## ObserverView

### CRC

|  |  |
| --- | --- |
| ObserverView | |
| Responsibilities: To display the Observer interface when a game is created and the user is an observer. | Collaborators: It inherits from BoardDisplay and ActionView. It is activated and deactivated through the ViewController. The game provides the information necessary to perform its task. |

### Description / Overview

The ObserverView class in our model-view-controller architecture is a view. It is activated by the ViewController when a game is created and the user is an Observer. Like the GameView, it is to create a board and display that in the interface since both involve a game being played. The Observer view is different from the GameView since it does not allow the user to interact with the board directly. Instead they are given viewing options; things like changing the robot think time which speeds up or slows down the game. Like the Gameview it has buttons to display rules of the game or to exit the match. Fog of war checkboxes are available as well so the user can choose which teams perspectives they can watch the game from. Similar to the gameView, it is redrawn whenever changes are made to the model or if the fog of war is changed. Once a game is finished the ViewController will deactivate the ObserverView.

### Instance Variables

#### gameMaster

Data Type:GameMaster

The gameMaster variable has the purpose of creating a link between ObserverView and the GameMaster Controller. It is useful for when events are triggered in the view, that it can then send the events to the GameMaster to be handled there.

### Method Overview

*public ObserverView(GameMaster gameMaster)*

### Method Writeups

#### public ObserverVew(GameMaster gameMaster)

This method is the constructor for ObserverView. It is called when an ObserverView object is created by the ViewController. It creates the ObserverView interface and displays it to the user. This includes the hexagon board and other interface elements as explained in the overview. It also contains listeners for events that are triggered from the observer interacting with the interface; the events such as changing robot think time, changing fog of war options, viewing rules, and exiting a match are all sent to the GameMaster controller to be handled. When change to the game state is made, the ObserverView will redraw to represent the most up-to-date model.