

### 1. Model:

The `TicTacToeModel` class represents the model in the MVC pattern. It encapsulates the game logic and data. It maintains the state of the game board, keeps track of the current player, and provides methods to make moves, check for a win or draw, and access the game board's state.

### 2. View:

The `TicTacToeView` class represents the view in the MVC pattern. It is responsible for the user interface and visual presentation of the game. It creates a graphical window with a 3x3 grid of buttons to represent the game board. The view provides methods to update the button labels and enable/disable buttons based on the game state. It also displays the game result in a separate pop-up window.

### 3. Controller:

The `TicTacToeController` class acts as the controller in the MVC pattern. It mediates between the model and the view, handling user interactions and updating the model and view accordingly. The controller attaches action listeners to the buttons in the view and responds to button clicks. When a button is clicked, the controller checks if the move is valid, updates the model by calling the appropriate methods, and updates the view accordingly. It also checks for a win or draw condition and displays the result using the view.

The MVC approach ensures a clear separation of concerns and promotes modularity and maintainability. The model represents the game logic and data, the view handles the user interface, and the controller acts as the intermediary, handling user input and updating the model and view. This separation allows for easier modifications and enhancements to specific components without affecting the others.