ViewerInterface

The following are methods which all Viewer derivatives must have implemented and use. Examples of how they can be implemented are given in the Viewer.h base class.

* void loop() : This method provides opportunities for any behavior that the Viewer needs to perform with every loop cycle. For example, checking for Notices form the Model, or refreshing the graphical display. This method is called by the client in the main game loop of client code.
* void update() : This method is used for processing all Notices the Viewer may have queued up since the last time the update() method was called.
* bool pollInput() : returns whether or not the Viewer has received any Input objects
* Input\* getInput() : returns a pointer to the Input at the front of the queue.
* void processNotice(Notice\* n) : takes a pointer to a Notice instance and performs any implemented operations according to the type of Notice passed. Notice derivatives must be implemented according to the project requirements which this MVC is being used in.
* void takeNotice(Notice\* n) : stores the passed Notice pointer in the queue of Notices to be processed in the next update() call.
* void updateDisplay(Notice\* n) : modifies the Viewer’s graphical display according to the Notice passed. This advanced behavior must be implemented according to the type of graphical library which the client chooses to use for the Viewer.