

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package mvctictactoe;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.MouseListener;
import java.util.Observable;
import java.util.Observer;
import javax.swing.JLabel;

/**
 *
 * @author Benjamin Chinwe 2016
 */
public class TicTacToeView extends JFrame implements Observer {
```

```
Container contentPane;

Font font;

private final JTextField outputField;
private JButton tempGameButton;
private final JButton[][] gameButton = new JButton[3][3];
private final JPanel buttonPanel;
JPanel outputPanel = new JPanel();
private String playerOneName;
private String playerTwoName;
private String player;
private TicTacToeSingleton colorSingleton;
private final JLabel gameStatusLabel;
private final JPanel gameStatusPanel;

@SuppressWarnings("OverrideMethodCallInConstructor")
public TicTacToeView() { // TicTacToeView class constructor
    // Set up UI
    addWindowListener(new WindowDestroyer());
    setTitle("TicTacToe");
    setSize(700, 700);
    setLocation(150, 50);
    contentPane = getContentPane();
    contentPane.setLayout(new BorderLayout());

    // Panel to contain the game buttons
    buttonPanel = new JPanel();
    buttonPanel.setBackground(Color.gray);
    buttonPanel.setLayout(new GridLayout(3, 3));
```

```
// font = new Font("Courier", Font.BOLD, 18); // Setting Textfield
//font propeties
outputField = new JTextField(" Welcome to TreIRad i Grafik ");
font = outputField.getFont().deriveFont(Font.BOLD, 30f);
outputField.setFont(font);
outputField.setForeground(Color.BLACK);

outputPanel.setBackground(Color.BLACK);
contentPane.add(outputField, BorderLayout.NORTH);

layerGameBoard(gameButton); // Method to initialize game board
contentPane.add(buttonPanel, BorderLayout.CENTER);

gameStatusLabel = new JLabel();
gameStatusLabel.setFont(font);
gameStatusPanel = new JPanel();
gameStatusPanel.add(gameStatusLabel);
contentPane.add(gameStatusPanel, BorderLayout.SOUTH);

}

// Method to initialize game board
private void layerGameBoard(JButton[][] gameButton) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            tempGameButton = new JButton();
            gameButton[i][j] = tempGameButton;
            gameButton[i][j].setBackground(new Color(
                TicTacToeSingleton.getInstance(),
```

```

        TicTacToeSingleton.getInstance(),
        TicTacToeSingleton.getInstance());
    buttonPanel.add(gameButton[i][j]);
    }
    }

    // Method to reset the gameboard after a win or draw
    public void setGameButton(JButton[][] gameButtonView) {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                gameButtonView[i][j].setBackground(new Color((int) (Math.random()
                    * 255 + 1), (int) (Math.random() * 255 + 1), (int) (Math.random() * 255 + 1)));
                this.gameButton[i][j] = gameButtonView[i][j];
            }
        }
    }

    // Method to accept players names
    void setPlayersName() {
        playerOneName = JOptionPane.showInputDialog(null,
            "Enter Player One Name").toUpperCase();
        playerTwoName = JOptionPane.showInputDialog(null,
            "Enter Player Two Name").toUpperCase();
    }

    //Method to accept player's ID - letter
    void setPlayerSeed() {
        player = JOptionPane.showInputDialog(null,

```

```
        "Enter ID letter - X or O : ").toUpperCase();

        while (((!(player.equals("X"))) && (!(player.equals("O"))))) {
            player = JOptionPane.showInputDialog(null,
                "Try Again! Enter letter - X or O : ").toUpperCase();
        }
        //outputField.setText(playerOneName + "! Please start the game! ");
    }
    // Method to add ActionListener to the game buttons

    void addGameButtonListener(ActionListener listenForButtonClick) {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                gameButton[i][j].addActionListener(listenForButtonClick);
                gameButton[i][j].addMouseListener((MouseListener) listenForButtonClick);
            }
        }
    }

    public void setGameStatusLabel(String messageLabel) {
        gameStatusLabel.setText(messageLabel);
    }

    public JLabel getGameStatusLabel() {
        return gameStatusLabel;
    }

    // Method to get or return player ID - letter
```

```
public String getPlayerSeed() {
    return player;
}

// Method to reSet player's ID - letter
void setPlayerSeed(String letter) {
    player = letter;
}

// Method to get or return player One Name
public String getPlayerOneName() {
    return playerOneName;
}

// Method to get or return player Two Name
public String getPlayerTwoName() {
    return playerTwoName;
}

// methd to display error message
void displayErrorMessage(String message) {
    //JOptionPane.showMessageDialog(null, message);
    gameStatusLabel.setText(message);
}

//Method to return textfield information
public JTextField getOutputPutText() {
    return outputField;
}
```

```
// Method to return the gameButton array
public JButton[][] getGameButton() {
    return gameButton;
}

@Override
public void update(Observable o, Object arg) { //Observer update method
    displayErrorMessage((String) arg);
}
}
```