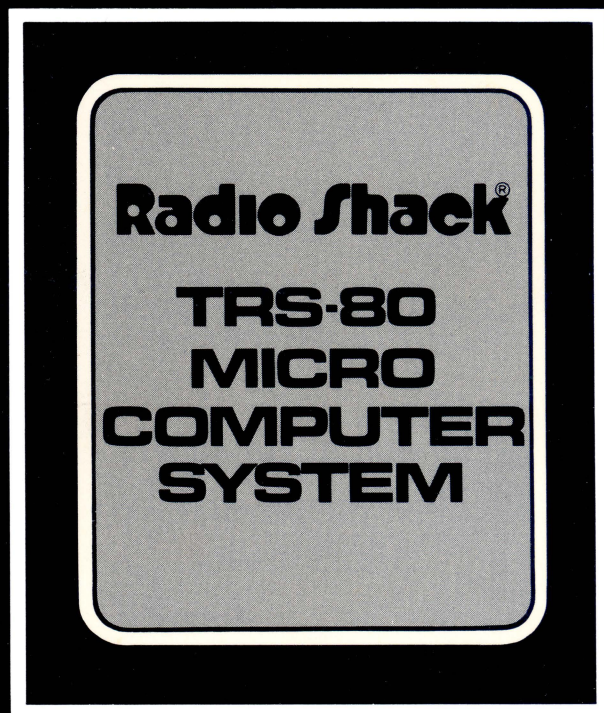


# GAMES PACK I For LEVEL I TRS-80

Catalog Number 26-1805



GAMES

**This package includes six games recorded on three cassettes. All the games are programmed in LEVEL I BASIC so they are simple to load and play. Have fun!**

## **Contents:**

**General Instructions  
Star-Pilot  
Hamurabi  
Space Taxi  
Random Tic-Tac-Toe  
Draw I, II and III  
Checkers  
Listings for the programs**



# General Instructions

All the programs in this package are written in BASIC. That means you can load each of them simply by typing **CLOAD** **ENTER**

1. Set up the Computer, Video Display and Cassette Recorder as described in the **User's Manual for LEVEL I**.
2. Pick out the cassette for the game you want to play — instructions for each game are contained later in this manual.
3. Insert the cassette into your recorder. Be sure the tape is set to start at the beginning. Put the recorder in the Play mode, and set the volume to between 7 and 9.
4. Now type:  
**CLOAD** **ENTER**

The tape will begin to load, and asterisks will flash in the upper left of the Display. When the program has been loaded,

**READY**

>\_

will be displayed below the last line you typed in. If the message:

**WHAT?**

>\_

is displayed instead, start over and try a slightly different volume setting.

5. After loading the program, type

**RUN** **ENTER**

Now the game will start.

**Note:** If for some reason you want to interrupt the game and start over, simply press the **BREAK** key and type **RUN** **ENTER** again.

## Star-Pilot

This game places you in control of an inter-planetary fightercraft. You are to destroy enemy space ships as they come in range of your on-board rockets. The number of enemy space ships and available rockets depends on your experience and confidence as a Star-Pilot (plus a slight "chance" factor).

LOAD the program and RUN it. You will be asked two questions:

**HOW MANY YEARS HAVE YOU BEEN A STAR PILOT?**

and

**HOW MANY SHIPS CAN YOU DESTROY?**

Type in the number of years and the number of ships. You will then be told how many rockets you have, and the screen will instantly be transformed into your target-tracking window.

There are two kinds of enemy space ships. One looks like this: <\*-> and the other looks like this: !\*-!

Wait till an enemy ship is centered in your window over the plus (+) sign. Then press the **CLEAR** key to fire a rocket.

If you destroy all enemy ships, you will be congratulated. Perhaps on your next mission you can try to destroy even more!

If you run out of rockets, the game is over. Ask for less enemy space ships next time!

The Computer will always ask if you wish to play again. Type **YES** or **NO** and press **ENTER**

## **Hamurabi**

You are the exalted ruler of an ancient kingdom. Your objective is to build your kingdom from its present population of 100 poverty-stricken peasants into a wealthy, thriving metropolis.

Agriculture is the only industry, and as a wise ruler, you will decide how many acres to buy and sell at current rates (paying with bushels of wheat). Of the acres you own, you must decide how many to plant. Last year's harvest must also be distributed. Some must go to food for your subjects, some must go to seed for this year's planting, and some may be saved for future years.

There are several obstacles you will encounter. The major one is your own ignorance. Since you just inherited the throne, you don't know how much food each peasant will need to keep from starving. You have no idea how many acres each peasant can plant or even how much wheat it takes to plant an acre.

The Secretary of Commerce would not have the audacity to tell you about your mistakes; he will, however, tell you when your instructions are impossible to execute. You will have to learn by trial and error (or by looking at the program listing) what is best for the kingdom.

To resign from your position, sell all your land.

## Space Taxi

This game challenges your skill as the pilot of a space taxi on a distant planet. Your mission is to maneuver your craft safely over treacherous and unpredictable terrain to a landing port in the next valley. You and your passengers are protected from high radiation by a dome. If you go beyond this dome you will die instantly of radiation poisoning.

You are given 70 units of oxygen and 80 units of fuel. You must take off from the surface by typing in a positive fuel burn for vertical thrust and an optional fuel burn for horizontal thrust. Each maneuver uses 5 seconds' worth of oxygen; you must land with enough oxygen remaining to allow the landing port to be pressurized.

Direction and altitude are controlled by burning thrust-jets in vertical and horizontal directions.

Type in a positive vertical burn to move the craft up or slow your descent. Type in a negative burn to move down or slow your ascent. The acceleration due to gravity for each 5-second burn period is approximately  $-2$  units. Negative velocities mean you are moving down.

Horizontal movement is controlled by entering positive or negative burns for horizontal thrust. Positive burns move the craft to the right and negative burns move the craft to the left.

You must land in the landing port with a vertical velocity of 0 to  $-2$  and a horizontal velocity of 0. Hitting any object outside the landing port will cause you to crash. If you run out of fuel before you have landed safely, you will free-fall and crash.

## Random Tic-Tac-Toe

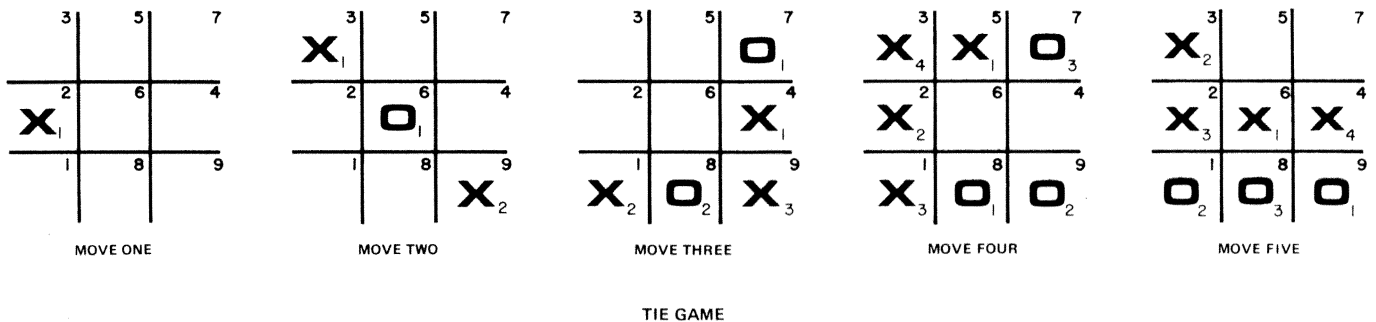
Random Tic-Tac-Toe is an exciting and challenging variation of the old school-yard pastime, with the same objective of the regular game, but requiring the logic and foresight of chess.

The squares of the Tic-Tac-Toe board are randomly numbered 1 through 9, and the player and the computer take turns moving. On the first turn, it is exactly like the regular game, but on the second turn all pieces shift to the next sequential number.

That is, if the Computer opened the game by placing an O in square 5 and you responded by putting an X in square 2, then the board would be re-drawn with an O in square 6 and an X in square 3. Continuing with this example, let's say the Computer next plays to square 8 and you put your mark in square 9, then the redrawn board will have O's in squares 7 and 9 and the X's will be in squares 4 and 1 (after 9 the sequence returns to 1). The first to get three in a row wins.

When your turn comes during play, type the number of the desired square and press **ENTER**. When the game is over, press **ENTER** to play again.

Sample Game:



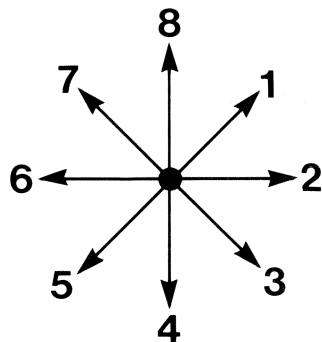
## Draw

This is not just a game — it's a means to let your artistic fancy run wild! The Draw cassette contains three programs:

- DRAW I Instructions for DRAW II and DRAW III
- DRAW II A program that lets you create pictures and designs on the Video Display; and then lets you save them on a blank cassette tape.
- DRAW III A program that loads your previously created designs and pictures from tape and displays them on the screen.

The instructions are contained in the DRAW I program; instead of repeating them here, we'll just offer a few hints on using the programs:

1. Here's an easy-to-remember diagram showing how to indicate the direction you want to "draw" in:



2. In addition to the numbers 1-8, you can type in the numbers 0 and 9. Typing in the number 0 takes the current direction and extends the line (or resets it) by ten units. This saves you from having to type the direction number ten times when you want to continue in the same direction.

Typing the number 9 changes the draw mode: from set to re-set, or vice-versa. In the re-set mode, the draw position will move in the desired direction, "erasing" all points in its path. The mode does not revert back to "set" automatically; you must type 0 again to change the mode.

3. Be careful not to draw outside the frame of the picture.
4. After you have drawn the picture, and you want to save it on tape, be sure to insert the blank cassette and place the recorder in the Record mode — before pressing the **ENTER** key to start the cassette-save process.



## Checkers

This is a completely electronic version of checkers — the board and pieces are created and moved on the screen, and the Computer is your never-tiring opponent.

CLOAD and RUN the program. The Computer will draw the board, and position the playing pieces.

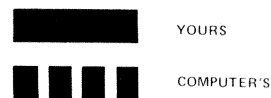
**Note:** After the board is drawn, you may find that the square-numbers and other messages become faint and hard to read. If this happens, adjust the Brightness Control on your Video Display until the messages and numbers are easy to read. When the game is over, you can set the Brightness back to normal.

First the Computer will ask if you want to make the first move. Type **YES** or **NO** and press **ENTER**

### Board and Pieces

The game is played on the 32 dark, numbered squares. The solid pieces at the bottom of the “board” are yours, and they move up, from higher to lower-numbered squares.

