

Design Pattern Assessment

Controller

GameController

```
private void createActionListener(int row, int col) {
    JButton[][] buttons = view.getButtons();
    buttons[row][col].addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            model.makeMove(row, col);
        }
    });
}
```

Model

GameModel

```
public void makeMove(int row, int col) {
    if (hasWinner) return;
    if (board[row][col] != null) return;

    if (playerTurn == "X") {
        board[row][col] = "X";
        playerTurn = "O";
    } else {
        board[row][col] = "O";
        playerTurn = "X";
    }

    if (undoned == true) undoned = false;

    undoList.push(new int[] {row, col});
    calculate(row, col);
    setChanged();
    notifyObservers();
}
```

View

GameView

```
public void update(Observable o, Object arg) {
    String[][] data = model.getBoardData();

    for (int i = 0; i < data.length; i++) {
        for (int j = 0; j < data.length; j++) {
            if (data[i][j] == "X") {
                buttons[i][j].setForeground(
                    new Color(255, 77, 77));
                buttons[i][j].setText(data[i][j]);
            } else {
                buttons[i][j].setForeground(
                    new Color(23, 192, 235));
                buttons[i][j].setText(data[i][j]);
            }
        }
    }
    if (model.hasWinner()) {
        message.setText(model.getWinner() + " won the game!");
        return;
    }
    message.setText(model.getTurn() + "'s Turn:");
}
```

Model notifies observers to update()

update() method grabs board data from model and changes the buttons based on the 2d array.

Strategy

Style

```
public interface Style {
    public void loadStyle();
    public String getStyleName();
}
```

Concrete Strategies

DarkStyle

LightStyle

```
private void initializeBoard() {
    gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    gui.setSize(500,350);
    gui.setResizable(true);

    Style[] styles = {
        new DarkStyle(gui, buttons, undo, message),
        new LightStyle(gui, buttons, undo, message)};

    for (int i = 0; i < styles.length; i++) {
        createStyleButton(styles[i]);
    }

    gui.setVisible(true);
}

private void createStyleButton(Style style) {
    JButton button = new JButton();
    button.setText(style.getStyleName());
    button.setFocusable(false);
    stylePanel.add(button);
    gui.add(stylePanel, BorderLayout.CENTER);

    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            stylePanel.setVisible(false);
            initializeGame();
            style.loadStyle();
        }
    });
}
```