Design Document

Team 2

CS 340

Observer Pattern:

The Game class will extend Java’s built in Observable class. When the client and model are instantiated the controller classes will add themselves to the Game class’ list of observers. Whenever a change occurs in the model the Game class will notify all of the listeners attached to it. The observers will then ensure the notification came from the Game class and will query for the updated state.

public class Game extends Observable implements IGame, JsonSerializable {  
 …  
}

Each of the controller classes (presenters) will implement Java’s built in Observer interface. Allowing these classes to update their internal state as values/states in the model change. When the Game class notifies the controllers of a change each controller will verify the origin of the change and determine if it needs to take an action. If an action is required, the controller will update its respective state/view.

*/\*\*  
 \* Implementation for the maritime trade controller  
 \*/*public class MaritimeTradeController extends Controller implements IMaritimeTradeController, Observer {  
 *…*

@Override  
 public void update(Observable o, Object arg) {  
   
 }  
}

State Pattern:

Controllers requiring different functionality based on the current game state will utilize the State Pattern to facilitate simplicity in code structure. The Map Controller specifically will utilize this pattern due to its high number of state dependent functions. The Map Controller will have a pointer to a new class representing the Map Controller’s state. This new class will be a base class for classes representing the various states in the game. The following states will be utilized by the map controller:

* Playing – Map Controller will display the map and wait for some sort of interaction
  + Phases/States such as Discarding, Rolling, etc. that don’t directly affect the map will fall under this category
* Placing Robber – Displays an overlay allowing the player to place/move the Robber
* Building
  + Road – Displays an overlay allowing the player to place the corresponding piece
  + Settlement – Displays an overlay allowing the player to place the corresponding piece
  + City – Displays an overlay allowing the player to place the corresponding piece
* Robbing – Displays a separate view for robbing players
* Playing Development Card
  + Soldier – This will start the Placing Robber and Robbing states
  + Road Building – Displays the overlay for building a road twice (player gets to place 2 roads)

\*Setup phases 1 and 2 will utilize the Building Road and Building Settlement states

\*See JavaDoc for additional details