DECUS
PROGRAM LIBRARY

DECUS NO.	5/8-54
TITLE	TIC-TAC-TOE LEARNING PROGRAM - T^3
AUTHOR	Michael Green
COMPANY	Stevens Institute of Technology
DATE	December, 1966
FORMAT	
Classification:	Games and Demonstrators

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

DECUS NO. 5/8-54

T^3 - Tic Tac Toe Learning Program

Operating Instructions:

1. Load the FORTRAN Operating System (DEC-08-AFA3-PB) using the binary loader.

2. Place FORTRAN Object tape for Tic-Tac-Toe in the reader. Load 200_8 and start with switch 1 up if loading from the high-speed reader, down if from the ASR-33.

3. Load the board printout routine using the binary loader.

4. Load 201_8 and start with all switches down.

Playing Instructions:

The program will type out YOUR MOVE= . Type in a digit from 1 to 9 then type a space. If you enter the wrong digit, type a rubout instead of a space and then reenter the digit and follow with a space. If the move is illegal, the program will type out YOU GOOFED, TRY AGAIN . It will ask for another move and you can enter a legal one. The machine will make its move and then type out the board. Your moves are X's, the machine's are 0's. If a move ends a game, the program will type YOU WIN, YOU LOSE or TIE GAME depending on the outcome. It will then start a new game.

Example of board printout:

```
1I2I3    XI I
-----    -----
4I5I6    0I0IX
-----    -----
7I8I9    XIXI0
```

The X's are in positions 1, 6, 7 and 8. The 0's are in positions 4, 5 and 9.

To reset the learning table to empty, deposit a 0 in location 7111_8. The program adds new moves to the learning table whenever it loses a game. To save any additions, punch out the move count at location 7111_8 and the learning table at locations 7234_8 through 7543_8. It can be reloaded after step 3 of the operating instructions.

Program Operation:

1. The program checks the move for validity (between 1 and 9).

2. The program checks to see if the move has been made before.

3. The move is converted to a number in a 3 by 3 magic square.

4. If any three opponent's moves add up to 15, the machine has lost the game. It prints out the board and YOU WIN and then stores in the game table all moves up to the last move that it had to guess.

5. If any two machine moves and a move not already made add up to 15, the machine makes that move and wins. It prints out the board and YOU LOSE.

6. If any two opponent's moves and a move not already made add up to 15, the machine makes that move and blocks an attempt to get three in a row. It prints out the board and asks for the next move.

7. If none of the above occur, the program must make an educated guess. It compares all moves up to a guess to the stored moves in the game table. If any stored game matches the current game, the program tries another guess. If the program has tried all unoccupied spaces on the board, it sets back the last move guessed counter so that when the machine loses it will try a new guess one move back. The guesses are made in the following order: 1, 2, 3, 4, 5, 6, 7, 8, 9.

8. If all 9 spaces have been used without a loss or win, the program prints out the board and TIE GAME.

The various variables and what they are used for:

IG the current game in magic square values.
MA the magic square.
JG the 9 squares of the board; 0 is space, 1 is X, 2 is O.
IS the game table; contains games in magic square values separated by -1.
MOVE the current move number.
LM the number of moves and -1's stored in IS.
LAST the number of the last move that the program had to guess.

The other variables are used for temporary storage.