The striped pieces are the Computer’s; they move down from lower to higher numbered squares.



When a piece is kinged, it becomes twice as “tall” and then can move in either direction, up or down the board.

### Your move

The computer will ask you **FROM?** in the solid square in the upper right of the board. Type in the number of the square you want to move away from, and press **ENTER**

Then the Computer will ask you **TO?**. Type in the number of the square you wish to move to. If your move involves jumping one of

the Computer's pieces, be sure to type in the number of the square you intend to land on.

Illegal moves:

- trying to move the Computer's piece
- specifying an invalid square number
- trying to move to an occupied square
- trying to jump your own piece, or an empty square
- trying to move backward before you are kinged.

**Double Jumps** should be executed one jump at a time. Type in **FROM?** square number and **TO?** square number so as to complete the first jump. The Computer will see that you have another jump, and will again ask, **FROM?** and **TO?**. Now type in the second part of the double-jump.

**Multiple (more than two) jumps** are handled the same way; if a piece you have just jumped with has another jump to make, you'll always have the opportunity to make such a jump.

## **The Computer's Move**

The Computer will take from 10 to 15 seconds to figure and complete its move. Then it will be your turn again.

### **To Jump or Not to Jump . . .**

Formal international checkers play requires that if a jump is available, the player must take it. TRS-80 checkers is not so strict — you always have the choice whether or not to take a possible jump.

# Listings for Programs

## Star-Pilot

```
00100  GOS. 7000:GOS. 6800

01000  GOS. 4000

01490  P. R. 10, "I"; :P. R. 54, "I"

01500  P. R. 213, "I"; :P. R. 213, "I";

01505  P. R. 280, "I"; :P. R. 280, "I";

01510  P. R. 341, "I"; :P. R. 341, "I";

01515  P. R. 412, "I"; :P. R. 412, "I";

01520  P. R. 469, "I"; :P. R. 469, "I"; :P. R. 469, "I"; :P. R. 469, "I";

01525  P. R. 540, "I"; :P. R. 540, "I";

01530  P. R. 597, "I"; :P. R. 597, "I"; :P. R. 597, "I"; :P. R. 597, "I";

01535  P. R. 664, "I"; :P. R. 664, "I"; :P. R. 664, "I"; :P. R. 664, "I";

01540  P. R. 725, "I"; :P. R. 725, "I"; :P. R. 725, "I"; :P. R. 725, "I"; :P. R. 906, "I";

02400  GOS. 4000

02520  P. R. 146, "I"; :P. R. 146, "I";

02525  P. R. 213, "I"; :P. R. 213, "I";

02530  P. R. 274, "I"; :P. R. 274, "I"; :P. R. 274, "I"; :P. R. 274, "I";

02540  P. R. 346, "I"; :P. R. 346, "I";

02550  P. R. 412, "I"; :P. R. 412, "I";

02560  P. R. 466, "I"; :P. R. 466, "I"; :P. R. 466, "I"; :P. R. 466, "I"; :P. R. 466, "I";

02570  P. R. 540, "I"; :P. R. 540, "I";

02580  P. R. 602, "I"; :P. R. 602, "I";

02590  P. R. 658, "I"; :P. R. 658, "I"; :P. R. 658, "I"; :P. R. 658, "I";

02600  P. R. 725, "I"; :P. R. 725, "I"; :P. R. 725, "I"; :P. R. 725, "I";

02610  P. R. 786, "I"; :P. R. 786, "I"; :P. R. 786, "I"; :P. R. 786, "I";

02620  P. R. 848, "I"; :P. R. 848, "I"; :P. R. 848, "I"; :P. R. 848, "I";

03400  GOS. 4000

03710  P. R. 79, "I"; :P. R. 113, "I"; :P. R. 146, "I"; :P. R. 174, "I"; :P. R. 207, "I";

03720  P. R. 207, "I"; :P. R. 207, "I"; :P. R. 207, "I"; :P. R. 207, "I";

03725  P. R. 274, "I"; :P. R. 274, "I"; :P. R. 274, "I"; :P. R. 274, "I";

03730  P. R. 335, "I"; :P. R. 335, "I"; :P. R. 335, "I"; :P. R. 335, "I";

03735  P. R. 412, "I"; :P. R. 412, "I";

03740  P. R. 458, "I"; :P. R. 458, "I"; :P. R. 458, "I"; :P. R. 458, "I"; :P. R. 458, "I";

03745  P. R. 540, "I"; :P. R. 540, "I";

03750  P. R. 591, "I"; :P. R. 591, "I"; :P. R. 591, "I"; :P. R. 591, "I";

03755  P. R. 658, "I"; :P. R. 658, "I"; :P. R. 658, "I"; :P. R. 658, "I";

03760  P. R. 719, "I"; :P. R. 719, "I"; :P. R. 719, "I"; :P. R. 719, "I";
```

```

03765 P. A. 786, ". - - - - - . ";
03770 P. A. 847, "I-----I";
03990 G. 1000
04000 IFP<P,0)=0G. 5000
04005 P. A. T, R$;
04010 IFW<EW=W+1:G. 5900
04100 K=RND(4)-1:L=RND(2)-1
04150 IFH>31K=-K
04200 IFV>6L=-L
04250 H=H+K:V=V+L:W=1
04320 P. A. T, " ";
04350 T=64*V+H-3
04400 P. A. T, R$;
04450 F. I=1T030+S+E:N. I
04500 G. 5900
05000 P. A. T, R$;:F. J=15T07S. -1
05005 P. A. 64*J+32+0*(J-7), "*" : N. J
05045 CLS : S. (P,0)
05100 IFT=477G0S. 6000
05110 IFS=0G. 8000
05150 R=R-1:IFR=0G. 8500
05200 D=-D
05900 RET.
06000 CLS:I=415:F. J=1T03:P. A. I, "***":I=I+64:N. J
06065 B$="* *":P. A. 350, B$;:P. A. 606, B$;
06070 B$="* *":P. A. 285, B$;:P. A. 669, B$;
06100 F. I=1T06
06110 A(I)=(RND(3)-2)*(64+RND(3)-2)
06112 IFA(I)=0G. 6110
06115 N. I
06120 A(7)=RND(9)-5:A(8)=RND(9)-5
06200 F. I=1T06
06205 F. K=11T018
06210 P. A. A(K), "*";
06215 P. A. A(K)-60, "*";
06220 N. K
06300 F. J=1T08
06305 C=J+10
06310 P. A. A(C), ". ";
06315 P. A. A(C)-60, ". ";

```

```

06320 A(C)=A(C)+A(J)
06330 N. J:N. I
06335 S=S-1
06340 P. A. 900, "YOU HAVE"; R-1; "ROCKETS LEFT. "
06345 P. A. 964, "THERE ARE"; S; "ALIEN SHIPS LEFT. ";
06350 F. I=1T0400*E:N. I
06800 W=0:V=RND(14):H=RND(55):T=64*V+H-3:A$=" <-*->"
06842 A(11)=415:A(12)=416:A(13)=417:A(14)=479:A(15)=481
06846 A(16)=543:A(17)=544:A(18)=545
06860 IFS/2=INT(S/2)A$=" !-0-!"
06870 E=1+INT(S/5):D=3:CLS:P=104:Q=44:S. (P,Q)
06990 RET.
07000 CLS:P. "IN THIS SIMULATION, YOU WILL BE TRYING TO DESTROY"
07100 P. "A FLEET OF ENEMY SPACE SHIPS BEFORE RUNNING OUT OF"
07110 P. "ROCKETS. WHEN A SHIP IS EXACTLY IN THE CENTER OF THE"
07120 P. "TARGET, FIRE AT IT BY PRESSING THE 'CLEAR' KEY. "
07170 P. "" :P. "THE NUMBERS OF SHIPS AND ROCKETS DEPEND UPON"
07180 P. "YOUR EXPERIENCE AS A FIGHTER PILOT AND YOUR CONFIDENCE"
07190 P. "IN YOUR OWN ABILITIES. ENEMY SHIPS BECOME BETTER AT"
07200 P. "TAKING EVASIVE ACTION AS THE SIMULATION PROGRESSES. "
07210 P. "" :IN. "HOW MANY YEARS HAVE YOU BEEN A FIGHTER PILOT":P
07220 R=INT(ABS(R)):IFR>30R=30
07230 IN. "HOW MANY SHIPS CAN YOU DESTROY":S
07240 IFS<85=8
07242 IFS>175=17
07245 S=INT(S+RND(5)-3)
07260 R=INT(S*(2+35/(R+10)))
07265 P. "YOU ARE TO DESTROY":S; "ALIEN SHIPS. "
07270 P. "YOU HAVE":R; "ROCKETS AVAILABLE GOOD LUCK ":
07280 F. I=1TOS*R+300:J=RND(1):N. I
07300 RET.
08000 CLS:P. "YOU HAVE DESTROYED THE LAST ENEMY SHIP!"
08010 P. "CONGRATULATIONS!"
08015 Y=1:N=2
08020 IN. "DO YOU WANT TO RUN ANOTHER SIMULATION":P
08030 IFR<>NG. 100
08040 P. "OK. COME BACK ANY TIME. "
08050 E.
08500 CLS:P. "YOU'RE OUT OF ROCKETS AND THERE ARE STILL ":S
08510 P. "SHIPS LEFT. YOU NEED MORE PRACTICE. " G. 8015

