/N)"
800 A$=INKEY$:IF A$="Y" THEN 130 ELSE IF A$="$"THEN CLS:PRIN
T"BYE. ":END
810 GOT0800
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.
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
- **LQ** - 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
CLS: PRINT@194, "PROTECTOR"
PRINTTAB(7) "BY L. BRAINE"
PRINTTAB(11) "CONVERSION BY" TAB(48) "DEREK SIMONSON"
FOR X = 5 TO 20 STEP 5
B$ = STR$(X)
A$ = "T" + B$ + "03E04CECD03B04C03ABAG"
IF PEEK(16380) = 0 THEN PLAYAS
NEXT
PRINT: PRINT "MISSION-> INTERCEPT WHERE-> ABOVE NEPTUNE ALIENS-> COME IN SWARMS OF FOUR TIME-> YOU HAVE TWO MINUTES 10 PTS-> HIT ALIENS IN CENTRE"
PRINT "CONTROLS-> KEYS(ARROWS, SPACE) JOYSTICK K(RIGHT)"
CLEAR200: PCLEAR4
PRINT: PRINT "PRESS FIRE BUTTON FOR JOYSTICK HIT <ENTER> FOR KEYBOARD"
E = PEEK(65280): IF E = 2540 THEN JK = 1: GOTO150 ELSEIF INKEYS = CHR$(13) THEN JK = 2: H = 31: AF = 1: GOTO150 ELSE 130
CLEAR200
DIM AL(5), PR(20), PL(16), BL(16), SB(5)
PX = 20: PY = 75
FOR A = 1 TO 4: A(A) = 240: B(A) = RND(160): NEXT
PMODE3, 1: PCLS5
DRAW "C7R6U2R6D2L4D6R4D2L6U2L6E3H3"
PAINT(130, 97), 7, 7
GET(128, 94) - (140, 104), AL, G
PCLS: GET(128, 94) - (140, 104), SB, G
PCLS: DRAW "C6E8F12BU4U4L4D4R4BD8D4L4U4R4BU4G12H8L4U8R4"
GET(125, 88) - (150, 113), PL, G
PCLS: DRAW "R4U8L4H8G12BU4R4U4L4D4R4BD8R4D4L4U4BU4F12E8"
GET(108, 80) - (133, 103), PR, G
PCLS: GET(108, 80) - (133, 103), BL, G
PCLS: DRAW "BM0, 191C6E5R4E4F6E6F4E9R10F9R5E6R8E3F8R5E7R3F3E4F8E7R6F8R4F3R7E10R3F4E8R3F7E6R6F3R10E4R4F4R4E4F11E4F10"
PCLS: GET(20, 191), 6, 6
PAINT(20, 191), 6, 6
SCREEN 1, 1
TIMER = 0
IF TIMER > 7200 THEN 710
ON JK GOTO340, 380
H = JOYSTK(0): V = JOYSTK(1)
A = PEEK(65280): IF A = 254 OR A = 126 THEN GOSUB 560
PY = V * 3
GOTO390
A$ = INKEY$: IF A$ = CHR$(94) THEN PY = PY - 10 ELSE IF A$ = CHR$(10) THEN PY = PY + 10 ELSE IF A$ = CHR$(9) THEN NAF = 1: GOTO 410: ELSE IF A$ = CHR$(8) THEN ENAF = 2: GOTO 410 ELSE IFA$ = CHR$(32): GOSUB 560
IF PY < 1 THEN PY = 1
390 IF PY > 150 THEN PY = 150
400 IF PY = LY THEN 450
410 PUT (LX, LY) - (LX + 25, LY + 23), BL, PSET
420 IF H > 31 OR A(N) = 1 THEN PX = 230: PUT (PX, PY) - (PX + 25, PY + 23), PL, PSET
S = -1
430 IF H < 31 OR A(N) = 2 THEN PX = 20: PUT (PX, PY) - (PX + 25, PY + 23), PR, PSET: S = 1
440 LX = PX: LY = PY
450 N = N + 1
460 IF N > 4 THEN N = 1
470 IF S = -1 THEN A(N) = A(N) + (RND(15) - 3)
