Team 11 Design Pattern Write-Up

We used several design patterns in our implementation of project three including Builder, Composite, Decorator, Iterator, Mediator, and State. We used Builder in our implementation to draw the UI of our project. We made several small draw methods and then called all of the draw methods inside one big DrawTemplate() method. We also used the builder design pattern in our card implementation. We made a general card implementation and then we built upon that implementation in our cardCatalog for each different card we included in the game. We also used builder for a lot of our game logic such as attacking the opponent. We used the Decorator design pattern to pass a lot of our methods from our smaller classes up to our main and UI classes. We used iterators several times throughout the project. Some examples of the iterators we used include the drawing of the boards and checking for the locations of clicks and other user input. The mediator for our project was our main.js file which is where most of our class interaction happens. This was helpful because it reduced dependencies from one class to another because they had to interact with each other through our main class. The most important design pattern we used was the State design pattern. Because we designed a game, the State design pattern was integral in allowing the game to progress and update. The UI has to update constantly based on game events to ensure that the game progresses as intended otherwise there would be no point to playing the game.