```

# Hamurabi

```
00030 REM COPYRIGHT BY CREATIVE COMPUTING 1978
00040 REM * WHAT THE INSTRUCTIONS DON'T TELL YOU (ON PURPOSE) *
00050 REM * IS THAT: *
00060 REM *      1. YOU BUY OR SELL LAND AT ITS CURRENT *
00070 REM *      VALUE AS LISTED. *
00080 REM *      2. EACH PERSON NEEDS 20 BUSHELS FOR FOOD. *
00090 REM *      3. EACH PERSON CAN PLANT A MAXIMUM OF *
00100 REM *      10 ACRES. *
00110 REM *      4. EACH ACRE COSTS .5 BUSHELS TO PLANT. *
00120 REM * THESE ARE THINGS YOU SHOULD DISCOVER FOR YOURSELF *
00130 REM * AS YOU PLAY HAMURABI. NOW AREN'T YOU ASHAMED FOR *
00140 REM * READING THIS?!? *
00150 REM * * * * *
00300 CLS:P:P:P:"HAMURABI -"
00310 P:"WHERE YOU GOVERN THE ANCIENT KINGDOM OF SUMERIA. "
00320 P:"THE OBJECT IS TO KEEP THE KINGDOM GROWING. "
00330 P:P:"(IF YOU WANT TO QUIT, SELL ALL YOUR LAND)"
00340 P:P:P:I:"PRESS ENTER TO BEGIN YOUR REIGN";A$
00400 A=100:B=5:C=0:D=2800:E=200:F=3:G=3000:H=1000:J=1:L=1
01010 CLS:P=0:P:"HAMURABI, I BEG TO REPORT THAT IN YEAR";L;": "
01020 IFC=1T.1035
01030 P:C:"PEOPLE STARVED, AND";:G.1040
01035 P:" 1 PERSON STARVED, AND";
01040 IFB=1THEN1048
01045 P:B:"PEOPLE CAME TO THE CITY.":G.1050
01048 P:" 1 PERSON CAME TO THE CITY. "
01050 IFJ>0T.1100
01060 A=A-INT(A/2):P:" THE PLAGUE KILLED HALF THE PEOPLE. "
01100 P:" THE POPULATION IS NOW";A;". "
01120 P:" WE HARVESTED";G;"BUSHELS AT";F;"BUSHELS PER ACRE. "

01130 P:" RATS DESTROYED";E;"BUSHELS, LEAVING";D;"BUSHELS IN STORAGE. "
01150 P:" THE CITY OWNS";H;"ACRES OF LAND. "
01170 K=16+RND(6):P:" LAND IS WORTH";K;"BUSHELS PER ACRE. "
01200 P.AT518,"HAMURABI . . . ."
01300 P.AT576,"":Q=576
01310 I:" BUY HOW MANY ACRES";I:I=INT(ABS(I)):IFI=0T.1500
```

```

01360 J=I*K:IFJ<DT.1400
01380 GOS.9000:G.1310
01400 D=D-J:H=H+I
01500 P.AT576," * YOU ARE BUYING":I:"ACRES.
01505 IFI>DT.1650
01510 P.AT640," "
01520 P.AT704," "
01530 Q=640:P.AT0,"";
01540 I." SELL HOW MANY ACRES":I:I=INT(ABS(I)):IFI=DT.1570
01550 IFI<HT.1570
01560 IFI=HT.9999
01565 GOS.9000:G.1540
01570 P=1:P.AT640," * YOU ARE SELLING":I:"ACRES.
01580 G.1600
01600 H=H-I:D=D+K*I
01650 Q=P*64+640
01670 P.AT0," "
01680 P.AT0+64," "
01690 P.AT0,"";
01700 I." HOW MANY BUSHELS SHALL WE DISTRIBUTE AS FOOD":I
01710 I=INT(ABS(I))
01740 IFI<DT.1770
01750 GOS.9000:G.1700
01770 D=D-I:C=A-INT(I/20):B=0:IFC=DT.1850
01810 B=-C/2:C=0
01850 Q=P*64+704
01860 P.AT0-64," * YOU ARE DISTRIBUTING":I:"BUSHELS.
01870 P.AT0," "
01880 P.AT0+64," "
01890 P.AT0,"";
01900 I." HOW MANY ACRES SHALL WE PLANT":I:I=INT(ABS(I))
01910 IFI>HT.1960
01940 J=INT(I/2):IFJ<DT.1980
01960 GOS.9000:G.1900
01980 IFI>10*AT.1960
01985 CLS
01990 D=D-J:F=RND(5):G=F*I:E=INT((D+G)*.07*RND(0)):D=D-E+G:J=RND(11)-1
02060 B=INT(B+(5-F)*D/600+1):IFB<=50T.2100
02080 B=50
02100 IFB<DT.B=0

```

```

02110 A=A+B-C:L=L+1.G.1010
09000 P. AT0+128."
09001 P. " ",
09010 P. AT0."",
09020 P. "--> HAMURABI! THINK AGAIN -- YOU ONLY HAVE "
09030 P. "--> ";A;"PEOPLE/";H;"ACRES, AND";D;"BUSHELS IN STOREHOUSES."
09040 RET
09999 CLS:P:P:P:P:P:"YOU HAVE SOLD ALL YOUR LAND."
10000 P:P:"THE GAME IS OVER, AND YOU CAN GO BACK TO BEING YOURSELF."
10001 P:P:"BYE" F.T=1T09999 N.T.CLS.END

```



# Space Taxi

```
00001 CLS: F=80

00002 U=1: F: Z=80T047: S: (Z,U): N: Z: F: A: 24: " RADIATION DOME: "

00003 F: Z=80T0127: S: (Z,U): N: Z

00011 T=70: V=0: H=0: C=22: D=30: I=0: J=22: K=32: G05: 30: G: 120

00030 S: (I, J)

00040 R=RND(3): ONRG: 50, 70, 80

00050 IFX=1G: 40

00060 I=I+1: X=1: Y=0: G: 100

00070 I=I+1: J=J+1: X=0: Y=0: G: 100

00080 IFY=1G: 40

00090 J=J+1: Y=1: X=0

00100 IFJ=KRET.

00110 IFI=127RET.

00111 G: 30

00120 F: Z=I+1T029: S: (Z, J): N: Z: K=RND(11)+4

00130 J=31: I=30: G05: 140: G: 230

00140 S: (I, J)

00150 R=RND(3): ONRG: 160, 180, 190

00160 IFX=1G: 150

00170 I=I+1: X=1: Y=0: G: 210

00180 I=I+1: J=J-1: X=0: Y=0: G: 210

00190 IFY=1G: 150

00200 J=J-1: Y=1: X=0

00210 IFJ=KRET.

00220 IFI=127RET.

00221 G: 140

00230 K=32: G05: 30: A=I+5: B=31

00240 F: Z=I+1T01+20: IFZ=127G: 310

00250 S: (Z, J): N: Z

00260 J=J-1: I=I+21: K=RND(12)+3: G05: 140

00270 IFI=127G: 310

00280 K=31: G05: 30: IFI=127G: 310

00290 F: Z=I+1T01+17: IFZ=127G: 310

00300 S: (Z, J): N: Z

00310 S: (C, D): S: (C-1, D+1): S: (C+1, D+1): S: (A-1, B): S: (A-2, B)

00311 S: (A-2, B-1): S: (A+7, B)

00320 S: (A+8, B): S: (A+8, B-1): G05: 321: G: 350

00321 P: A: 708: "HORZ. SPEED: =" : H

00330 P: A: 772: "OXY/SEC: =" : T
```

```

00340 IFV<=0P. A. 739; "VERT. SPEED =" ;ABS(V)
00341 IFV>0P. A. 739; "VERT. SPEED =" ;0-V
00342 P. A. 803; "FUEL =" ;F:RET.
00350 P. A. 832; "HORIZ. -J+ THRUST" ;I. X
00355 IFABS(X)>FG05.1000;G. 350
00360 P. A. 896; "VERT. +E\= THRUST" ;I. Y
00361 X=INT(X);Y=INT(Y)
00365 IFABS(Y)>FG05.1000;G. 360
00367 IFABS(X)+ABS(Y)>FG05.1000;G0T0350
00370 H=H+X;V=V-Y;H=INT(H);V=INT(V);M=0;N=0
00371 F=F-(ABS(X)+ABS(Y))
00380 R. (C,D);R. (C-1,D+1);R. (C+1,D+1)
00390 IFM=HG. 550
00400 IFN=VG. 500
00410 IFH<0C=C-1;M=M-1;G. 430
00420 C=C+1;M=M+1
00430 IFV<0D=D-1;N=N-1;G. 441
00440 D=D+1;N=N+1
00441 G05. 450;S. (C,D);S. (C-1,D+1);S. (C+1,D+1);G. 380
00450 IF(C=1)+(C=126)+(D=0)CLS;G. 780
00455 IF(D=30)*((C<0)+(C>0)+7)*((V<=2)+(H=0);T. V=0;G05. 1100;G. 320
00460 IF(P. (C-2,D+1)=1)+(P. (C+2,D+1)=1);G. 700
00470 IF(D=30)*(V<=2)V=0;G. 800
00480 IF(D=31)+(D=30);G. 700
00490 RET.
00500 S. (C,D);S. (C-1,D+1);S. (C+1,D+1);IFM=HG. 600
00510 R. (C,D);R. (C-1,D+1);R. (C+1,D+1)
00520 IFH<0C=C-1;M=M-1;G. 540
00530 C=C+1;M=M+1
00540 G05. 450;S. (C,D);S. (C-1,D+1);S. (C+1,D+1);G. 500
00550 S. (C,D);S. (C-1,D+1);S. (C+1,D+1);IFN=VG. 600
00560 R. (C,D);R. (C-1,D+1);R. (C+1,D+1)
00570 IFV<0D=D-1;N=N-1;G. 590
00580 D=D+1;N=N+1
00590 G05. 450;S. (C,D);S. (C-1,D+1);S. (C+1,D+1);G. 550
00600 T=T-5;IFT=0P. A. 192; "GASP!! YOUR OXYGEN IS GONE" ;G. 720
00610 IFD<>30V=V+2
00621 P. A. 704
00630 F. Z=1T03;P. ;N. Z
00640 IFF>0T. 685
00680 IFF<=0P. A. 838; "=== O U T O F F U E L ===";V=V+2

```

```

00685 GOSUB321
00690 IFF<=0G.370
00691 G.350
00700 CLS:P.A.474;"CRASH.....":F.Z=1T01000:N.Z
00710 P.A.523;"BOOM!! YOU JUST HIT THE SURFACE OF ZARBOR"
00711 P."YOUR FINAL VERTICAL VELOCITY WAS";
00712 IFV<=0P.V
00713 IFV>0P.0-V
00714 P."YOUR HORIZONTAL VELOCITY WAS";H
00720 I."> PLAY AGAIN (Y/N) ":Z:IFZ=YF=80:CLS:G.2
00730 IFZ<>NG.720
00731 E.
00780 P."YOU HAVE LEFT THE SAFETY OF THE RADITION DOME"
00790 P."AND HAVE DIED OF RADITION POISONING":G.720
00800 S.(C,D):S.(C-1,D+1):S.(C+1,D+1):IFHC>0G.700
00810 F.Z=ATORA+5:IF(C-1=Z)+(C+1=Z)G.860
00811 P.A.704
00820 N.Z:P."YOU HAVE LANDED"
00850 F.M=1T02000:N.M.G.490
00860 RESTORE
00880 DATA-1,-2,7,-2,0,-3,6,-3,1,-4,5,-4,2,-5,3,-5,4,-5
00890 F.Z=1T09:READU,W:S.(A+U,B+W)
00900 T=T-.8:IFT<=0G.950
00901 V=0:GOS.321
00910 F.E=1T0250:N.E.N.Z
00920 CLS:P."CONGRATULATIONS!! YOU HAVE MADE IT TO THE BASE":G.720
00950 P."UNFORTUNATLY YOUR OXYGEN SUPPLY RAN OUT BEFORE"
00960 P."THE LANDING PORT COULD BE PRESSURIZED. YOU"
00970 P."SUFFOCATED " :G.720
00980 RET
01000 P.A.965;"*** YOU DON'T HAVE THAT MUCH FUEL ***";
01010 F.X=1T.1000:N.X
01020 P.A.965;" " " " RET
01100 P.A.960;"** YOU HAVE LANDED IN THE WRONG PLACE **";
01105 S.(C,D):S.(C-1,D+1):S.(C+1,D+1)
01110 F.X=1T01500:N.X
01120 P.A.960;" " " " RET