480 IF S = 1 THEN A(N) = A(N) - (RND(15) - 3)
490 PUT (LA(N), LB(N)) - (LA(N) + 12, LB(N) + 10), SB, PSET
500 B(N) = B(N) + (RND(14) - 7)
510 IF A(N) < 1 THEN A(N) = 240 ELSE IF A(N) > 240 THEN A(N) = 1
520 IF B(N) < 1 THEN B(N) = 1 ELSE IF B(N) > 160 THEN B(N) = 160
530 PUT (A(N), B(N)) - (A(N) + 12, B(N) + 10), AL, PSET
540 LA(N) = A(N): LB(N) = B(N)
550 GOTO 320
560 Q = PY + 12
570 IF S = 1 THEN FOR L = PX + 26 TO 270 STEP 3 ELSE FOR L = PX - 1 TO 1 STEP -3
580 IF PPOINT(L, Q) = 7 THEN 650
590 PSET (L, Q, 6)
600 PSET (BL, BY, 5)
610 BL = L: BY = Q
620 NEXT
630 PSET (BL, BY, 5)
640 RETURN
650 SC = SC + 5
660 FOR N = 1 TO 4
665 IF L < 13 THEN L = 13
670 IF A(N) < L AND A(N) + 12 > L THEN B(N) = RND (160): A(N) = 240: SC = S + 10: PUT (L - 12, Q - 11) - (L + 13, Q + 12), BL, PSET: PLAY "T15 CG#C#F": RETURN
680 NEXT
690 SOUND 200, 2
700 RETURN
710 FOR A = 1 TO 200 STEP 5
720 AS = "T" + STRS (A) + "CG#C#F"
730 PLAY AS
740 NEXT
750 cls: printstring$(32, "=")
760 print "YOUR 2 MINUTES HAS EXPIRED" , YOU SC
770 print (Y/N)
780 A$ = inkey$: if a$ = "" then 780: else if A$ = "Y" then poke 16380, 1: run else poke 16380, 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 EXTENDED 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 READ TOP.
60 INPUT"SHOTS FACTOR (1-9)";JJ:IF JJ>9 OR JJ<1 THEN 60
70 RESTORE
80 XX=0:YY=0:X=0:Y=0
90 FOR J=1 TO 8:READU(J),V(J):NEXT
100 FOR I=1 TO 8:READN(I):NEXT
110 AS="ETYUJMNBG"
120 INPUT"(1) JOYSTICKS OR (2) KEYS";A:IF A<>1 AND A<>2 THEN 120 ELSECLS
130 IF A=2 THEN PRINT@0,"TO MOVE USE: TYU";PRINT" BM";IF INKEYS="THEN 130
140 CLS:PRINT@233,"GOOD LUCK"
150 SC=0:CS=0:POKE65494,0:FOR W=1 TO 8:M(W)=0:NEXT
160 PMODE 4,1:CLS
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"BM43,40ND1R1ND10BR2ND10R1D10";GET(40,40)-(50,50),C,G
220 PCLS:GOSUB780
230 DRAW"BM4,36S4NR5D120R5U120BR2NR7NE2F2"
240 SCREEN1,1:IF A=1 THEN 300
250 FOR D,D=1 TO 8;J=INSTR(A$,INKEYS)
260 IF I>1 THEN I=I-1:GOSUB 430
270 NEXT
280 GOSUB 560
290 GOTO250
300 FOR DD = 1 TO JJ: A = JOYSTK(0): B = JOYSTK(1)
310 IF A < 21 AND B < 21 THEN I = 1
320 IF A > 20 AND A < 41 AND B < 21 THEN I = 2
330 IF A > 40 AND B < 41 THEN I = 3
340 IF A > 20 AND A < 41 AND B < 21 THEN I = 3
350 IF A > 40 AND B < 40 THEN I = 4
360 IF A > 20 AND A < 41 AND B < 40 THEN I = 5
370 IF A < 21 AND B < 40 THEN I = 6
380 IF A < 21 AND B < 21 AND B < 41 THEN I = 7
