

204212 ADT and Problem Solving Project 2 Game Tree

Due: 6 November 2017 (before midnight)

We are serious about cheating. If the similarity score for your work is above 80%, you receive zero credit for the work. If you cheat for the second time, you fail the course.

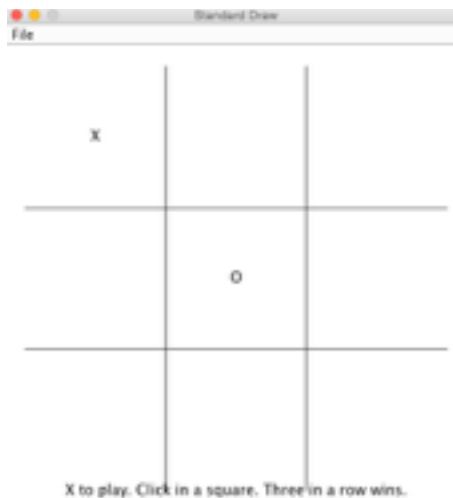
In this project, you will write a program that plays Tic-Tac-Toe intelligently. Learn how to play Tic-Tac-Toe from the following link:

<https://www.factmonster.com/games/tictactoe>

Select level 3 where the computer never loses.

The starting code for drawing a 3x3 grid Tic-Tac-Toe as well as interacting with it using the mouse has been provided for you. The code again uses the StdDraw library from Princeton. You can see how the code work by issuing these commands on the command prompt:

```
javac TicTacToe.java StdDraw.java
java TicTacToe
```



You are to write a program for a user to play Tic-Tac-Toe against the computer. Instead of playing on a 3x3 grid, a user plays against the computer on **a 5x4 grid (5 rows and 4 columns)**. The rules for winning will be the same as in a 3x3 grid except a winner needs to succeed in placing **four** of their marks in a horizontal, vertical, or diagonal row wins the game.

Your program must allow for a user to pick whether he/she wants to start first or the computer is to start first. The computer's every move must be the best move for a given grid configuration. Hence, **your program must use the chooseMove method** discussed in class in the play engine of the computer. You do not have to use Alpha-Beta pruning for this project. The simple pruning strategy will suffice.

Once you have successfully programmed this two-player game, modify the code so that it makes the computer plays against itself where it takes turn choosing the best move. Run this

modified program 10 times and captures all the outcomes on screen. Include these screen captures in the README file. Analyze the result and see if you can come up with a winning strategy if a user moves first or second.

You are to submit the following to Google's Classroom on or before the due date:

- A java source file containing the solution to this project.
- Screen captures of the outcome when:
 - o you move first
 - o the computer moves first
- A README file describing 1.) list of each file, its description, and instructions on how to compile/run the program 2.) any known bugs or limitations and 3.) whether there is always a winning strategy for a player who moves first. You have to show the outcome when the computer is playing against itself from 10 times here.