Ryan Madigan OOP Project Documentation

# Requirement Specification

For my OOP project I chose to make simple Tic Tac Toe game. Here is a brief overview of the rules of the game:

* The board is made up of a 3X3 grid.
* There are two players, one is X and the other is O.
* Players take turns placing their character on the board.
* The player can only place a character on an empty tile.
* To win one player must get three of the character in a row through horizontally, vertically or diagonally.
* If the board is filled and neither player has a winning combination then the game is a draw.

Now that the rules are covered let’s move onto the running of the program itself. The program begins by running the Game class. This class doesn’t do much it just starts the game, begins playing the music and calling the openStats() method. This opens the current data is saved. The menu has three buttons: Play, Stats and Quit. Clicking play will begin the game, clicking stats will show the current values for the number of games played, wins, losses and draws. Clicking quit simply closes the application.

Once the player clicks play the GameBoardGUI constructor creates the game board which is made up of nine JButtons each which is automatically blank at the start. At the bottom of the board there is a JLabel which text changes depending on which players turn it is. When a player clicks on a tile an Boolean array which keeps track of which tiles have been selected or not. If the tile is empty the players character will be put onto the button , if the tile already is chosen a simple message will display telling the player that the tile has been taken and cannot be selected. The actionPerformed method keeps track of player turns with a turns variable. Using the modulus operator if the remainder is 0 it’s player one’s turn otherwise it’s player two’s turn. When a tile is successfully chosen the turns variable is incremented.

While the game is running , every time a tile is selected the checkForWinners() method is called seeing if any of the possible winning combinations have been met. If any of the winning conditions are met a message displays announcing the match result. The game stats are then updated based on the match outcome. The saveStats() method is then called to save the new data. The player is then brought back to the main menu.

# UML Diagram – Game class

# 

# UML Diagram – MainMenuGUI class



# UML Diagram – GameBoardGUI class



# UML Diagram – Player class



# VOPC Diagram

