FLOW

The flow component regulates the flow of the game based on user and player input. The Flow component is essentially divided into two sub-components, MenuManager and EventCatcher. The MenuManager sub-component is designed for the events initiated by the User, which is outside of the match and the EventCatcher sub-component is designed for all Player responses.

The Flow Class, which implements the AWT Event Listener Interface responds to various events that the User and the Player initiate. These events could be a click on the screen, a click on a button, or a key press on the keyboard. The Swing toolkit is used to handle these possible events, which is located in the Javax package.

The eventDispatched(AWTEvent) function is the hub of all events and it is located in the main Flow class. It calls the delegate functions to perform the specific job based on the particular event. This function accepts the AWTEvent as a parameter.

Each of the MenuManger and EventCatcher functions contain an array of Strings which stores all the possible button names in their scope of screens.

A private function IsValidButton() is used to check whether or not any of the arguments of button names are valid.

The function clickedButton(String ) is located both in MenuManager class and EventCatcher class. The function in the MenuManager deals with the buttons clicked on all screens other than the game screen. The function in the EventCatcher handles the buttons on the game screen. Both the functions take in the name of the button which is clicked as a parameter, throw an exception if the button name is invalid, and return nothing.

The function clickedEndTurn() is called when the Player decides to end their turn. The function neither accepts any parameters nor returns a value.

The function clickedMove(int ) is called when the Player decides to move their piece to a position. This void-returning function accepts an integer variable which refers to the position at which the piece wants to move.

The function clickedShoot(int ) is called when the Player decides to shoot at a piece at a position. The function accepts an integer variable which refers to the position at which the piece wants to shoot and does not return any value.

The void-returning function clickedExitScreen() is called when the Player wants to quit the Match. This function does not accept any parameters.

The function clickedPanArrow() is called when the Player chooses to switch between the Pan mode and the Arrow mode.

The draggedGameBoard(mousePosition pre, mousePosition post ) function is called when the player is panning the game board. This is a void-returning function and it accepts initial and final positions of the mouse.

**ONLY FOR SCOTT** OR WHOVER IS PUTTING IT TOGETHER:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Flow Component**   |  | | --- | | Flow Class implements AWTEventListener  //We could probably use singleton Pattern like Design Pattern here - I think.  //Basically just contains one function which figures out what event it is and it disperses the event to the functions of classes.  ~Flow()  [eventDispatched](https://docs.oracle.com/javase/7/docs/api/java/awt/event/AWTEventListener.html#eventDispatched(java.awt.AWTEvent))([AWTEvent](https://docs.oracle.com/javase/7/docs/api/java/awt/AWTEvent.html) event)  Invoked when an event is dispatched in the AWT. |  |  |  | | --- | --- | | MenuManager Class - User | EventCatcher Class - Player | | //Has just one function  String [] Buttons  clickedButton throws InvalidButton Exception  isValidButton - private | //Has functions like:  String [] Buttons  clickedButton throws InvalidButton Exception  isValidButton - private  draggedGameBoard  clickedMove  clickedShoot  clickedExit  clickedPanArrow | |