

```
/*
 * 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 javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JOptionPane;

/**
 *
 */
 * @author Benjamin Chinwe 2016
 */
public class TicTacToeModel {

    public enum Seed { //Enumerator with three values
        EMPTY, CROSS, NOUGHT
    }

    // Class variables
    private String playerSeed;
    private Seed[][] arr;
    private int xnum;
    private int onum;

    public Seed currentSeed;
```

```
private boolean go = false;
private JButton[][] gameButton;
private String playerTwoName;
private String playerOneName;

public TicTacToeModel() { //Argumentless constructor method
}

// Argumented constructor method
public TicTacToeModel(JButton[][] gameButtonCont,
String playerOneNameCont, String playerTwoNameCont,
String playerSeedCont) {
    arr = new Seed[3][3]; // Creates enum Seed array
    gameButton = gameButtonCont;
    //this.go = go;
    playerSeed = playerSeedCont;
    playerOneName = playerOneNameCont;
    playerTwoName = playerTwoNameCont;

    for (int i = 0; i < 3; i++) { // For- to initialise Seed array arr
        for (int j = 0; j < 3; j++) {
            arr[i][j] = Seed.EMPTY;
        }
    }

    //Method to check if game board is full
    public boolean boardFull() {
        boolean full = true;
```

```
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (arr[i][j] == Seed.EMPTY) {
                    full = false;
                }
            }
        }
        return full;
    }

    // Method to set current or clicked button Seed
    public void setCurrentSeed(Seed currentSeedNew, int r, int c) {
        currentSeed = currentSeedNew;
        arr[r][c] = currentSeedNew;
        //System.out.println("r = " + r + " c = " + c);
        //System.out.println(" TicTacToeModel.Seed - " + currentSeed);
    }

    // Method to reset the game after a win
    public void reset() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                //gameButton[i][j].setText("");
                gameButton[i][j].setIcon(null);
                arr[i][j] = Seed.EMPTY;
                gameButton[i][j].setEnabled(true);
            }
        }
    }
}
```

```

playerSeed = JOptionPane.showInputDialog(null,
    "Enter letter - X or O : ").toUpperCase();
while ((!(playerSeed.equals("X")) && !(playerSeed.equals("O"))))) {
    playerSeed = JOptionPane.showInputDialog(null,
        "Try Again! Enter letter - X or O : ").toUpperCase();
    }
    setGo(false);
}

// Method to determine if there is a winner
public void whoWins() {
    boolean xwin = false;
    boolean owin = false;
    try {
        //Check rows for wins
        for (int row = 0; row < 3; row++) {
            for (int col = 0; col < 3; col++) {
                if (arr[row][0] == Seed.CROSS
                    && arr[row][1] == Seed.CROSS
                    && arr[row][2] == Seed.CROSS) {
                    xwin = true;
                }
                // Check if row win
            } else if (arr[0][col] == Seed.NOUGHT
                && arr[1][col] == Seed.NOUGHT
                && arr[2][col] == Seed.NOUGHT) {
                    owin = true;
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
}

    if (arr[row][0] == Seed.NOUGHT
        && arr[row][1] == Seed.NOUGHT
        && arr[row][2] == Seed.NOUGHT) {
        owin = true;
    }

    // Check if row win
    } else if (arr[0][col] == Seed.CROSS
        && arr[1][col] == Seed.CROSS
        && arr[2][col] == Seed.CROSS) {
        xwin = true;
    }

    // Check if down diagonal win
    if (row == col) {
        if (arr[0][0] == Seed.CROSS
            && arr[1][1] == Seed.CROSS
            && arr[2][2] == Seed.CROSS) {
            xwin = true;
        }
    } else if (arr[0][0] == Seed.NOUGHT
        && arr[1][1] == Seed.NOUGHT
        && arr[2][2] == Seed.NOUGHT) {
        owin = true;
    }
}
```

```
//Check if up diagonal win
if (row + col == 2) {
    if (arr[0][2] == Seed.CROSS
        && arr[1][1] == Seed.CROSS
        && arr[2][0] == Seed.CROSS) {
        xwin = true;
    } else if (arr[0][2] == Seed.NOUGHT
        && arr[1][1] == Seed.NOUGHT
        && arr[2][0] == Seed.NOUGHT) {
        owin = true;
    }
}

if (xwin) {
    //Keep score, report win for player one
    JOptionPane.showMessageDialog(null, getPlayerOneName()
        + " Wins!!! Score X:" + (xnum + 1) + " O:" + onum,
        " WINNER : Click a box to continue!",
        JOptionPane.INFORMATION_MESSAGE, new ImageIcon(
            ImageIO.read(TicTacToeModel.class.
                getResourceAsStream("image/cross.jpg"))));

    xnum++;
    setGo(true);
    reset(); // Reset game
}
```

```
if (owin) {
    //Keep score, report win for player one
    JOptionPane.showMessageDialog(null, getPlayerTwoName()
        + " Wins!!! Score X:" + xnum + " O:" + (onum + 1),
        " WINNER : Click a box to continue! ",
        JOptionPane.INFORMATION_MESSAGE, new ImageIcon(
            ImageIO.read(TicTacToeModel.class.
                getResourceAsStream("image/cross.jpg"))));

    onum++;
    setGo(true);
    reset(); // Reset game
}
} catch (Exception e) { // Capture error
    System.err.println(e.getMessage());
}
}

// Method to set booleana value go
public void setGo(boolean go) {
    this.go = go;
}

// Method to return sun of X seed after a draw
int getXTotal() {
    return xnum;
}

// Method to return sun of O seed after a draw
int getOTotal() {
```

```
        return onum;
    }

    // Method to return new seed after a win or draw
    public String getPlayerSeed() {
        return playerSeed;
    }

    // Method return current go value
    public boolean isGo() {
        return go;
    }

    //Method return game buuton array after a draw
    public JButton[][] getGameButton() {
        return gameButton;
    }

    //Methid gets second player name
    public String getPlayerTwoName() {
        return playerTwoName;
    }

    //Methid gets first player name
    public String getPlayerOneName() {
        return playerOneName;
    }

    //Method to return current enum value
```



```
public Seed getCurrentSeed() {  
    return currentSeed;  
}  
  
void setPlayerSeed(String playerSeedCont) {  
    playerSeed = playerSeedCont;  
}  
}
```