```

# Random Tic-Tac-Toe

```

00001 CLS:P.AT463,"RADIO SHACK'S RANDOM TIC TAC TOE":F,X=1T01500:N,X
00002 CLS:P.AT471,"ENTER FIRST NAME ":IN B$:J=0:U=0:V=0:W=0
00003 CLS:F,Q=1T020:A(Q)=0:N,Q:P.AT472,"I'M RANDOMIZING":P.AT0,"":
00004 Q=R:(9):F,X=11T019:IFA(X)=0A(X)=Q:G,4
00005 IFX=19T,8
00006 IFA(X)=QT,4
00007 N,X
00008 GOS,3333:GOS,10:G,95
00010 CLS:X=41:T=82:F,Y=3T043:S,(X-1,Y):S,(X,Y):S,(T,Y):S,(T+1,Y):N,Y
00020 Y=15:Z=31:F,X=12T0113:S,(X,Y):S,(X,Z):N,X:P.AT272,A(11)
00030 P.AT293,A(12):P.AT311,A(13):P.AT592,A(14):P.AT613,A(15)
00040 P.AT631,A(16):P.AT912,A(17):P.AT933,A(18):P.AT951,A(19)
00050 P.AT448,"WON":P.AT512,U:P.AT508,"LOST":P.AT572,V
00060 P.AT989,"TIED ":W:P.AT0,"":RET.
00095 IFJ=1J=0:G,7000
00097 J=1
00098 P,T,(22),B$: " SELECT SQUARE":P.AT0,"":IN,Q:P.AT0,"":
00101 IF(Q>0)*(Q<10)T,103
00102 P,"INVALID PLAY":F,T=1T01000:N,T:P.AT0,"":P.AT0,"":G,98
00103 IFA(Q)<>0T,102
00104 A(Q)=1:GOS,555:GOS,10:GOS,3333:M=1:S=4:GOS,666:T=1:G,8888
00110 S,(X-4,Y-4):S,(X+4,Y-4):S,(X-3,Y-3):S,(X+3,Y-3):S,(X-2,Y-2)
00120 S,(X+2,Y-2):S,(X-1,Y-1):S,(X+1,Y-1):S,(X,Y):S,(X-1,Y+1):S,(X+1,Y+1)
00130 S,(X-2,Y+2):S,(X+2,Y+2):S,(X-3,Y+3):S,(X+3,Y+3):S,(X-4,Y+4)
00140 S,(X+4,Y+4):G,680
00199 A(R)=4:P,"":P.AT0,"":
00200 F,T=-4T04:S,(X+T,Y-4):N,T:F,T=-3T03:S,(X-4,Y+T):S,(X+4,Y+T):N,T
00210 F,T=-4T04:S,(X+T,Y+4):N,T:G,690
00555 F,R=9T01S,-1:A(R+1)=A(R):N,R:A(1)=A(10):RET.
00666 F,R=1T09:IF(A(R)=M)+(A(R)=S)Q=R:G,1000
00670 IFA(R)=MT,110
00680 IFA(R)=ST,199
00690 N,R:RET.
01000 K=20:L=7:F,T=1T09:IFQ=A(T+10)X=K:Y=L:G,1050
01010 K=K+42:IFK>104K=20:G,1030
01020 G,1040
01030 L=L+16
01040 N,T

```

```

01050 G. 670
03333 A=R(A(11)):B=R(A(12)):C=R(A(13)):D=R(A(14)):E=R(A(15)):F=R(A(16))
03334 G=R(A(17)):H=R(A(18)):I=R(A(19)):RET.
04000 Q=(A(M))-T:IFQ=-2Q=7
04001 IFQ=-1Q=8
04002 IFQ=0Q=9
04003 IFA(Q)=0A(Q)=9 G. 8500
04004 IFM=17M=13
04005 IFM=19M=17
04006 IFM=15M=19
04007 G. 4000
07000 P. T. (25). "I'M THINKING"...P. AT0. "...Y=0 Z=Y X=8
07001 IFA(20)=0A(20)=1:M=15 T=3 G. 4000
07010 T=A+B+C:IFT=XT. 8000
07779 T=A+D+G:IFT=XT. 8030
07780 T=A+E+I:IFT=XT. 8060
07781 T=B+E+H:IFT=XT. 8090
07782 T=C+E+G:IFT=XT. 8120
07783 T=C+F+I:IFT=XT. 8150
07784 T=D+E+F:IFT=XT. 8180
07785 T=G+H+I:IFT=XT. 8210
07786 IFY=0Y=1:Z=1 G. 7800
07787 IFY=1Y=2:X=2 G. 7010
07788 IFY=2Y=3:X=8:Z=2 G. 7800
07789 IFY=3Y=4:X=2 G. 7010
07790 IFY=4Y=5:X=8:Z=3 G. 7800
07791 IFY=5Y=6:X=2 G. 7010
07792 IFY=6Y=7:X=8:Z=4 G. 7800
07793 IFY=7Y=8:X=2 G. 7010
07799 G05. 8450 M=19 T=2 G. 4000
07800 G05. 555 G05. 3333 G. 7010
08000 IFA=0T. 8490
08010 IFB=0T. 8491
08020 G. 8492
08030 IFA=0T. 8490
08040 IFD=0T. 8493

```

```

08050 G. 8496
08060 IFA=0T. 8490
08070 IFE=0T. 8494
08080 G. 8498
08090 IFB=0T. 8491
08100 IFE=0T. 8494
08110 G. 8497
08120 IFC=0T. 8492
08130 IFE=0T. 8494
08140 G. 8496
08150 IFC=0T. 8492
08160 IFF=0T. 8495
08170 G. 8498
08180 IFD=0T. 8493
08190 IFE=0T. 8494
08200 G. 8495
08210 IFG=0T. 8496
08220 IFH=0T. 8497
08230 G. 8498
08400 R(R(0))=9. GOS. 8450 G. 8500
08450 S=0. T=9-Z
08451 GOS. 555-S=S+1. IFS<TT. 8451
08452 RET.
08490 O=11 G. 8400
08491 O=12 G. 8400
08492 O=13 G. 8400
08493 O=14 G. 8400
08494 O=15 G. 8400
08495 O=16 G. 8400
08496 O=17 G. 8400
08497 O=18 G. 8400
08498 O=19 G. 8400
08500 M=5. S=9. GOS. 666. GOS. 3333
08888 X=12. Z=0
08900 IF (A+B+C=X)+(A+D+G=X)+(A+E+I=X)+(B+E+H=X)+(C+E+G=X)T. Z=1
08910 IF (C+F+I=X)+(D+E+F=X)+(G+H+I=X)Z=1

```

```

08911 X=3
08912 IF(A+B+C=X)+(A+D+G=X)+(A+E+I=X)+(B+E+H=X)+(C+E+G=X)T, Z=Z+2
08913 IF(C+F+I=X)+(D+E+F=X)+(G+H+I=X)Z=Z+2
08914 IFZ=1T, 8930
08915 IF((Z=2)+(Z=4))T, 8931
08916 IF(Z=3)+(Z=5)T, 10020
08920 G, 9999
08930 P, AT984, B$; " YOU LOSE"; :P, AT0, ""; :V=V+1:IN, A$; G, 3
08931 P, AT984, B$; " YOU WIN"; :P, AT0, ""; :U=U+1:IN, A$; G, 3
09000 IFT=1T, 7000
09001 G, 98
09999 F, Q=1T09:IF(A(Q)=0)Q=9:G, 9000
10010 N, Q
10020 P, AT984, "TIE GAME "; B$; :P, AT0, ""; :IN, A$; W=W+1:G, 3
22222 END

```

# Draw

## DRAW I

```
00001 REM
00002 REM
00003 REM AUTHOR CLOAD STAFF (RF)
00004 REM
00005 CLS:P." *** DRAW I PROGRAM ***":P.
00010 P." THESE ARE THE INSTRUCTIONS TO THE GRAPHICS PROGRAM.
00020 P.
00030 P."THE NEXT PROGRAM ON THIS TAPE (DRAW II) ALLOWS YOU TO
00040 P."ACTUALLY DRAW ON THE SCREEN WITHOUT PROGRAMMING IN EACH POINT.
00050 P."AFTER YOU HAVE CREATED YOUR MASTERPIECE, THE PROGRAM WILL READ
00060 P."TO YOU THE X AND Y FOR EACH POINT YOU PLOTTED. EVEN MORE THAN
00070 P."THIS, THE PROGRAM WILL ALLOW YOU TO SAVE YOUR ARTWORK ON A
00080 P."DATA TAPE. THE THIRD PROGRAM ON THIS TAPE, DRAW III, WILL TAKE
00090 P."YOUR ARTWORK FROM THE DATA TAPE AND PLOT IT ONCE AGAIN.
00100 P.:P.:P.
00110 GOS.800
00115 CLS
00120 P."HERE IS HOW IT WORKS.
00130 P.
00140 P."WHEN YOU RUN THE PROGRAM, THE FIRST THING YOU WILL SEE IS:"
00150 P.
00160 P."/STAND-BY FOR THE COMPUTER TO SAY W-H-E-R-E!!!!/"
00170 P.
00180 P."THIS IS A STALL TACTIC.
00185 P."THE COMPUTER IS FILLING ALL 'A' ARRAY POSITIONS WITH ZERO.
00200 P."NEXT YOU WILL SEE:
00210 P.
00220 P."/YOU CAN 'SET' #### TIMES. WHERE?/"
00230 P.
00240 P."THE NUMBER AFTER 'SET' IS BASED ON THE MEMORY FOR YOUR COMPUTER
00242 GOS.800
00244 CLS
00250 P."SO 16K'ERS, THIS PROGRAM CAN USE ALL THE ROOM YOU HAVE.
00260 P."/WHERE', IS ASKING WHERE YOU WOULD LIKE TO START WE HAVE
00270 P."FOUND THAT 22,22 IS A GOOD PLACE TO START INPUT AT THIS POINT
00280 P."MUST BE #,# WITH X COMING FIRST AFTER ENTERING THIS YOU WILL
00290 P."SEE:"
```



```

00300 P.
00310 P. "X= 22 Y= 22 DRTN= 0 #### RESERVE SET 2/"
00320 P.
00330 P. "THE X AND Y ALWAYS SHOW THE POSITION OF THE LAST SET OR RESET.
00340 P. "DRTN, MEANS THE CURRENT DIRECTION AS DESCRIBED LATER. EACH
00360 P. "TIME YOU SET, THE RESEVE IS REDUCED BY ONE.
00370 P. "EACH TIME YOU RESET, RESERVE IS RAISED BY ONE. SET REFERS TO
00372 P.
00374 GOS. 800
00376 CLS
00380 P. "WHETHER YOU WILL BE SETTING OR RESETTING ON THE NEXT MOVE.
00400 P. "THE QUESTION MARK IS ASKING WHICH DIRECTION YOUR NEXT MOVE WILL
00410 P. "BE. YOU WILL NOTICE THAT THIS LINE OF DATA TAKES THE TOP LINE
00420 P. "OF THE SCREEN. NO GRAPHICS CAN BE DRAWN HERE
00430 P.
00432 GOS. 800
00434 CLS
00440 P. "THIS IS THE CODE FOR DIRECTION. BEST WRITE IT DOWN!
00450 P. "1 UP AND RIGHT
00460 P. "2 RIGHT
00470 P. "3 DOWN AND RIGHT
00480 P. "4 DOWN
00490 P. "5 DOWN AND LEFT
00500 P. "6 LEFT
00510 P. "7 UP AND LEFT
00520 P. "8 UP
00530 P. "9 CHANGES SET TO RESET OR VICE VERSA
00540 P. "0 TAKES THE LAST DIRECTION (1-8) AND REPEATS IT TEN TIMES
00550 P. "100 SAVES THE ARTWORK. DON'T DO THIS UNTIL
00551 P. " YOU'RE ALL FINISHED!!
00553 P.
00554 GOS. 800
00556 CLS
00560 P. "WHEN YOU HAVE COMPLETED YOUR ARTWORK ENTER 100.
00570 P. "THE COMPUTER WILL THEN SAY:
00580 P.
00590 P. "<THE COMPUTER IS NOW SAVING YOUR ARTWORK! IT TAKES A WHILE.>"
00600 P.
00610 P. "AFTER ABOUT 2 MINUTES THE COMPUTER WILL BEGIN SHOWING YOU LINES