390 IF A < 21 AND B > 40 THEN I = 8
400 GOSUB 430
410 NEXT
420 GOSUB 560: GOTO 300
430 X = U(I) * 1.3 + 128: Y = V(I) * 1.3 + 96: PUT(X-2, Y-2)-(X+3, Y+3), A, PSET: PLAY"04EDF"
440 XX = 128 + 9.8 * U(I): YY = 96 + 9.8 * V(I)
450 PUT(XX-2, YY-2)-(XX+2, YY+2), B, PSET: LINE(XX-2, YY-2)-(XX+2, YY+2), BF
460 XX = XX + U(I): YY = YY + V(I): IF XX < 420 OR XX > 200 OR YY < 50 OR YY > 182 THEN
470 IF PPOINT(XX, YY) = S THEN 490
480 GOTO 450
490 IF I/2 = INT(I/2) THEN S = 75 ELSE S = 50
500 SC = SC + S: CS = CS + S: M(I) = 0: IF SC > 1199 THEN GOSUB 780: SC = 0: LINE (6, 156)-(7, 37), BF: PLAY"02004002D04D03DF01DFEG04DEFG02DEFFG01DFFGCCA"
510 PLAY"07FDE"
520 LINE(X-2, Y-2)-(X+2, Y+2), BF
530 X = U(I) * 17 + 128: Y = V(I) * 17 + 96: LINE(X-5, Y-5)-(X+4, Y+4), PRESET, BF
540 LINE(6, 156)-(7, 156-INT(SC/10)), PSET, BF
550 RETURN
560 FOR DD = 3 TO JJ/3 STEP 1: J = RND(8)
570 IF M(J) = 2 THEN GOSUB 630
580 IF M(J) = 1 THEN M(J) = 2: GOSUB 620
590 IF M(J)=0 THEN M(J)=1
600 NEXT
610 RETURN
620 X=U(J)*17+128:Y=V(J)*17+96:PUT(X-5,Y-5)-(X+4,Y+4),D,PSET
RETURN
630 XX=U(J)*14+128:YY=V(J)*14+96
640 I=N(J)
650 LINE(XX,YY)-(XX+U(I)/5,YY+V(I)/5),PSET:LINE(XX,YY)-(XX+U(I)/5,YY+V(I)/5),PRESET
660 IF H THEN H=0:LINE(XX-6,YY-6)-(XX+6,YY+6),PSET,BF:LINE(XX-6,YY-6)-(XX+6,YY+6),RESET,BF:PLAY"01FF":RETURN
670 IF XX=128 AND YY=96 THEN 700
680 XX=XX+U(I)*2:YY=YY+V(I)*2:IF PPOINT(XX,YY) OR PPOINT(XX+U(I),YY+V(I)) THEN H=1
690 GOTO 650
700 FOR W=1 TO 44:SCREEN 1,RND(2)-1:NEXT:SCREEN 1,1
710 FOR W=1 TO 18:PLAY"V"+STR$(W+12)+"01C":CIRCLE(128,96),W,5:NEXT:FOR W=18 TO 1 STEP -1:PLAY"V"+STR$(W+12)+"DC";CIRCLE(128,96),W,0:NEXT
720 DRAW"BM125,93S4":GOSUB 180:FOR W=1 TO 44:SCREEN 1,1:NEXT:FOR W=1 TO 36:SCREEN 1,0:PRINT"V"+STR$(W);GOSUB 180:NEXT
730 PCLS:CLS:FOR W=1 TO 50:NEXT:SCREEN 1,0:PRINT@0,"TIME-";INT(TIMER/33):PRINT:"SCORE-";CS
740 IF CS>S(JJ) THEN S(JJ)=CS
750 A$=INKEY$:IF A$="H" THEN CLS:PRINT@10,"high scores":PRINT:FOR W=1 TO 9:PRINT"LEVEL"W;TAB((16-LEN(STR$(S(W)))/2));S(W):NEXT:FOR W=1 TO 100:NEXT:GOTO 30
760 IF A$="C" THEN 30 ELSE IF A$<>"" THEN CLS:GOTO 140
770 GOTO 750
780 FOR W=4 TO 12:STEP 4
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,-5,0
820 DATA 5,6,7,8,1,2,3,4
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$="GREG"
L$=""
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 FOR I=0 TO 480 STEP 32: PRINT @I, CHR$(175); STRINGS(30, " "); CHR$(175);: NEXT