PAUSE 5 is used to call the board printout subroutine in locations 7000_8 through 7073_8. The routine is straightforward and uses JG for the current board and locations 15_8, 16_8, and 17_8 for temporary storage. PAUSE 5 is an effective JMS 5.

```
C;   TIC-TAC-TOE LEARNING PROGRAM
C;   INITIALIZE LM TO ZERO
     DIMENSION IG(9),MA(9),JG(9),IS(200)
     MA(1)=8
     MA(2)=1
     MA(3)=6
     MA(4)=3
     MA(5)=5
     MA(6)=7
     MA(7)=4
     MA(8)=9
     MA(9)=2
C;   START OF GAME
1;   MOVE=-1
     DO 6 I=1,9
     JG(I)=0
6;   CONTINUE
C;   START OF MOVE
2;   MOVE=MOVE+2
     TYPE 900
900; FORMAT(/,"YOUR MOVE= ")
     ACCEPT 901,I
901; FORMAT(I)
C;   CHECK CORRECT RANGE
     GO TO (5,5,5,5,5,5,5,5,5),I
3;   TYPE 902
902; FORMAT("  YOU GOOFED, TRY AGAIN",/)
     MOVE=MOVE-2
     GO TO 2
5;   IG(MOVE)=MA(I)
C;   CHECK NOT USED BEFORE
     IF (MOVE-1) 23,24,23
23;  IT1=MOVE-1
     DO 7 J=1,IT1
     IF (IG(MOVE)-IG(J)) 7,3,7
7;   CONTINUE
     JG(I)=1
C;   CHECK IF LOSS OR TIE
     IF (MOVE-5) 26,25,25
25;  IT1=MOVE-2
     IT2=MOVE-4
     DO 8 I=1,IT2,2
     IT3=I+2
     DO 8 J=IT3,IT1,2
     IT4=J+2
     DO 8 K=IT4,MOVE,2
     IF (IG(I)+IG(J)+IG(K)-15) 8,100,8
8;   CONTINUE
     IF (MOVE-9) 9,101,9
C;
C;
C;
C;
C;
C;
```

```
C;
C;
C;
C;
C;   CHECK IF POSSIBLE WIN
9;   IT1=MOVE-1
     IT2=MOVE-3
     DO 10 I=2,IT2,2
     IT3=I+2
     DO 10 J=IT3,IT1,2
     DO 11 K=1,MOVE
     IF (IG(I)+IG(K)+IG(J)-15) 11,10,11
11;  CONTINUE
     K=15-IG(I)-IG(J)
     GO TO (27,27,27,27,27,27,27,27,27),K
     GO TO 10
27;  IG(MOVE+1)=K
     JG(MA(10-IG(MOVE+1)))=2
     PAUSE 5
     TYPE 905
     GO TO 1
905; FORMAT(/,"YOU LOSE",/,/)
10;  CONTINUE
C;   CHECK IF POSSIBLE LOSS
26;  IT1=MOVE-2
     DO 12 I=1,IT1,2
     IT2=I+2
     DO 12 J=IT2,MOVE,2
     DO 13 K=1,MOVE
     IF (IG(I)+IG(J)+IG(K)-15) 13,12,13
13;  CONTINUE
     K=15-IG(I)-IG(J)
     GO TO (28,28,28,28,28,28,28,28,28),K
     GO TO 12
28;  IG(MOVE+1)=K
103; JG(MA(10-IG(MOVE+1)))=2
     PAUSE 5
     GO TO 2
12;  CONTINUE
     IF (MOVE-7) 14,101,14
24;  JG(I)=1
C;   MUST MAKE EDUCATED GUESS
14;  LAST=MOVE+1
     LOOK=0
15;  DO 16 I=1,9
     DO 17 J=1,MOVE
     IF (MA(I)-IG(J)) 17,16,17
17;  CONTINUE
     IG(MOVE+1)=MA(I)
     IF (LM) 18,103,18
18;  K=0
     IF (LOOK-LM) 19,103,19
19;  K=K+1
     LOOK=LOOK+1
     IF (IS(LOOK)) 16,16,20
20;  IF (IG(K)-IS(LOOK)) 22,19,22
21;  LOOK=LOOK+1
22;  IF (IS(LOOK)) 18,18,21
16;  CONTINUE
     LAST=LAST-2
     GO TO 103
C;
C;
C;
C;
```

```
 C;
 C;
 C;
 C;
 100;PAUSE 5
     TYPE 903
 903;FORMAT(/,"YOU WIN",/,/)
 C;   SAVE LOST GAME
     DO 109 I=1,LAST
     IF (I+LM-200) 110,1,110
 110;IS(I+LM)=IG(I)
 109;CONTINUE
     IS(LM+LAST+1)=-1
     LM=LM+LAST+1
     GO TO 1
 101;PAUSE 5
     TYPE 904
 904;FORMAT(/,"TIE GAME",/,/)
     GO TO 1
     END
F
```

```
IG     7566
MA     7555
JG     7544
IS     7234
MOVE   7211
I      7207
IT1    7200
J      7176
IT2    7171
IT3    7165
IT4    7162
K      7160
LAST   7121
LOCK   7117
LM     7111
  6764  7100
```

```
                 *7000
7000    7300     B,   CLA CLL
7001    1232          TAD PG
7002    3015          DCA AA
7003    1234          TAD PL
7004    3016          DCA AB
7005    1246          TAD CR
7006    4251          JMS OT
7007    1247          TAD CR+1
7010    4251          JMS OT
7011    1247          TAD CR+1
7012    4251          JMS OT
7013    1416     L1,  TAD I AB
7014    7450          SNA
7015    5405          JMP I 5
7016    3017          DCA AC
7017    1415          TAD I AA
7020    1231          TAD PC
7021    3233          DCA T
7022    1633          TAD I T
7023    4251          JMS OT
7024    1417     L2,  TAD I AC
7025    7450          SNA
7026    5213          JMP L1
7027    4251          JMS OT
7030    5224          JMP L2
7031    7057     PC,  CH
7032    7543     PG,  7543
7033    0000     T,   Ø
7034    7061     PL,  LT-1
7035    0311     LI,  311
7036    0000          Ø
