**Project 2: Simple Games using AI**

**Due date: October 22. (125 points)**

1. **Description:** In this project, you will develop an agent that proposes the next possible move in a TIC-TAC-TOE game using minimax adversarial search described in Week5.

The output of the project will look similar as follows:

First, you will need to display a table to start the game, you can use “-“ and “|” to draw the table in text mode:

--------------------------------------------------

| | | |

--------------------------------------------------

| | | |

--------------------------------------------------

| | | |

--------------------------------------------------

Then you will prompt the user to enter their move first. Each move is described by a pair (x,y) with x being the index of row, and y being the index of columns. For simplicity purpose, we assume that x has values in [0,2] and y has values in [0,2].

For example, the user enter: 0 1

The program redraws the table to reflect the user’s move: an X is displayed in the node corresponds to row 0 and column 1, as shown below:

--------------------------------------------------

| | x | |

--------------------------------------------------

| | | |

--------------------------------------------------

| | | |

--------------------------------------------------

The program can sleep for a minute or so, then print out the system’s move based on the calculation of minimax algorithm that we learned in Week 5. The table can then be re-drawn as follows:

--------------------------------------------------

| O | x | |

--------------------------------------------------

| | | |

--------------------------------------------------

| | | |

--------------------------------------------------

You will repeat the above steps until the game is over. There are three outcomes for the game:

Either: the user wins: Please print out the congratulations to the user

The system wins: Please print out “ You tried your best. Thank you for playing”.

Neither wins: Please print out “Tie game. Thank you for playing”.

1. **Submission:**

All \*.java or \*.cpp files, all header files or jar files if needed. Codes are expected to be fully commented.

A README file that describes how the system is run.

All of these files need to be submitted to Project 2 dropbox by midnight of the due date.

Those of you who are C++ and use Washington.uww.edu server, please just inform me the path. Thanks.