Design Pattern

TicTacToeTest class

This is the main class for the TicTacToe game. It generates objects of Model, View and Controller classes and aggregates them.

Name of Controller class: Controller

The Controller class is responsible for requesting the model to update its state whenever there is an event on a button on the game board.

Segment:

```
JButton[][] board = theView.getBoard();
for (int i = 0; i < board.length; i++) {
        for (int j = 0; j < board[i].length; j++) {
                board[i][j].setText(theModel.getBoard()[i][j] + "");
        }
}
```

Name of Model class: Model

The model class is where the current state of the game, player symbol, as well as the logic for the result of the game, . The model class calls the view to update the GUI according to the current state of the game.

Segment Name: validMove()

```
public boolean validMove(int r, int c) {
    if (board[r][c] == '\0') {  // if the spot on the board is null

        // if new player has player, reset undo count
        if(lastPlayer != playerSymbol) {
           undos = 0;
        }

        board[r][c] = playerSymbol;
```

```
            // update last move stats
            lastR = r;
            lastC = c;
            lastPlayer = playerSymbol;

            // switch player
            switchPlayer();

            return true;
        }
        return false;
    }
```

Name of View class: View

This class is responsible for setting up the GUI and showcasing the nature of the game on the GUI as notified by the model.

Segment:

```
public void addUndoButtonListener(ActionListener listener) {
        btnUndo.addActionListener(listener);
}
```

Strategy Pattern: UILook

```
import javax.swing.JButton;

// To define different strategy for UI
public interface UILook {
    public void formatSquare(JButton button);
}
```

Concrete Strategies:

BasicLook
ModernLook

We created classes to format the boards in two different types of views namely: Basic Look and ModernLook class that acts as a display board of 2 different types using the UILook interface.

Code Segment for BasicLook:

```
String[] choices = {"Basic Look", "Modern Look"};
    int choice = JOptionPane.showOptionDialog(null, "Please choose what type of style you would like", "Style Selection", JOptionPane.OK_OPTION, JOptionPane.QUESTION_MESSAGE, null, choices, choices[0]);
    UILook uiLook;
    if(choice == 0)
       uiLook = new BasicLook();
    else
       uiLook = new ModernLook();

    View theView = new View(uiLook);
```