**Programming Project Report**

Name: Riley Nadwodny

Date: 12/2/19

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

For this programming assignment, the objective was to finish a 4x4 Tic-Tac-Toe game that included a programmer-made header and class file, provided by the instructor. The goal of the assignment was to provide experience with using pre-defined classes. The pre-defined ‘Tic4’ class provides a four by four tic-tac-toe board using a 2D array, methods to clear the board, let a player make a move, display the board, and check to see if either of the players have won the game. The program inputs for this program are the user inputs, either X or O depending on whoever’s turn it is and the program outputs are the tic-tac-toe board updated after every player’s move that shows the filled spaces on the board. For this program, there was not much error handling required except for checking that the player inputs a valid character, either an ‘X’ or an ‘O’.

**Design:**

The design for this project was relatively short and simple for the main program. There are three parts overall to the design of this project. There is a header file for the tic4 class named ‘tic4.h’, the implementation of the ‘tic4’ class in the main .cpp file, and then the ‘int main()’ section of the main .cpp file. In tic4.h, there are six public methods available for use and two private variables used by the tic4 class. The methods themselves are defined in the ‘Project6.cpp’ file. All six of the methods are defined here, and easily accessible to be changed and updated as needed in the future. In the future, they may be moved elsewhere for project #7.

In the main body of the program, most of the code is contained in a ‘do while’ loop. The program begins by telling the user X’s should go first, but ultimately letting the user decide. Then, it prompts for user input as row, column, input. The program will not let the user place anything outside of the array, nor will it allow the user to input any character other than an ‘X’ or an ‘O’. The program then checks if the CheckWin method is false and if it is, the while loop continues, prompting the user for their move and then outputting the updated board to the players. Once CheckWin is true with either X’s or O’s, the do while loop ends, and the winner is declared by the program.

**Testing**

In testing the program, the primary focus was that all the methods worked as intended, the players take alternating turns until one player won or there were no more possible moves on the board and displayed the board after every move. The testing cases were pretty simple; test with a case of X winning, a case of O winning, the board being filled up, and checking user input upon receiving their move to see if the move is valid.

Player O, it is your turn.

3 0 O

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | X | | |

+---+---+---+---+

2 | O | | X | |

+---+---+---+---+

3 | O | | | |

+---+---+---+---+

Player X, it is your turn.

3 3 X

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | X | | |

+---+---+---+---+

2 | O | | X | |

+---+---+---+---+

3 | O | | | X |

+---+---+---+---+

Player O, it is your turn.

Hold on... X wins!

0 1 2 3

+---+---+---+---+

0 | | | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

X's should go first. Input your move as [Row] [Column] [Character].

0 0 X

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

1 0 O

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

1 1 X

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | X | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

2 0 O

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | X | | |

+---+---+---+---+

2 | O | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

2 2 X

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | O | X | | |

+---+---+---+---+

2 | O | | X | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

0 1 2 3

+---+---+---+---+

0 | | | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

X's should go first. Input your move as [Row] [Column] [Character].

0 1 X

0 1 2 3

+---+---+---+---+

0 | | X | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

0 0 O

0 1 2 3

+---+---+---+---+

0 | O | X | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

0 2 X

0 1 2 3

+---+---+---+---+

0 | O | X | X | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

1 1 O

0 1 2 3

+---+---+---+---+

0 | O | X | X | |

+---+---+---+---+

1 | | O | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

0 3 X

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | | O | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+ +---+---+---+---+

Player O, it is your turn.

1 0 O

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | O | O | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

2 2 X

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | O | O | | |

+---+---+---+---+

2 | | | X | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

2 0 O

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | O | O | | |

+---+---+---+---+

2 | O | | X | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

2 1 X

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | O | O | | |

+---+---+---+---+

2 | O | X | X | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

3 0 O

0 1 2 3

+---+---+---+---+

0 | O | X | X | X |

+---+---+---+---+

1 | O | O | | |

+---+---+---+---+

2 | O | X | X | |

+---+---+---+---+

3 | O | | | |

+---+---+---+---+

Player X, it is your turn.

Hold on... O wins!

0 1 2 3

+---+---+---+---+

0 | | | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

X's should go first. Input your move as [Row] [Column] [Character].

0 0 X

0 1 2 3

+---+---+---+---+

0 | X | | | |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

4 4 O

0 4 O

4 0 O

0 3 O

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | | | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

1 1 X

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | | X | | |

+---+---+---+---+

2 | | | | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player O, it is your turn.

2 2 O

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | | X | | |

+---+---+---+---+

2 | | | O | |

+---+---+---+---+

3 | | | | |

+---+---+---+---+

Player X, it is your turn.

2 2 X

2 2 X

4 4 X

3 0 X

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | | X | | |

+---+---+---+---+

2 | | | O | |

+---+---+---+---+

3 | X | | | |

+---+---+---+---+

Player O, it is your turn.

2 3 O

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | | X | | |

+---+---+---+---+

2 | | | O | O |

+---+---+---+---+

3 | X | | | |

+---+---+---+---+

Player X, it is your turn.

1 0 X

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | X | X | | |

+---+---+---+---+

2 | | | O | O |

+---+---+---+---+

3 | X | | | |

+---+---+---+---+

Player O, it is your turn.

2 1 O

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | X | X | | |

+---+---+---+---+

2 | | O | O | O |

+---+---+---+---+

3 | X | | | |

+---+---+---+---+

Player X, it is your turn.

2 0 X

0 1 2 3

+---+---+---+---+

0 | X | | | O |

+---+---+---+---+

1 | X | X | | |

+---+---+---+---+

2 | X | O | O | O |

+---+---+---+---+

3 | X | | | |

+---+---+---+---+

Player O, it is your turn.

Hold on... X wins!

**Conclusion**

Overall, this programming assignment was not too challenging, and provided good experience with using public class methods and going more in depth about how classes work. One thing next time that could be done differently is to separate the definition of the class methods into a separate file rather than having them in the main program and allowing them to be accessible, but sense outside users tempering with the code is not a worry, it was not given too much thought this time around. Total time to completion took roughly 5-6 hours, and the overall project is certainly a success.