```

```

00620 P. "OF DATA. IT WILL LOOK LIKE: 2222 2322 2422 2522 ETC.
00630 P. "THE LAST TWO NO'S ARE ALWAYS THE Y POINT. THE LEADING NO'S
00640 P. "ARE THE X POINT. X GOES 0 TO 127; Y GOES 3 TO 47.
00650 P. "AFTER YOU STEP THROUGH THE POINTS THE COMPUTER WILL SHOW YOU
00660 P. "THE MEMORY USED, BOTH IN % AND AMOUNT. THEN IT WILL SAY, IF
00670 P. "YOU WISH TO SAVE THE ARTWORK, PREPARE A CASSETTE TO RECEIVE
00680 P. "THE DATA AND PRESS ENTER. YOU MAY SAVE MORE THAN ONCE.
00690 P. "(PRESS BREAK TO STOP THE PROGRAM AT THIS POINT.)
00700 GOS. 800
00710 CLS
00720 P. "AS WAS STATED, THE THIRD PROGRAM ON THIS TAPE, DRAW III, WILL
00730 P. "TAKE A DATA TAPE AND TURN IT BACK INTO YOUR ARTWORK.
00740 P.
00750 P. "WE HAVE HAD LOTS OF FUN WITH THIS PROGRAM. HOPE YOU FIND
00760 P. "IT FUN ALSO.
00780 END
00800 IN. "PRESS ENTER TO TURN THE PAGE ";A$:RET

```

## DRAW II

```

00003 REM BYTES LEFT 14006/1718
00030 CLS
00040 P. :P. :P. :P. :P. :P. :P.
00050 P. "STAND-BY FOR THE COMPUTER TO SAY W-H-E-R-E !!!!!!"
00070 A=0:K=INT(M./4-1)
00080 F. I=0T0(M./4-1):A(I)=0:N. I
00090 R=1
00110 CLS
00120 P. AT0, "YOU CAN /SET/";K; "TIMES. "; :IN. " WHERE";X,Y
00130 S. (X,Y)
00135 P. AT0, "X=";X; " Y=";Y; " DRTN=";A; " ";K; "RESERVE ";
00136 IFS<>0P. "SET "; :IN. B
00137 IFS=0P. "RESET "; :IN. B
00140 IFB=100G. 600
00141 IFB=90G. 500
00142 IFS=0R. (X,Y)
00143 IFB>0G. 148
00144 F. I=1T010
00146 G. 150

```

```

00148 A=B
00150 IFA=1X=X+1:Y=Y-1
00160 IFA=2X=X+1
00170 IFA=3X=X+1:Y=Y+1
00180 IFA=4Y=Y+1
00190 IFA=5X=X-1:Y=Y+1
00200 IFA=6X=X-1
00210 IFA=7X=X-1:Y=Y-1
00220 IFA=8Y=Y-1
00222 X=ABS(X):Y=ABS(Y)
00223 IF(S=0)*(P,(X,Y))K=K+1
00224 IFS<>0K=K-1:IFP,(X,Y)K=K+1
00225 S=(X,Y)
00230 IFS=0R=(X,Y)
00232 IFB>0G.240
00236 N.I
00240 IFS=0S=(X,Y)
00250 G.135
00500 R=R+1
00510 S=R/2-INT(R/2)
00520 IFS<>0K=K-1
00525 IFS=0K=K+1
00530 RET
00600 REM LOGGING
00602 P.AT0,"THE CMPT. IS NOW SAVING YOUR ART WORK' IT TAKES A WHILE.";
00605 K=0
00610 F.J=3T047
00620 F.I=0T0127
00630 IFP,(I,J)=1R(K)=(I*100+J):K=K+1
00640 N.I
00650 N.J
00690 K=0
00691 M=A(0)/100-INT(A(0)/100)
00695 CLS
00699 P."THIS IS YOUR FIRST LINE OF X DATA."
00700 REM LOOKING
00710 F.I=0T010000
00711 IFK>9K=1
00713 L=(A(I)/100-INT(A(I)/100))-M

```

```

00716 IFL>.001K=1:P:IN."PRESS ENTER FOR THE NEXT X LINE OF DATA.":A$
00720 IFA(I)>1P:A(I)
00725 M=A(I)/100-INT(A(I)/100)
00730 K=K+1
00731 IFK=9P.
00732 IFA(I)<>0G.760
00733 P:P:"YOU USED":INT(I/M*.400):"% OF THE ROOM IN THE MEMORY."
00734 P:"OUT OF A POSSIBLE":INT(M/.4-1):"YOU USED":I+1: "."
00735 P:P:"IF YOU WISH TO SAVE YOUR ART WORK, PREPARE A CASSETTE TO
00740 P:"RECORD DATA. EITHER PRESS 'BREAK' TO STOP, OR
00745 IN."WHEN DATA TAPE IS READY, PRESS ENTER.":A$
00750 L=INT(I/10)+2
00755 G.900
00760 N.I
00800 P:" SOMETHING WENT WRONG. I IS NOW = TO 10000."
00801 STOP
00900 F.K=0T01000STEP10
00904 L=L-1
00905 P.L
00910 A=A(K):B=A(K+1):C=A(K+2):D=A(K+3):E=A(K+4)
00920 F=A(K+5):G=A(K+6):H=A(K+7):I=A(K+8):J=A(K+9)
00940 P:#A:":":B:":":C:":":D:":":E:":":F:":":G:":":H:":":I:":":J
00945 IFJ=0T.960
00950 N.K
00960 P:P:"THE SAVING IS COMPLETE."
00970 G.733

```

### DRAW III

```

00003 REM DRAW III PROGRAM
00004 K=0
00005 CLS
00006 P:TAB(15):"*** DRAW III ***":P.
00007 P:"REDRAWS GRAPHICS DATA TAPES
00008 P:"PRESS 'BREAK' TO STOP AFTER PICTURE IS DRAWN.":P.
00010 IN."WHEN YOU HAVE THE CASSETTE READY TO LOAD PRESS ENTER ":A$
00020 F.I=0T01000STEP10
00025 K=K+1:P.K
00030 IN.#A:B:C:D:E:F:G:H:L:J
00040 A(I)=A:A(I+1)=B:A(I+2)=C:A(I+3)=D:A(I+4)=E
00042 A(I+5)=F:A(I+6)=G:A(I+7)=H:A(I+8)=L:A(I+9)=J

```

```
00045 IFJ=0G.100
00050 N. I
00100 CLS
00110 F. I=0T010000
00120 X=INT(R(I)/100)
00130 Y=R(I)-INT(R(I)/100)*100
00140 IFR(I)=0G.200
00150 S. (X, Y)
00160 N. I
00200 G. 200
```

# Checkers

```

00100  CLS:GOS.3000:FORU=0T01:GOS.2000:N,U:GOS.2500

00230  FORN=1T064:GOS.3500:N,N=N+1:P.AT192,"WANT TO GO FIRST";

00240  IN,R:P.AT192,"          ";:GOS.6000:IFA=1T.GOS.5000

00300  GOS.1000:GOS.5000:G.300

01000  P.AT184,"          ";:P.AT184,"FROM";:IN,F:Q=1

01005  IF<F<1>+<F>64>T.GOS.6000:G.1000

01006  P.AT184," TO ";:IN,T:IF<T=0>*<Q=0>T.RET.

01010  IF<T<1>+<T>64>T.GOS.6000:G.1006

01015  FORX=112T0127:FORY=6T011:S.<X,Y>;S.<X,Y+12>;N,Y:N,X

01020  GOS.1500:IFQ=0T.P.AT440,"ILLEGAL";:P.AT504," MOVE  ";:G.1000

01100  A<T>=A<F>;A<F>=0:N=F:GOS.3500:N=T:GOS.3500

01150  IFT<9T.A<T>=-2:N=T:GOS.3500

01200  IF<ABS<T-F>=7>+<ABS<T-F>=9>T.RET.

01250  N=<T+F>/2:A<N>=0:GOS.3500

01255  IFA<T>=-1T.1270

01260  FORB=7T09S.2:G=T+B:IF<G>64>+<G+B>64>T.1266

01265  IF<<A<G>=1>+<A<G>=2>>*<A<G+B>=0>T.F=T:Q=0:G.1006

01266  N,B

01270  FORB=-9T0-7S.2:G=T+B:IF<G<0>+<G+B<0>T.1290

01280  IF<<A<G>=1>+<A<G>=2>>*<A<G+B>=0>T.F=T:Q=0:G.1006

01290  N,B:RET.

01500  Q=0:IF<A<F>>-1>+<A<T><0>T.RET.

01520  IFA<F>=-2T.FORB=-9T0-7S.2:GOS.1560:N,B

01540  FORB=7T09S.2:GOS.1560:N,B:RET.

01560  IF<F-T=B>+<<F-T=2*B>*<<A<<F+T>/2>=1>+<A<<F+T>/2>=2>>>>T.Q=1

01580  RET.

02000  FORX=UT08S.2:W=16*X:FORY=UT08S.2:Z=6*Y

02010  FORI=WT0W+15:FORJ=ZT0Z+5:SET<I,J>;N,J:N,I:N,Y:N,X

02020  RET.

02500  F,N=1T064:REA.A<N>;N,N:RET.

02511  D.99,1,99,1,99,1,99,1,1,99,1,99,1,99,1,99,99,1,99,1,99,1

02512  D.99,1,0,99,0,99,0,99,0,99,99,0,99,0,99,0,99,0,-1,99,-1,99

02513  D.-1,99,-1,99,99,-1,99,-1,99,-1,99,-1,-1,99,-1,99,-1,99

03000  FORX=0T07:FORY=0T07:P.AT<<128*X>+<8*Y>>,<8*X+Y+1>;N,Y:N,X

03015  P.AT62:RET.

03500  Y=<INT<<N-1>/8>+1>;X=N-<Y-1>*8:I=<X-1>*16:J=<Y-1>*6

