P?S

PROGRAM LIBRARY

P?S NUMBER	P?S-Ø8-2.5D-D
TITLE	3D Tic-Tac-Toe (PDP-8 light pen version)
AUTHOR	Jim Roth
DATE	April 1971
SECURITY CLASSIFICATION	Declassified

P?S

3D Tic-Tac-Toe

Abstract

This program allows the user to play a game of three-dimensional Tic-Tac-Toe (on a 4 by 4 by 4 board) against the computer. The board is displayed on a CRT and the player makes moves using a light pen. The computer rarely loses.

Minimum equipment needed: PDP-8, 8/I, or 8/E with EAE, teletype

Digital-to-analog converter, type AAØ1A

Analog-to-digital converter, type AFØ1A

Light pen

Oscilloscope

(optional) Music register, type P?S-MR1

PPS

3D Tic-Tac-Toe

Acknowledgement

P?S

3D Tic-Tac-Toe

Preface

This program obsoletes versions 2.1D through 2.4D which were written by other P?S members in FORTRAN II, FORTRAN IV, CPS, and APL respectively.

Another version, which uses teletype input instead of light pen control is not currently ready for distribution.

A version which runs on the PDP-12 is working and will shortly be submitted to DECUS.

3D Tic-Tac-Toe

I. Description.

Three-dimesnional Tic-Tac-Toe is a game very similar to ordinary Tic-Tac-Toe. It is played by two players, "X" and "O". It is played on a 4 by 4 by 4 cube which has been partitioned up into 64 small unit cubes by planes parallel to the sides of the large cube. In any of these 64 "compartments", players may place either X's or O's. The object of the game is for one player to get 4 of his markers in a straight line.

Players alternate putting their markers into empty "compartments". The game ends when one player has succeeded in getting 4 of his markers to line up in any direction, be it vertical, horizontal, up and down, or along some diagonal (either a diagonal of some planar cross section, or a major diagonal of the cube). All together, there are 72 ways of winning. If neither player gets 4 in a row anywhere, the game is a draw.
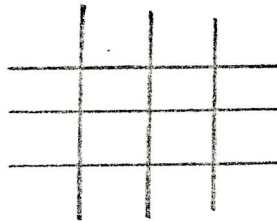
Since it is difficult to draw a three-dimensional cube partitioned in this manner, it is convenient to slice the cube up into 4 consecutive planes (each 4 by 4) and draw these 4 planes side by side (in order) on a flat two dimensional surface as shown below.
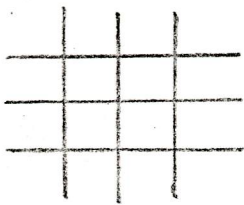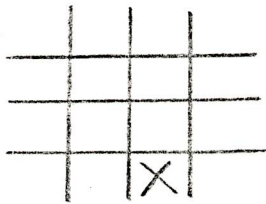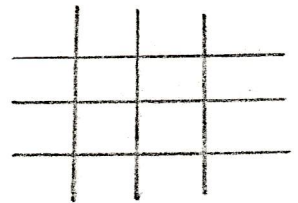
plane 1	plane 2	plane 3	plane 4

The planes are numbered in order from 1 to 4. When describing a particular square, its "coordinates" are usually specified by first stating which plane the square is in, and then giving the row and column numbers for that square in that plane. The "X" in the above diagram is in plane 2 row 4 column 3.

We show some examples of winning combinations in the following diagram. Four squares which are in a straight line are indicated by placing the same digit in them.
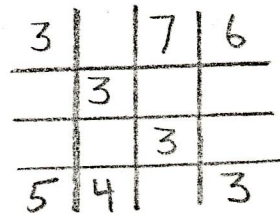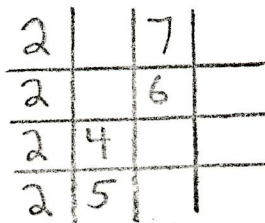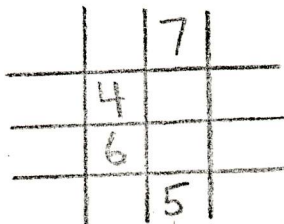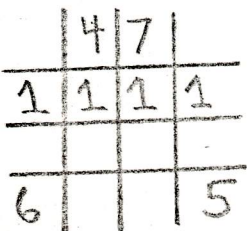
3D Tic tac-Toe

II. Distribution and Loading.

3D Tic-Tac-Toe is distributed both on paper tape and on DECtape.
The source is split up into 8 parts, referred to as TTTØ through TTT7.
That is the name of the file which the corresponding program segment
resides in on the DECtape. The 8 paper tapes for the source are labelled
P?S-Ø8-2.5D-PA/Ø through P?S-Ø8-2.5D-PA/7 respectively. The binary file
on the DECtape is called TTT. The paper tape copy (in DEC binary format)
is labelled P?S-Ø8-2.5D-PB. The DECtape contains the RL Monitor System   LA
(WCFMPG version). A source listing is also available, labelled P?S-Ø8-2.5D-S.
The binary can be obtained from the source by assembling TTTØ through TTT7
in order.

The starting address for the program is location 2ØØ. (It can also
be restarted from this address.) The program can be loaded into core and
started from the RL Monitor System by the command,

RUN BIN=2ØØ,TTT   .

3D Tic-Tac-Toe

III. Setup and operation

    This program requires a 4K PDP-8, 8/I, or 8/E with EAE, a digital to analog converter type AA01A, an analog-to-digital converter type AF01A, a light pen, and an oscilloscope. Optionally, it can also use a music register, type P?S-MR1 with amplifier and speaker.

    Connect the light pen output (amplified if necessary) to A/D channel 0. Connect the oscilloscope horizontal input to D/A channel 0; and connect the oscilloscope vertical input to D/A channel 1.

    If a music register is available, connect its output to a speaker using an amplifier if necessary. The device code for your music register must be put into the program source before assembling. See the source listing for details. If this option is used, the computer will play notes while it is "thinking" which indicate by pitch what depth level of analysis the computer look-ahead is currently at.

    The program starts at location 200. The board (initially empty) will be displayed on the scope with a small dot in empty squares. The computer is always "O" and the player "X", but you may decide who is to go first.

    To make a move, you point the light pen at the dot in the center of the empty square you wish to move in. When the computer recognizes this, an "X" will appear there. It will then make its move which will result in an "O" appearing on the board. This "O" will appear brighter than the others so that you will more easily notice what move the computer made. If either player wins, the 4 X's or O's which are in a straight line will all light up brightly, showing you along which line the win was made. At this point, if you attempt to make another move, a new game will be started.

    The following special commands can be entered via the teletype during the course of a game:

R - restarts the program (resigns)
F - forfeits a move
CTRL/D - reads the switch register and uses this number to set a new depth
    of recursion for the internal logic of the program. This depth is
    initially set to about 16 decimal. It should not be reset to too
    small a value, and should not be changed at all for tournament play.

You can tell the computer to go first by initially hitting F to forfeit your first move; however we warn you that the computer has only been beaten once when it has gone first.

    You can also set up an initial, or test position, to start with by the following controls:

CTRL/X - changes mode so that pointing light pen at square places an "X" there.
CTRL/O - changes mode so that pointing light pen at square places an "O" there.
CTRL/E - changes mode so that pointing light pen at occupied square
    removes the marker in that square. This mode should not be used
    during a game.
space - resets mode to the standard mode so that a game can be played.

    Pointing the light pen at a random light will not upset the program.