70 PLAY "T255"
80 B=RND(100)/100
90 PRINT @0, "SCORE-"; S;
100 IF TIMER>3960 AND TIMER<6930 THEN CC=.65; DD=.8; CO=2
110 IF TIMER>6930 THEN CC=.75; DD=.7; CO=1
120 C=RND(100)/100
130 IF B<CC THEN J=J+1: GOTO 150
140 J=J-1
150 IF C<CC THEN G=G+1: GOTO 170
160 G=G-1
170 IF J>0 AND G>18 THEN J=J-1: G=G+1
180 IF J+G>28 THEN J=J-1: G=G-1
190 IF J<1 THEN J=1
200 IF G<1 THEN G=1
210 GOSUB 270
220 IF RND(8)=5 THEN PRINT @509-J, CHR$(132);
230 IF RND(3)=2 THEN PRINT @480+G, "@";
240 PRINT @480, STRINGS(G, 175+16*CO);
250 PRINT @512-J, STRINGS(J, 175+16*CO);
260 GOTO 80
270 IF J0 = 0 THEN C$ = INKEY$: IF C$ = CHR$(94) THEN U = -32 ELSE IF C$ = CHR$(10) THEN U = 32 ELSE IF C$ = CHR$(8) THEN U = -1 ELSE U = 0
280 IF J0 = 0 THEN IF C$ = "@" THEN GOSUB 370 ELSE IF C$ = " " THEN GOSUB 420
290 IF J0 = 1 THEN Q = JOYSTK(0): Z = JOYSTK(1): IF (Z < 21 AND Q > 21 AND Q < 42) THEN U = -32 ELSE IF (Z > 42 AND Q > 21 AND Q < 42) THEN U = 32 ELSE IF Q < 21 THEN U = -1 ELSE IF Q > 42 THEN U = 1 ELSE U = 0
300 IF J0 = 1 THEN IF PEEK(65280) = 126 THEN PEEK(65280) = 254 THEN GOSUB 420 ELSE IF INKEY$ = " " THEN GOSUB 370
310 IF M + U < 640 OR M + U > 511 THEN M = 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$(132)CHR$(136): PRINT@M, CHR$(139)CHR$(135)
340 IF P = 139 AND PP = 135 THEN 360
350 IF P < 96 AND PP < 96 THEN PLAY"03": FOR K = 1 TO 10: PRINT@M - 32, CHR$(136 + K)CHR$(132 + K): PLAYSTR$(K): NEXT: GOTO 490
360 N = M: RETURN
370 I = M - 33
380 P = PEEK(I + 1024): IF P = 94 THEN S = S + 250
390 IF P = 64 THEN S = S + 200: PRINT@I, CHR$(35)" ": PLAY"03DEFGG": RETURN
400 IF P = 175 THEN R = 191: OR R = 207 THEN S = S - 50: PLAY"04D01D03F": PRINT@I, CHR$(134)" ": PRINT@I, CHR$(137): RETURN
410 PRINT@I, CHR$(60)" ": I = I - 1: GOTO 380
420 I = M
430 P=PEEK(I+1024):PP=PEEK(I+1025):IF P=132ORPP=132THEN S=S+50
:PLAY"01DFDGD"
440 IF P=640ORPP=64THEN S=S+200:PLAY"ABCDE"
450 IF P=175+16*C0 ORPP=175+16*C0 THEN RETURN
460 PRINT@I,CHR$(133)CHR$(138);: :PRINT@I," ";
470 I=I+32:IFI>M+1500RI>511THEN RETURN
480 GOTO430
490 FOR K=1TO90:NEXT:CLS:PRINT@0,"YOUR SCORE WAS: ";S
500 PRINT:INPUT"TRY AGAIN (Y/N)";AA$:IF AA$="Y"THEN GOTO30
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 INKEYS""THEN RETURN 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 <SPACE 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