```

```

03510 ON(A(N)+3)GOS. 3600, 3700, 3800, 3900, 4000
03520 RET
03600 FORZ=2T013: S: (I+Z, J+3): S: (I+Z, J+4): N, Z: RET
03700 FORZ=2T013: S: (I+Z, J+4): N, Z: RET
03800 FORZ=2T013: R: (I+Z, J+3): R: (I+Z, J+4): N, Z: RET
03900 GOS. 3700: FORZ=4T013S, 3: R: (I+Z, J+4): N, Z: RET
04000 GOS. 3600: FORZ=4T014S, 3: R: (I+Z, J+3): R: (I+Z, J+4): N, Z: RET
05000 Q=-99: FORI=1T064: IF(A(I)<1)+(A(I)>2)T. 5050
05010 IFA(I)=2T: FORB=-9T0-7S, 2: GOS. 5100: N, B
05020 F, B=7T09S, 2: GOS. 5100: N, B
05050 N, I: IFQ=-99T. 9900
05070 A(T)=A(F): A(F)=0: N=F: GOS. 3500: N=T: GOS. 3500
05075 IFT>56T: A(T)=2: N=T: GOS. 3500
05080 IF(ABS(F-T)<>14)+(ABS(F-T)<>18)T. RET
05085 N=(T+F)/2: A(N)=0: GOS. 3500: IFA(T)=1T. 5090
05087 F, B=-9T0-7S, 2: C=T+B: IF(C<0)+(C+B<0)T. 5089
05088 IF(A(C)<0)*(A(C+B)=0)T. F=T: T=C+B: G. 5070
05089 N, B
05090 FORB=7T09S, 2: C=T+B: IF(C>64)+(C+B>64)T. 5098
05091 IF(A(C)<0)*(A(C+B)=0)T. F=T: T=C+B: G. 5070
05098 N, B: RET
05100 R=-99: J=I+B: K=0: IF(J<0)+(J>64)T. 5200
05105 IFA(J)=0T: R=0: L=J
05108 IF(J+B<0)+(J+B>64)T. 5115
05110 IF(A(J+B)=0)*(A(J)<0)T. R=5: K=1: L=J+B
05115 IFR=-99T. 5200
05116 IFA(I)=2T. R=R-1: G. 5126
05120 IFI<9T. R=R-2
05123 IFL>47T. R=R+3
05125 IFL>56T. R=R+3
05126 FORA=7T09S, 2: GOS. 5250: N, A
05190 C=RND(0): IF(R>Q)+(R=Q)*(C>.4))T. Q=R: F=I: T=L
05200 RET
05250 G=I+A: H=I-A: IF(G>64)+(G+A>64)T. 5270
05255 IF((A(G)=1)+(A(G)=2))*(A(G+A)<0)T. R=R-8
05256 IFH<0T. 5300
05260 IF(A(G)<0)*(A(H)=0)T. R=R+4
05265 IF(A(H)=-2)*(A(G)=0)T. R=R+4

```

```

05270 IF (H<0)+(H-A<0)T. 5300
05280 IF ((A(H)=1)+(A(H)=2))* (A(H-A)=-2)T. R=R-8
05300 S=R: G=L+A: H=L-A: IFG>64T. 5335
05302 IFH<0T. 5307
05305 IF (A(G)<0)* (A(H)=0)+(I=H))T. R=R-10: G. 5307
05306 IF (A(H)=-2)* (A(G)=0)+(I=G))T. R=R-10
05307 IFG+A>64T. 5335
05310 IF (A(G)<0)* (A(G+A)=0)T. R=R+5+K*10
05330 IF ((A(G)=1)+(A(G)=2))* (A(G+A)<0)T. R=R+4: S=S+4
05335 IF (H<0)+(H-A<0)T. 5350
05340 IF ((A(H)=1)+(A(H)=2))* (A(H-A)=-2)T. R=R+4: S=S+4
05350 IFR>ST. R=S
05500 RET.
06000 F. Y=6T011: F. X=1T075. 2: F. Z=X*16T0X*16+15: S. (Z, Y): N. Z: N. X: N. Y
06200 F. N=9T0155. 2: GOS. 3500: N. N: F. X=0T01: F. Y=12T014: S. (X, Y): N. Y: N. X: RET.
09900 P. AT00, "YOU WIN": G. 9900

```









All Radio Shack computer programs are distributed on an "AS IS" basis without warranty.

Radio Shack shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or programs sold by Radio Shack, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer programs.

**NOTE:** Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

Refer to User's Manual for warranties. Failure to adhere to procedures set forth in User's Manual may result in the loss of warranties.

RADIO SHACK  A DIVISION OF TANDY CORPORATION

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

---

**TANDY CORPORATION**

---

AUSTRALIA

280 316 VICTORIA ROAD  
RYDALMERE N S W 2116

---

BELGIUM

PARC INDUSTRIEL DE NANINNE  
5140 NANINNE

---

U K

BILSTON ROAD WEDNESBURY  
WEST MIDLANDS WS10 7JN

\*\*\*\*\*

ADDENDUMS TO THE LEVEL I GAMES PACK

\*\*\*\*\*

THE FOLLOWING PAGES LIST THE CHANGES REQUIRED AFTER CONVERSION  
FROM LEVEL I TO LEVEL II USING THE PROGRAM CONVERSION TAPE

HAMURABI IS NOT LISTED SINCE IT  
REQUIRES NO CHANGES AFTER CONVERSION

TO QUICKEN THE INITIAL DRAWING OF 'CHECKERS'  
TRY CHANGING/ADDING THE FOLLOWING LINES :

44 DIM A(100)

```
100 GOSUB 3000 : GOSUB 2500
3000 CLS : FOR X=0TO7 : FOR Y=0TO7 : PRINT@((128*X)+(8*Y)),
      (8*X+Y+1) : NEXT Y : NEXT X
3005 A$=STRING$(8,191)
3010 FOR X=0TO6 STEP2 : FOR Y=0TO6 STEP2 : Z=((128*X)+(8*Y)) :
      PRINT@Z,8*X+Y+1 : PRINT@Z,A$ : PRINT@Z+64,A$ : PRINT@Z+
      136,A$ : IF Z<>816 PRINT@Z+200,A$ : NEXT Y,X
3012 FOR X=112TO127 : FOR Y=45TO47 : SET(X,Y) : NEXT Y,X
3015 PRINT@62,"" : RETURN
```

## \* \* \* \* \* ADDENDUM TO CHECKERS \* \* \* \* \*

```

2  FOR N=1TO64 : GOSUB 3500 : NEXT N : N=1 : PRINT@192,"WANT TO GO
    FIRST (Y/N) ";
235 A$=INKEY$ : IF A$="" THEN 235
240 PRINT@192,"      "; : GOSUB 6000; : IF A$="N"
    GOSUB 5000
1000 PRINT@184,"      "; : PRINT@184,"FROM";
1001 A$=INKEY$ : IF A$="" THEN 1001 ELSE PRINT@188,A$;
1002 B$=INKEY$ : IF B$="" THEN 1002 ELSE PRINT@189,B$;
1003 F$=A$+B$ : F=VAL(F$) : Q=1
1005 IF (F<1)+(F>64) THEN GOSUB 6000 : GOTO 1000
1006 PRINT@184," TO ";
1007 A$=INKEY$ : IF A$="" THEN 1007 ELSE PRINT@188,A$;
1008 B$=INKEY$ : IF B$="" THEN 1008 ELSE PRINT@189,B$;
1009 T$=A$+B$ : T=VAL(T$) : IF (T=0) AND (Q=0) THEN RETURN
1010 IF (T<1)+(T>64) THEN GOSUB 6000 : GOTO 1006
1200 IF (ABS(T-F) =7) OR (ABS(T-F) =9) THEN RETURN
1260 FOR B=7 TO 9 STEP2 : G=T+B : IF (G>64) OR (G+B>64) THEN 1266
1265 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+B)=0) THEN F=T : Q=0 :
    GOTO 1006
1270 FOR B=-9TO-7 STEP2 : G=T+B : IF (G<0) OR (G+B<0) THEN 1290
1280 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+B)=0) THEN F=T : Q=0 :
    GOTO 1006
1500 Q=0 : IF (A(F)>-1) OR (A(T)<0) THEN RETURN
1560 IF (F-T=B) OR ((F-T=2*B) AND ((A((F+T)/2)=1) OR (A(
    (F+T)/2)=2))) THEN Q=1
    000 IF U=1 H=8 ELSE H=6
2001 FOR X=UTOH STEP2 : W=16*X
2005 FOR Y=UTOH STEP2 : Z=6*Y
3015 PRINT@62,""; : RETURN
5000 Q=-99 : FOR I=1TO64 : IF (A(I)<1) OR (A(I)>2) THEN 5050
5080 IF (ABS(F-T)<14) AND (ABS(F-T)<18) THEN RETURN
5087 FOR B=-9TO-7 STEP2 : C=T+B : IF (C<0) OR (C+B<0) THEN 5089
5088 IF (A(C)<0) AND (A(C+B)=0) THEN F=T : T=C+B : GOTO 5070
5090 FOR B=7TO9 STEP2 : C=T+B : IF (C>64) OR (C+B>64) THEN 5098
5091 IF (A(C)<0) AND (A(C+B)=0) THEN F=T : T=C+B : GOTO 5070
5100 R=-99 : J=I+B : K=0 : IF (J<0) OR (J>64) THEN 5200
5108 IF (J+B<0) OR (J+B>64) THEN 5115
5110 IF (A(J+B)=0) AND (A(J)<0) THEN R=5 : K=1 : L=J+B
5190 C=RND(0) : IF (R>Q) OR ((R=Q) AND (C>.4)) THEN Q=R : F=I : T=L
5250 G=I+A : H=I-A : IF (G>64) OR (G+A>64) THEN 5270
5255 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+A)<0) THEN R=R-8
5260 IF (A(G)<0) AND (A(H)=0) THEN R=R+4
5265 IF (A(H)=-2) AND (A(G)=0) THEN R=R+4
5270 IF (H<0) OR (H-A<0) THEN 5300
5280 IF ((A(H)=1) OR (A(H)=2)) AND (A(H-A)=-2) THEN R=R-8
5305 IF (A(G)<0) AND ((A(H)=0)+(I=H)) THEN R=R-10 : GOTO 5307
5306 IF (A(H)=-2) AND ((A(G)=0) OR (I=G)) THEN R=R-10
5310 IF (A(G)<0) AND (A(G+A)=0) THEN R=R+5+K*10
5330 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+A)<0) THEN R=R+4 : S=S+4
5335 IF (H<0) OR (H-A<0) THEN 5350
5340 IF ((A(H)=1) OR (A(H)=2)) AND (A(H-A)=-2) THEN R=R+4 : S=S+4
5350 IF R>S R=S
6200 FOR N=9TO15 STEP2 : GOSUB 3500 : NEXT N : FOR X=0TO1 :
    FOR Y=12TO14 : SET(X,Y) : NEXT Y : NEXT X : RETURN

