```java
//Kevin Yu
//11-3-2021
//Imports all packages
package TicTacToGUI;

import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class TicTacToGUI implements ActionListener{

      //Creates all of the variables
      JFrame frame = new JFrame();
      JButton[][] button = new JButton[3][3];
      int [][]board = new int[3][3];
      final int BLANK = 0;
      final int X_MOVE = 1;
      final int O_MOVE = 2;
      final int X_TURN = 0;
      final int O_TURN = 1;
      int turn = X_TURN;
      Container center = new Container();
      JLabel xLabel = new JLabel("X wins:0");
      JLabel oLabel = new JLabel("O wins:0");
      JButton xChangeName = new JButton("Change X's Name.");
      JButton oChangeName = new JButton("Change O's Name.");
      JTextField xChangeField = new JTextField();
      JTextField oChangeField = new JTextField();
      Container north = new Container();
      String xPlayerName = "X";
      String oPlayerName = "O";
      int xwins = 0;
      int owins = 0;

      public TicTacToGUI() {
            frame.setSize(400, 400);
            //Center grid container
            frame.setLayout(new BorderLayout());
            center.setLayout(new GridLayout(3,3));
            for (int i = 0; i < button[0].length; i++) {
                  for (int j = 0; j < button[0].length; j++) {
                        button[j][i] = new JButton();
                        center.add(button[j][i]);
                        button[j][i].addActionListener(this);
                  }
            }
            //Creates the Names
            frame.add(center, BorderLayout.CENTER);
            //North container
            north.setLayout(new GridLayout(3,2));
            north.add(xLabel);
            north.add(oLabel);
```

```
        north.add(xChangeName);
        xChangeName.addActionListener(this);
        north.add(oChangeName);
        oChangeName.addActionListener(this);
        north.add(xChangeField);
        north.add(oChangeField);
        frame.add(north, BorderLayout.NORTH);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        new TicTacToGUI();
    }

    @Override
    //Making the X's and O's when you click on the board and setting the turns
and checks if you wins
    public void actionPerformed(ActionEvent event) {
        JButton current;
        boolean gridButton = false;
        for (int i = 0; i < button.length; i++) {
            for (int j = 0; j < button[0].length; j++) {
                if (event.getSource().equals(button[j][i])) {
                    gridButton = true;
                    current = button[j][i];
                    if (board[j][i] == BLANK) {
                        if (turn == X_TURN) {
                            current.setText("X");
                            board[j][i] = X_MOVE;
                            turn = O_TURN;
                        }
                        else {
                            current.setText("O");
                            board[j][i] = O_MOVE;
                            turn = X_TURN;
                        }
                        //check for wins and ties
                        if (checkWin(X_MOVE) == true) {
                            //X Wins!
                            xwins++;
                            xLabel.setText(xPlayerName + " wins:" +
xwins);

                            clearBoard();
                        }
                        else if (checkWin(O_MOVE) == true) {
                            //O Wins!
                            owins++;
                            oLabel.setText(oPlayerName + " wins:" +
owins);

                            clearBoard();
                        }
                        else if (checkTie() == true) {
                            //Tie!
                            clearBoard();
                        }
                    }
                }
```

```
                  }
            }
            //This is where you rename yourself
            if (gridButton == false) {
                  if (event.getSource().equals(xChangeName) == true) {
                        if (xChangeField.getText().equals("") == false) {
                              xPlayerName = xChangeField.getText();
                              xLabel.setText(xPlayerName + " wins:" + xwins);
                        }
                  }
                  else if (event.getSource().equals(oChangeName) == true) {
                        if (oChangeField.getText().equals("") == false) {
                              oPlayerName = oChangeField.getText();
                              oLabel.setText(oPlayerName + " wins:" + owins);
                        }
                  }
            }
      }
      //All the conditions for wins
      public boolean checkWin(int player) {
            if (board[0][0] == player &&
                        board[0][1] == player &&
                        board[0][2] == player) {
                  return true;
            }
            if (board[1][0] == player &&
                        board[1][1] == player &&
                        board[1][2] == player) {
                  return true;
            }
            if (board[2][0] == player &&
                        board[2][1] == player &&
                        board[2][2] == player) {
                  return true;
            }
            if (board[0][0] == player &&
                        board[1][0] == player &&
                        board[2][0] == player) {
                  return true;
            }
            if (board[0][1] == player &&
                        board[1][1] == player &&
                        board[2][1] == player) {
                  return true;
            }
            if (board[0][2] == player &&
                        board[1][2] == player &&
                        board[2][2] == player) {
                  return true;
            }
            if (board[0][0] == player &&
                        board[1][1] == player &&
                        board[2][2] == player) {
                  return true;
            }
            if (board[0][2] == player &&
                        board[1][1] == player &&
                        board[2][0] == player) {
                  return true;
            }

            return false;
      }
      //Checks for tie
      public boolean checkTie() {
            for (int row = 0; row < board.length; row++) {
                  for (int column = 0; column < board.length; column++) {
                        if (board[row][column] == BLANK) {
                              return false;
                        }
                  }
            }
            return true;
      }
      //Clears the board
      public void clearBoard() {
            for (int a = 0; a < board.length; a++) {
                  for (int b = 0; b < board[0].length; b++) {
                        board[a][b] = BLANK;
                        button[a][b].setText("");
                  }
            }
            //always starts on x
            turn = X_TURN;
      }
}
```