- SI, 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=0T09
40 READ CH
50 FOR GP=22 TO 12 STEP -1
60 DATA 33, 73, 78, 86, 65, 83, 73, 79, 78, 33
70 POKE GP+1054+MC, CH
80 IF MC=9 THEN POKE GP+1055+MC, 96
90 NEXT: NEXT
100 PRINT: PRINT: TAB(8) "BY ALEX KIERNAN"
110 PRINT "CONVERSION BY DEREK SIMONSON"
120 PRINT: PRINT "ENTER: 1->FOR KEYBOARD"
130 GOSUB 910: JK=VAL(A$): IF JK<10 OR JK>2 THEN 130
140 PRINT: PRINT "INSTRUCTIONS (Y/N)"
150 GOSUB 910: IF A$="Y" THEN 160 ELSE 300
160 CLS: PRINT: PRINT: TAB(12) "!INVASION!
170 PRINT "YOU ARE IN CONTROL OF THE SOLE REMAINING
ALPHA BASE ON THE TERRAN PLANET EARTH IT IS YOUR
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 GOTO 220, 250
220 PRINT: PRINT: "<<<<<<<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
310RD - 9:EASY
320V=VAL(A$):IFV<1 OR V>9THEN310
330PRINT "WHAT'S YOUR MARKSMANSHIP RATING 1:EXCELLENT
- 5:POOR
340GOSUB910
350X=VAL(A$):IFX<10RX>5THEN340
360IX=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
390DRAW "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=1 TO 255 STEP 9
490 CIRCLE(A,192),15,8
500 NEXT
510 FORA=120 TO 1+(V*X) STEP -1: PSET(RND(240)+10, RND(165)),7: NEXT
520 IX=IX+RND(20)-10
530 IY=IY+RND(9)-INT(V/2)
540 IF IX<10 THEN IX=15 ELSE IF IX>240 THEN IX=238
550 IF IY<10 THEN IY=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 IF IY>162 THEN 870
600 ON JK GOTO 610, 650
610 A$=INKEY$
620 IF A$="" THEN 520
630 IF A$="Z" THEN PB=PB-V ELSE IF A$="X" THEN PB=PB+V ELSE IF A$="."
THEN TP=TP-1 ELSE IF A$=CHR$(32) THEN LB=PB: GOSUB 750
640 GOTO 680
650 I=PEEK(65280): IFI=1260 THEN LB=PB: GOSUB 750
660 LE=JOYSTK(0): VO=JOYSTK(1): IF LE<22 THEN PB=PB-V ELSE IF LE>41 THEN PB=PB+V
670 IF VO<22 THEN TP=TP+1 ELSE IF VO<44 THEN TP=2 ELSE TP=3
680 IF TP=4 THEN TP=0 ELSE IF TP=8 THEN TP=1
690 IF PB<10 THEN PB=10 ELSE IF PB>240 THEN PB=240
700 PUT(LB,165)-(LB+14,175), EB, PSET
710 ON TP GOSUB 820, 830, 840
720 LB=PB
730 IF NN=0 THEN 850
740 GOTO 520
750 IF TP=1 THEN MX=PB+2 ELSE IF TP=2 THEN MX=PB+8 ELSE MX=PB+12
760 MY=165
770 IF MX<165 THEN MX<10 OR MX=255 THEN RETUR
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:

<table>
<thead>
<tr>
<th>1. Write down your ideas.</th>
<th>2. Draw up a flowchart.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Use subroutines.</td>
<td>4. Use the same subroutines where possible.</td>
</tr>
</tbody>
</table>

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:

- **START**
- **INPUT or OUTPUT**
- **CALCULATE**
- **MAKE A DECISION**
- **STOP**
Here is a flowchart for SPACE DUEL.

1. **Start**
2. Draw title screen
3. Create men, ship and asteroids
4. Initialize variables
5. Check keyboard
6. Move UFO
   - Move player one
     - Yes
     - Move player one
     - No
     - Fire
       - No
       - Did shot hit player two?
         - Yes
         - Fire
         - No
       - Player one # shots?
         - Yes
         - Player one to fire
         - No
         - Computer playing
           - Yes
           - Computer to move
           - No
           - Computer to fire
             - Yes
             - Move computer
             - No
           - Player two to move
             - Yes
             - Computer to fire
               - Yes
               - Move computer
               - No
             - Computer to fire
               - Yes
               - Move computer
               - No
             - Both players out of shots
               - Yes
               - Stop
               - No
             - Fire
               - No
               - Did shot hit player one?
                 - Yes
                 - Move player one
                 - No
               - Player two to fire
                 - Yes
                 - Fire
                 - No
               - Both players out of shots
                 - Yes
                 - Stop
                 - No
               - Fire
                 - No
                 - Did shot hit player one?
                   - Yes
                   - Move player one
                   - No
                 - Player two to fire
                   - Yes
                   - Fire
                   - No
                 - Both players out of shots
                   - Yes
                   - Stop
                   - No
                 - Fire
                   - No
                   - Did shot hit player one?
                     - Yes
                     - Move player one
                     - No
                   - Player two to fire
                     - Yes
                     - Fire
                     - No
                   - Both players out of shots
                     - Yes
                     - Stop
                     - No
                   - Fire
                     - No
                   - Did shot hit player one?
                     - Yes
                     - Move player one
                     - No
                   - Player two to fire
                     - Yes
                     - Fire
                     - No
                   - Both players out of shots
                     - Yes
                     - Stop
                     - No
                   - Fire
                     - No
                     - Did shot hit player one?
                       - Yes
                       - Move player one
                       - No
                     - Player two to fire
                       - Yes
                       - Fire
                       - No
                     - Both players out of shots
                       - Yes
                       - Stop
                       - No
                     - Fire
                       - No
                       - Did shot hit player one?