```

\*\*\*\*\* SPACE TAXI ADDENDUM \*\*\*\*\*

FOR THIS PROGRAM CHANGE THE SEMICOLONS (;) TO COMMAS (,) AFTER ALL PRINT@ STATEMENTS IN THE FOLLOWING LINES :

```

321
330
340                (ADD SEMICOLONS TO THE END
341                OF LINES 340 & 341.)
350
360
600
621  (ADD COMMA TO END OF LINE)
680
681
700
710
811  (ADD COMMA TO END OF LINE)
1020
1100
1120

```

ADD THE FOLLOWING CHANGES :

```

410 IF H<0 THEN C=C-1 : M=M-1 : GOTO 430
430 IF V<0 THEN D=D-1 : N=N-1 : GOTO 441
450 IF (C=1) OR (C=126) OR (D=0) CLS : GOTO 780
455 IF (D=30) AND ((C<A) OR (C>A+7)) AND (V<=2) THEN V=0 :
    GOSUB 1100 : GOTO 320
460 IF (POINT(C-2,D+1)) OR (POINT(C+2,D+1)) GOTO 780
470 IF (D=30) AND (V<=2) V=0 : GOTO 800
480 IF (D=31) OR (D=30) GOTO 780
720 INPUT "> PLAY AGAIN (Y/N) "; Z$ : IF Z$="Y" F=80 :
    CLS : GOTO 2
730 IF Z$ <> "N" GOTO 720
810 FOR Z=A TO A+5 : IF (C-1=Z) OR (C+1=Z) GOTO 860

```

\*\*\*\*\* STAR PILOT ADDENDUM \*\*\*\*\*

ADD OR CHANGE THE FOLLOWING LINES :

```

99 DIM A(50)
1000 GOSUB 3999
3999 D$=INKEY$ : IF D$<>" "CLS : ELSE GOTO 4000
8020 INPUT "DO YOU WANT ANOTHER SIMULATION",R$
8030 IF R$<>"N" GOTO 100
8510 (CHANGE 'GOTO 8015' TO 'GOTO 8020')

```

DELETE LINE 8015

\* \* \* \* \* TIC TAC TOE \* \* \* \* \*

```

CLS : PRINT@471, "ENTER FIRST NAME"; INPUT B$ : J=0 : U=0 : V=0 :
W=0 : DIM A(50)
98 PRINT TAB(22), B$; "SELECT SQUARE"; : PRINT@00, "?";
100 Q$=INKEY$ : IF Q$="" THEN 100 ELSE Q=VAL(Q$) : PRINT@3, Q; :
PRINT@0, " ";
101 IF (Q>0) AND (Q<10) THEN 103
102 PRINT "INVALID PLAY" : FOR T=1TO1000 : NEXT T : PRINT@0, " "; :
GOTO98
199 A(R)=4: PRINT"";
666 FOR R=1TO9 : IF (A(R)=M) OR (A(R)=S) Q=R : GOTO 1000
680 IF A(R)=S THEN 199
4000 Q=(A(M))-T : IF Q=-2 Q=7
4001 IF Q=-1 Q=8
4002 IF Q=0 Q=9
4003 IF A(Q)=0 A(Q)=9 : GOTO 8500
4004 IF M=17 M=13
4005 IF M=19 M=17
4006 IF M=15 M=19
7787 IF Y=1 Y=2 : X=2 : GOTO7010
7788 IF Y=2 Y=3 : X=8 : Z=2 : GOTO 7800
7789 IF Y=3 Y=4 : X=2 : GOTO 7010
7790 IF Y=4 Y=5 : X=8 : Z=3 : GOTO7800
7791 IF Y=5 Y=6 : X=2 : GOTO 7010
7792 IF Y=6 Y=7 : X=8 : Z=4 : GOTO 7800
7793 IF Y=7 Y=8 : X=2 : GOTO 7010
8900 IF (A+B+C=X) OR (A+D+G=X) OR (A+E+I=X) OR (B+E+H=X) THENZ=1
8910 IF (C+F+I=X) OR (D+E+F=X) OR (G+H+I=X) THEN Z=1
8912 IF (A+B+C=X) OR (A+D+G=X) OR (A+E+I=X) OR (B+E+H=X) OR
(C+E+G=X) THEN Z=Z+2
8913 IF (C+F+I=X) OR (D+E+F=X) OR (G+H+I=X) THEN Z=Z+2
8915 IF ((Z=2) OR (Z=4)) THEN 8931
8916 IF (Z=3) OR (Z=5) THEN 10020

```

\* \* \* \* \* DRAW I, DRAW II, DRAW III \* \* \* \* \*

DRAW I ----- NO CHANGES

```

DRAW II ----- ADD :
      4 DIM A(MEM/4-1)
      70 A=0 : K=INT(MEM/4-1)
      80 I=0TO(MEM/4-1) : A(I)=0 : NEXT I
      233 IF (S=0) AND (POINT(X,Y)) K=K+1
      630 IF POINT (I,J) A(K)=(I*100+J) : K=K+1

```

```

DRAW III ----- ADD :
      1 DIM A(3000)

```




# Radio Shack®

## Games Pack I

For Level-I TRS-80™ Systems  
With 4K RAM or More



On 3  
Cassettes

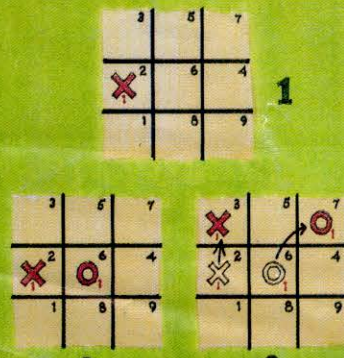


### Space Taxi

You're the pilot of a rocket craft. Your task: Fly over a mountain, but stay under a radiation shield, and land safely in the spaceport. A challenge for space pilots of all ages!

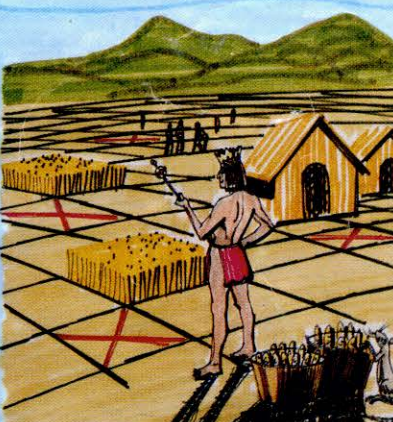
### Random Tic-Tac-Toe

A challenging twist on an old game. After each move, the Xs and Os are shifted to the next numbered squares on the board. To win, you must plan several moves ahead. Try to "outsmart" the computer!



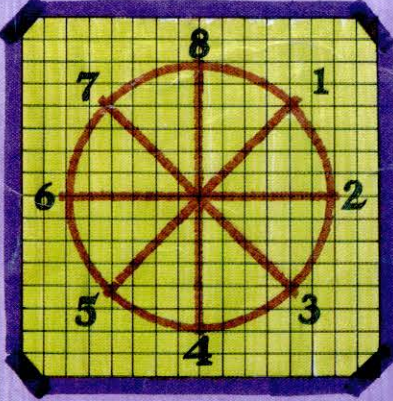
### Hammurabi

Your chance to rule a nation! Start with a given amount of assets and manage your "kingdom" by planting the right amount of grain, buying and selling "acres" of land. Your success depends on careful planning!



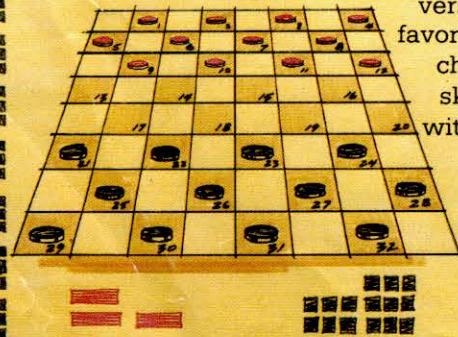
### Draw

This fascinating game allows you to "draw" graphic designs on TRS-80's video monitor, store them and recall them, if desired. Useful and entertaining!



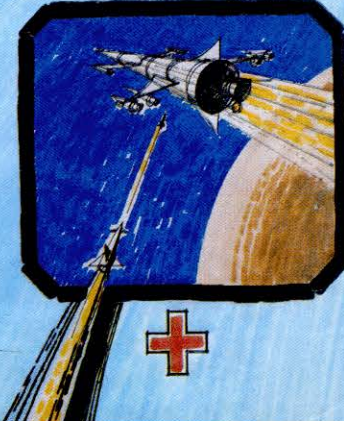
### Checkers

Here's another updated version of an old favorite! Test your checker-playing skill in a match with your TRS-80 computer. Exciting fun for everyone!



### Star Pilot

Pilot a fighter plane and shoot down enemy aircraft before they take evasive action. This fast-paced game allows you to fire at a constantly moving target. Hours of entertainment!





