ModelInterface

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

* void update() : processes all Event objects in the queue. Events are processed using updateModel(Event\* e). How Event derivatives are processed can be determined by overwriting the updateModel() method in a Model derivative class. Notice objects can then be created according to the type of Event processed. Notices are then passed to the connected Viewer object.
* Void notifyViewer(Notice\* n) : used to pass Notices to the connected Viewer class
* Void updateModel(Event\* e) : operates on the Model data based on the kind of Event passed. How the Model operates on itself can be defined by implementing updateModel(Event\* e) in a derived Model class.
* Void takeEvent(Event\* e) : used to receive Events from the connected Controller object