                         - Yes
                         - Move player one
                         - No
                       - Player two to fire
                         - Yes
                         - Fire
                         - No
                       - Both players out of shots
                         - Yes
                         - Stop
                         - No
                       - Fire
                         - No
                         - Did shot hit player one?
                           - Yes
                           - Move player one
                           - No
                         - Player two to fire
                           - Yes
                           - Fire
                           - No
                         - Both players out of shots
                           - Yes
                           - Stop
                           - No
                         - Fire
                           - No
                           - Did shot hit player one?
                             - Yes
                             - Move player one
                             - No
                           - Player two to fire
                             - Yes
                             - Fire
                             - No
                           - Both players out of shots
                             - Yes
                             - Stop
                             - No
                           - Fire
                             - No
                             - Did shot hit player one?
                               - Yes
                               - Move player one
                               - No
                             - Player two to fire
                               - Yes
                               - Fire
                               - No
                             - Both players out of shots
                               - Yes
                               - Stop
                               - No
                             - Fire
                               - No
                               - Did shot hit player one?
                                 - Yes
                                 - Move player one
                                 - No
                               - Player two to fire
                                 - Yes
                                 - Fire
                                 - No
                               - Both players out of shots
                                 - Yes
                                 - Stop
                                 - No
                               - Fire
                                 - No
                                 - Did shot hit player one?
                                   - Yes
                                   - Move player one
                                   - No
                                 - Player two to fire
                                   - Yes
                                   - Fire
                                   - No
                                 - Both players out of shots
                                   - Yes
                                   - Stop
                                   - No
                                 - Fire
                                   - No
                                   - Did shot hit player one?
                                     - Yes
                                     - Move player one
                                     - No
                                   - Player two to fire
                                     - Yes
                                     - Fire
                                     - No
                                   - Both players out of shots
                                     - Yes
                                     - Stop
                                     - No
                                   - Fire
                                     - No
                                     - Did shot hit player one?
                                       - Yes
                                       - Move player one
                                       - No
                                     - Player two to fire
                                       - Yes
                                       - Fire
                                       - No
                                     - Both players out of shots
                                       - Yes
                                       - Stop
                                       - No
                                     - Fire
                                       - No
                                       - Did shot hit player one?
                                         - Yes
                                         - Move player one
                                         - No
                                       - Player two to fire
                                         - Yes
                                         - Fire
                                         - No
                                       - Both players out of shots
                                         - Yes
                                         - Stop
                                         - No
                                       - Fire
                                         - No
                                         - Did shot hit player one?
                                           - Yes
                                           - Move player one
                                           - No
                                         - Player two to fire
                                           - Yes
                                           - Fire
                                           - No
                                         - Both players out of shots
                                           - Yes
                                           - Stop
                                           - No
                                         - Fire
                                           - No
                                           - Did shot hit player one?
                                             - Yes
                                             - Move player one
                                             - No
                                           - Player two to fire
                                             - Yes
                                             - Fire
                                             - No
                                           - Both players out of shots
                                             - Yes
                                             - Stop
                                             - No
                                           - Fire
                                             - No
                                             - Did shot hit player one?
                                               - Yes
                                               - Move player one
                                               - No
                                             - Player two to fire
                                               - Yes
                                               - Fire
                                               - No
                                             - Both players out of shots
                                               - Yes
                                               - Stop
                                               - No
                                             - Fire
                                               - No
                                               - Did shot hit player one?
                                                 - Yes
                                                 - Move player one
                                                 - No
                                               - Player two to fire
                                                 - Yes
                                                 - Fire
                                                 - No
                                               - Both players out of shots
                                                 - Yes
                                                 - Stop
                                                 - No
                                               - Fire
                                                 - No
                                                 - Did shot hit player one?
                                                   - Yes
                                                   - Move player one
                                                   - No
                                                 - Player two to fire
                                                   - Yes
                                                   - Fire
                                                   - No
                                                 - Both players out of shots
                                                   - Yes
                                                   - Stop
                                                   - No
                                                 - Fire
                                                   - No
                                                   - Did shot hit player one?
                                                     - Yes
                                                     - Move player one
                                                     - No
                                                   - Player two to fire
                                                     - Yes
                                                     - Fire
                                                     - No
                                                   - Both players out of shots
                                                     - Yes
                                                     - Stop
                                                     - No
                                                   - Fire
                                                     - No
                                                     - Did shot hit player one?
Here is a verbal description of how the game **SPACE DUEL** works:

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

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.