\* \* \* \* \*

ADDENDUMS TO THE LEVEL I GAMES PACK

\* \* \* \* \*

THE FOLLOWING PAGES LIST THE CHANGES REQUIRED AFTER CONVERSION  
FROM LEVEL I TO LEVEL II USING THE PROGRAM CONVERSION TAPE

HAMURABI IS NOT LISTED SINCE IT  
REQUIRES NO CHANGES AFTER CONVERSION

TO QUICKEN THE INITIAL DRAWING OF 'CHECKERS'  
TRY CHANGING/ADDING THE FOLLOWING LINES :

```
100  GOSUB 3000 : GOSUB 2500
3000  CLS : FOR X=0TO7 : FOR Y=0TO7 : PRINT@((128*X)+(8*Y)),
      (8*X+Y+1); : NEXT Y : NEXT X
3005  A#=STRING$(8,191)
3010  FOR X=0TO6 STEP2 : FOR Y=0TO6 STEP2 : Z=((128*X)+(8*Y)) :
      PRINT@Z,8*X+Y+1; : PRINT@Z,A#; : PRINT@Z+64,A#; : PRINT@Z+
      136,A#; : IF Z<>816 PRINT@Z+200,A#; : NEXT Y,X
3012  FOR X=112TO127 : FOR Y=45TO47 : SET(X,Y) : NEXT Y,X
3015  PRINT@62,""; : RETURN
```

## \* \* \* \* \* ADDENDUM TO CHECKERS \* \* \* \* \*

```

230 FOR N=1TO64 : GOSUB 3500 : NEXT N : N=1 : PRINT@192,"WANT TO GO
    FIRST (Y/N) ";
235 A$=INKEY$ : IF A$="" THEN 235
240 PRINT@192," " ; : GOSUB 6000; : IF A$="N"
    GOSUB 5000
1000 PRINT@184," " ; : PRINT@184,"FROM";
1001 A$=INKEY$ : IF A$="" THEN 1001 ELSE PRINT@188,A$;
1002 B$=INKEY$ : IF B$="" THEN 1002 ELSE PRINT@189,B$;
1003 F$=A$+B$ : F=VAL(F$) : Q=1
1005 IF (F<1)+(F>64) THEN GOSUB 6000 : GOTO 1000
1006 PRINT@184," TO ";
1007 A$=INKEY$ : IF A$="" THEN 1007 ELSE PRINT@188,A$;
1008 B$=INKEY$ : IF B$="" THEN 1008 ELSE PRINT@189,B$;
1009 T$=A$+B$ : T=VAL(T$) : IF (T=0) AND (Q=0) THEN RETURN
1010 IF (T<1)+(T>64) THEN GOSUB 6000 : GOTO 1006
1200 IF (ABS(T-F)=7) OR (ABS(T-F)=9) THEN RETURN
1260 FOR B=7 TO 9 STEP2 : G=T+B : IF (G>64) OR (G+B>64) THEN 1266
1265 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+B)=0) THEN F=T : Q=0 :
    GOTO 1006
1270 FOR B=-9TO-7 STEP2 : G=T+B : IF (G<0) OR (G+B<0) THEN 1290
1280 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+B)=0) THEN F=T : Q=0 :
    GOTO 1006
1500 Q=0 : IF (A(F)>-1) OR (A(T)<0) THEN RETURN
1560 IF (F-T=B) OR ((F-T=2*B) AND ((A((F+T)/2)=1) OR (A(
    (F+T)/2)=2))) THEN Q=1
2000 IF U=1 H=8 ELSE H=6
2001 FOR X=UTOH STEP2 : W=16*X
2005 FOR Y=UTOH STEP2 : Z=6*Y
3015 PRINT@62,""; : RETURN
5000 Q=-99 : FOR I=1TO64 : IF (A(I)<1) OR (A(I)>2) THEN 5050
5060 IF (ABS(F-T)<0)14) AND (ABS(F-T)<0)18) THEN RETURN
5087 FOR B=-9TO-7 STEP2 : C=T+B : IF (C<0) OR (C+B<0) THEN 5089
5088 IF (A(C)<0) AND (A(C+B)=0) THEN F=T : T=C+B : GOTO 5070
5090 FOR B=7TO9 STEP2 : C=T+B : IF (C>64) OR (C+B>64) THEN 5098
5091 IF (A(C)<0) AND (A(C+B)=0) THEN F=T : T=C+B : GOTO 5070
5100 R=-99 : J=I+B : K=0 : IF (J<0) OR (J>64) THEN 5200
5108 IF (J+B<0) OR (J+B>64) THEN 5115
5110 IF (A(J+B)=0) AND (A(J)<0) THEN R=5 : K=1 : L=J+B
5190 C=RND(0) : IF (R>0) OR ((R=0) AND (C>.4)) THEN Q=R : F=I : T=L
5250 G=I+A : H=I-A : IF (G>64) OR (G+A>64) THEN 5270
5255 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+A)<0) THEN R=R-8
5260 IF (A(G)<0) AND (A(H)=0) THEN R=R+4
5265 IF (A(H)=-2) AND (A(G)=0) THEN R=R+4
5270 IF (H<0) OR (H-A<0) THEN 5300
5280 IF ((A(H)=1) OR (A(H)=2)) AND (A(H-A)=-2) THEN R=R-8
5305 IF (A(G)<0) AND ((A(H)=0)+(I=H)) THEN R=R-10 : GOTO 5307.
5306 IF (A(H)=-2) AND ((A(G)=0) OR (I=G)) THEN R=R-10
5310 IF (A(G)<0) AND (A(G+A)=0) THEN R=R+5+K*10
5330 IF ((A(G)=1) OR (A(G)=2)) AND (A(G+A)<0) THEN R=R+4 : S=S+4
5335 IF (H<0) OR (H-A<0) THEN 5350
5340 IF ((A(H)=1) OR (A(H)=2)) AND (A(H-A)=-2) THEN R=R+4 : S=S+4
5350 IF R>S R=S
6200 FOR N=9TO15 STEP2 : GOSUB 3500 : NEXT N : FOR X=0TO1 :
    FOR Y=12TO14 : SET(X,Y) : NEXT Y : NEXT X : RETURN

```

## \* \* \* \* \* SPACE TAXI ADDENDUM \* \* \* \* \*

FOR THIS PROGRAM CHANGE THE SEMICOLONS (;) TO COMMAS (,) AFTER ALL PRINT\$ STATEMENTS IN THE FOLLOWING LINES :

```

321
330
340                (ADD SEMICOLONS TO THE END
341                OF LINES 340 & 341.)
350
360
600
621  (ADD COMMA TO END OF LINE)
680
681
700
710
811  (ADD COMMA TO END OF LINE)
1020
1100
1120

```

ADD THE FOLLOWING CHANGES :

```

410 IF H<0 THEN C=C-1 : M=M-1 : GOTO 430
430 IF V<0 THEN D=D-1 : N=N-1 : GOTO 441
450 IF (C=1) OR (C=126) OR (D=0) CLS : GOTO 780
455 IF (D=30) AND ((C<A) OR (C>A+7)) AND (V<=2) THEN V=0 :
    GOSUB 1100 : GOTO 320
460 IF (POINT(C-2,D+1)) OR (POINT(C+2,D+1)) GOTO 700
470 IF (D=30) AND (V<=2) V=0 : GOTO 800
480 IF (D=31) OR (D=30) GOTO 700
720 INPUT "> PLAY AGAIN (Y/N) "; Z$ : IF Z$="Y" F=80 :
    CLS : GOTO 2
730 IF Z$ <> "N" GOTO 720
810 FOR Z=A TO A+5 : IF (C-1=Z) OR (C+1=Z) GOTO 860

```

## \* \* \* \* \* STAR PILOT ADDENDUM \* \* \* \* \*

ADD OR CHANGE THE FOLLOWING LINES :

```

99 DIM A(50)
1000 GOSUB 3999
3999 D$=INKEY$ : IF D$<>"" CLS : ELSE GOTO 4000
8020 INPUT "DO YOU WANT ANOTHER SIMULATION",R$
8030 IF R$<>"N" GOTO 100
8510 (CHANGE 'GOTO 8015' TO 'GOTO 8020')

```

DELETE LINE 8015

\* \* \* \* \* T I C T A C T O E \* \* \* \* \*

```

2 CLS : PRINT@471,"ENTER FIRST NAME" : INPUT B$ : J=0 : U=0 :
  V=0 : W=0 : DIM A(50)
98 PRINTTAB(22),B$;"SELECT SQUARE" : PRINT@0,"?";
100 Q#=INKEY$ : IF Q#="" THEN 100 ELSE Q=VAL(Q#) : PRINT@3,Q) :
  PRINT@0," ";
101 IF (Q>0) AND (Q<10) THEN 103
102 PRINT "INVALID PLAY" : FOR T=1TO1000 : NEXT T : PRINT@0," " :
  GOT098
199 A(R)=4 : PRINT"";
666 FOR R=1TO9 : IF (A(R)=M) OR (A(R)=S) Q=R : GOT0 1000
680 IF A(R)=S THEN 199
4000 Q=(A(M))-T : IF Q=-2 Q=7
4001 IF Q=-1 Q=8
4002 IF Q=0 Q=9
4003 IF A(Q)=0 A(Q)=9 : GOT0 8500
4004 IF M=17 M=13
4005 IF M=19 M=17
4006 IF M=15 M=19
7787 IF Y=1 Y=2 : X=2 : GOT0 7010
7788 IF Y=2 Y=3 : X=8 : Z=2 : GOT0 7800
7789 IF Y=3 Y=4 : X=2 : GOT0 7010
7790 IF Y=4 Y=5 : X=8 : Z=3 : GOT0 7800
7791 IF Y=5 Y=6 : X=2 : GOT0 7010
7792 IF Y=6 Y=7 : X=8 : Z=4 : GOT0 7800
7793 IF Y=7 Y=8 : X=2 : GOT0 7010
8900 IF (A+B+C=X) OR (A+D+G=X) OR (A+E+I=X) OR (B+E+H=X) THEN Z=1
8910 IF (C+F+I=X) OR (D+E+F=X) OR (G+H+I=X) THEN Z=1
8912 IF (A+B+C=X) OR (A+D+G=X) OR (A+E+I=X) OR (B+E+H=X) OR
  (C+E+G=X) THEN Z=Z+2
8913 IF (C+F+I=X) OR (D+E+F=X) OR (G+H+I=X) THEN Z=Z+2
8915 IF ((Z=2) OR (Z=4)) THEN 8931
8916 IF (Z=3) OR (Z=5) THEN 10020

```

\* \* \* \* \* DRAW I, DRAW II, DRAW III \* \* \* \* \*

DRAW I ----- NO CHANGES

DRAW II ----- ADD THE FOLLOWING :

```

  4 DIM A(MEM/4-1)
  70 A=0 : K=INT(MEM/4-1)
  80 FOR I=0TO(MEM/4-1) : A(I)=0 : NEXT I
  233 IF (S=0) AND (POINT(X,Y)) K=K+1
  630 IF POINT(I,J) A(K)=(I=100+J) : K=K+1

```

\*\* IF OUT OF MEMORY ERROR OCCURS, CHANGE  
 LINE 4 TO READ : 'DIM A(3000)'

DRAW III ----- ADD THE FOLLOWING :

```

  1 DIM A(3000)

```