7037    0215     CM,  215
7040    0212          212
7041    0255          255
7042    0255          255
7043    0255          255
7044    0255          255
7045    0255          255
7046    0215     CR,  215
7047    0212          212
7050    0000          Ø
7051    0000     OT,  Ø
7052    6041          TSF
7053    5252          JMP .-1
7054    6046          TLS
7055    7200          CLA
7056    5651          JMP I OT
```

```
7057   0240   CH, 240
7060   0330        330
7061   0317        317
7062   7034   LT, LI-1
7063   7034       LI-1
7064   7036       CM-1
7065   7034       LI-1
7066   7034       LI-1
7067   7036       CM-1
7070   7034       LI-1
7071   7034       LI-1
7072   7045       CR-1
7073   0000       0
              AA=15
              AB=16
              AC=17
              *404
0404   7000       NOP
              *5
0005   0000       0
0006   5407       JMP I .+1
0007   7000       B
              *7111 /LM
7111   0206       206 /MOVE COUNT
              *7234 /IS
7234   0010       10 /STORED GAMES
7235   0001       1
7236   7777       -1
7237   0010       10
7240   0006       6
7241   7777       -1
7242   0002       2
7243   0010       10
7244   0004       4
7245   0001       1
7246   7777       -1
```

```
7247   0003     3
7250   0010     10
7251   0006     6
7252   0001     1
7253   7777     -1
7254   0011     11
7255   0010     10
7256   0006     6
7257   0001     1
7260   7777     -1
7261   0007     7
7262   0010     10
7263   0011     11
7264   0001     1
7265   7777     -1
7266   0001     1
7267   0010     10
7270   0003     3
7271   0006     6
7272   7777     -1
7273   0010     10
7274   0003     3
7275   7777     -1
7276   0004     4
7277   0010     10
7300   7777     -1
7301   0006     6
7302   0010     10
7303   7777     -1
7304   0004     4
7305   0001     1
7306   7777     -1
7307   0005     5
7310   0010     10
7311   0002     2
7312   0001     1
7313   7777     -1
7314   0002     2
7315   0010     10
7316   7777     -1
7317   0002     2
7320   0001     1
7321   7777     -1
7322   0002     2
7323   0006     6
7324   7777     -1
7325   0001     1
7326   0010     10
7327   0002     2
7330   0006     6
7331   7777     -1
7332   0006     6
7333   0001     1
7334   7777     -1
```

```
7335   0004        4
7336   0006        6
7337   7777       -1
7340   0004        4
7341   0003        3
7342   7777       -1
7343   0006        6
7344   0003        3
7345   7777       -1
7346   0006        6
7347   0005        5
7350   0004        4
7351   0010       10
7352   7777       -1
7353   0002        2
7354   0003        3
7355   7777       -1
7356   0010       10
7357   0005        5
7360   0011       11
7361   0001        1
7362   7777       -1
7363   0010       10
7364   0005        5
7365   0011       11
7366   0006        6
7367   7777       -1
7370   0010       10
7371   0005        5
7372   0011       11
7373   0003        3
7374   0007        7
7375   0001        1
7376   7777       -1
7377   0011       11
7400   0010       10
7401   7777       -1
7402   0011       11
7403   0001        1
7404   0006        6
7405   0010       10
7406   7777       -1
7407   0011       11
7410   0001        1
7411   0010       10
7412   0006        6
7413   7777       -1
7414   0007        7
7415   0010       10
7416   0004        4
7417   0001        1
7420   7777       -1
```

```
7421  0007     7
7422  0010     10
7423  0004     4
7424  0006     6
7425  0001     1
7426  0003     3
7427  7777     -1
7430  0011     11
7431  0001     1
7432  0006     6
7433  0003     3
7434  7777     -1
7435  0004     4
7436  0005     5
7437  0006     6
7440  0010     10
7441  7777     -1

AA         0015
AB         0016
AC         0017
B          7000
CH         7057
CM         7037
CR         7046
LI         7035
LT         7062
L1         7013
L2         7024
OT         7051
PC         7031
PG         7032
PL         7034
T          7033
```