**COP 3330 Intro To OOP**

**Assignment 4 - Tic-Tac-Toe**

* Create a class TicTacToe that will enable you to write a program to play Tic-Tac-Toe. The class contains a private 3-by-3 two-dimensional array.
* Use an enumeration to represent the value in each cell ofthe array.
* The enumeration’s constants should be named X, O and EMPTY (for a position that does not contain an X or an O).
* The constructor should initialize the board elements to EMPTY.
* Allow for two human players.
* Wherever the first player moves, place an X in the specified square, and place an O wherever the second player moves.
* Each move must be to an empty square.
* After each move, determine whether the game has been won and whether it’s a draw.
* If you feel ambitious, modify your program so that the computer makes the moves for one of the players.

Output (next page)…

|  |  |
| --- | --- |
| Macintosh HD:Users:aeisler:Desktop:Screen shot 2012-11-05 at 2.21.32 PM.png | Macintosh HD:Users:aeisler:Desktop:Screen shot 2012-11-05 at 2.22.56 PM.png |

**Analysis:**

(Describe the problem including input and output in your own words.)

The input for my programs were Actionlisteners added on to JButtons.

The Actionlisteners listened for the user to click once that button is clicked that is the position were you will place your “X” or “O” piece. The output was the Jbutton text to display the piece of that players turn.

**Design:**

(Describe the major steps for solving the problem. Create a UML diagram to accompany your major steps ).

The major steps for solving this problem was to make the methods to check the winner. This was not an easy task as there are multiple ways to win. In this case I had to make several methods to see if you can win diagonally left or right and horizontally or verticall.

|  |
| --- |
| Menu |
|  |
| <<constructor>>Menu() |

|  |
| --- |
| ButtonManger |
|  |
| +Actionperformed(e:ActionEvent)  +gameOver(piece:String, nmoves:int):Boolean  +sbdyWon(piece:String) :Boolean  +horiz(piece:String) :Boolean  +vert(piece:String) :Boolean  +diag(piece:String) :Boolean  +leftdiag(piece:String) :Boolean  +right(piece:String) :Boolean |

|  |
| --- |
| Board |
| + counter:int  +panel1:JPanel  +Button:JButton[][] |
| <<constructor>>Board(size:int) |

**Testing**: (Describe how you tested, or will test this program)

I played against my self and tested all possible outcomes for wins and draws.

**Submit the following items:**

1. Save this Word file; submit it via Canvas Assignments on or before the due date and time.

2. Compile, Run, and Submit your .java file(s) to Canvas. You must submit the program regardless whether it is complete or incomplete, correct or incorrect.