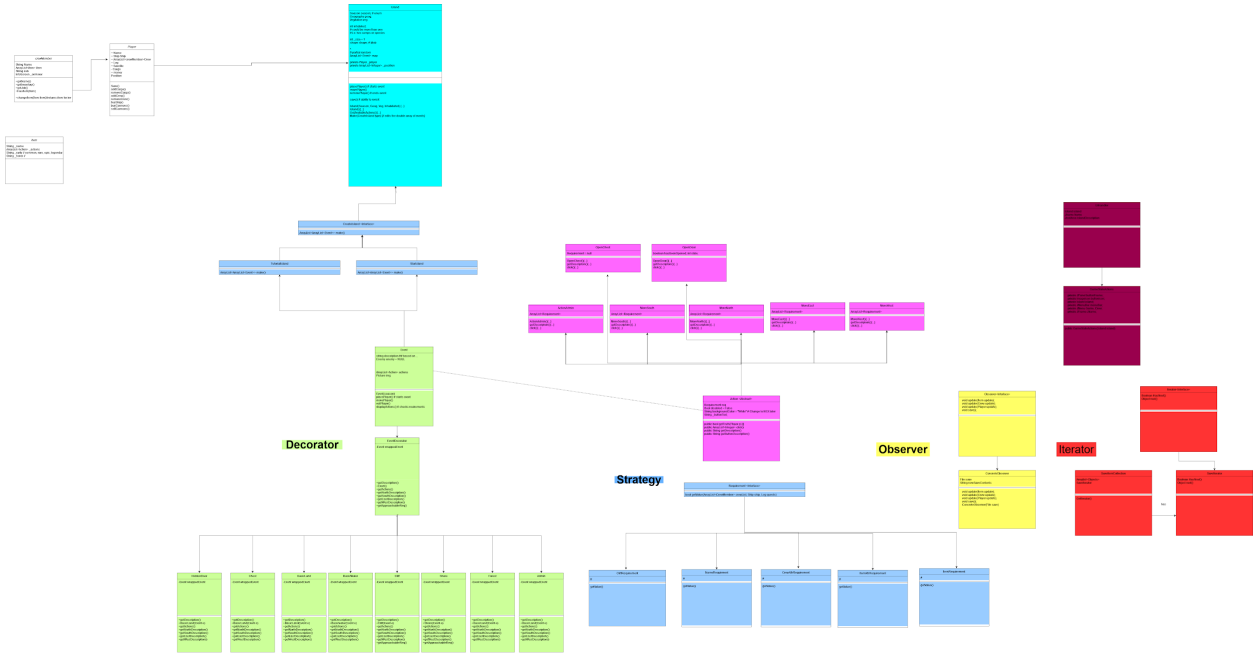
Lab 7 Writeup

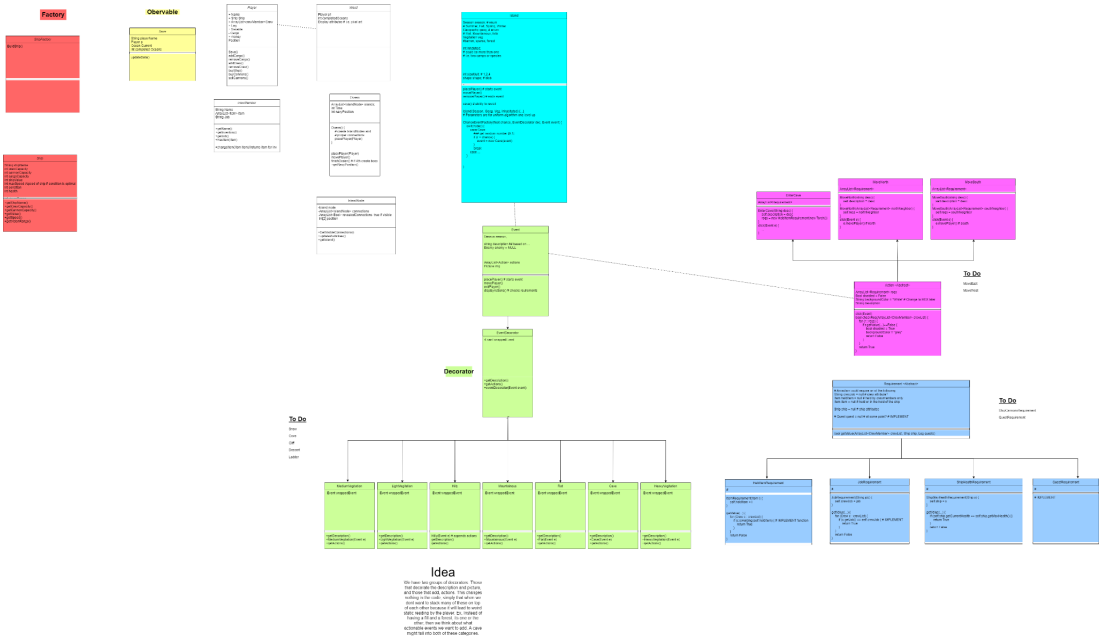
1. Pirate Game: Thomas Mahre , Nathan Reed
2. Final State of System:

   Our game started out strong with the team implementing the first two design patterns and UI on schedule. Proceeding this we started to get bunked down with implementing edge cases due to the connections between the UI and the architecture as we had failed to include the UI into our UML. Specifically: Purely Decorative Events, Purely Action Events (No description), Events that happen once, Events that change over time, Visible Events, Conditional Events, etc. For our last design pattern we decided to pivot to loading from our save files using the Iterator pattern rather than our original ship builder idea. This was successful as our save files were object types such as Player, Crew, and Items saved as Strings. The iterator was able to parse these files and load back into our game with the correct information and location transformed back into their appropriate objects.

   Project 7

3. 

   Project 5:



4. Small portions of our project were inspired by stackoverflow. Single lines can be found that are exactly the same with variable name variations. These were always cited and the links can be found in the code whenever this approach was used to handle a problem we encountered.
5. Our largest issue was the idea of the game itself. As we wanted it to be multiple things and had a variety of ideas, without a single overarching plot line it was difficult to stay within the bounds we set for ourselves. UI was a large part of this as we wanted our game to feel sleek and usable. The second challenging issue was not being aware of all our use cases for our patterns, specifically the decorator pattern. While it was a great idea, if the Decorator had been planned with use cases as stated in paragraph 1 in mind (Purely Decorative Events, Purely Action Events (No description), Events that happen once, Events that change over time, Visible Events, Conditional Events, etc.), we would have run into far fewer problems. We also ran into many issues implementing a UI, specifically it was difficult to update the UI, layout managers were very difficult to handle.