**OOAD Project 6 Update**

Team: Anna Rahn, Jason Mu, Sam Carruthers
Project Title: Dungeon Crawler

**Status Summary:**

Completed Work -

- Sam - Room class for containing room information (safe room vs. monster room, etc.), dictionary to track rooms visited and room states, and mini-map to track player movement and rooms visually
- Anna - Object pool pattern for instantiating all items on startup, chest animation, starter player class and inventory, when chest is clicked a random item from pool is added to inventory, battle mockup
- Jason - Monster class that contains all the information about the monster including their health, damage, defense, and graphics using factory and strategy patterns

Changes / Issues Encountered -

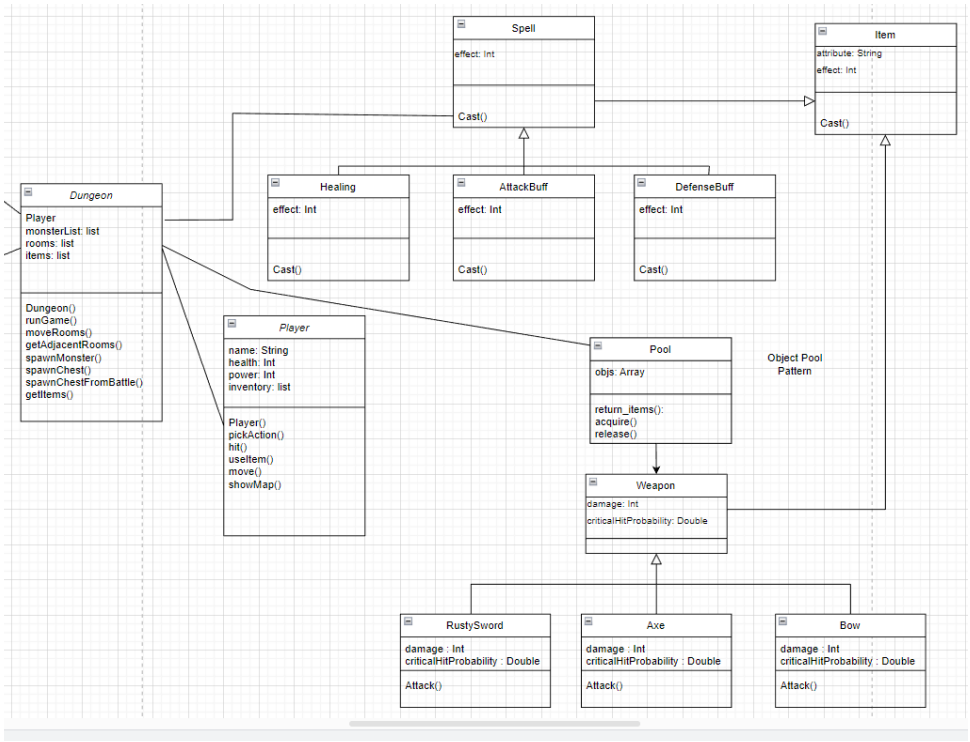
- Map - we have defined the map more precisely than we described in our previous project. Currently, we are using a coordinate system to track the location of rooms in relation to one another and to draw our mini-map.
- Items - spells and weapons now both inherit from an Items class since they are similar in design, which helps standardize our design

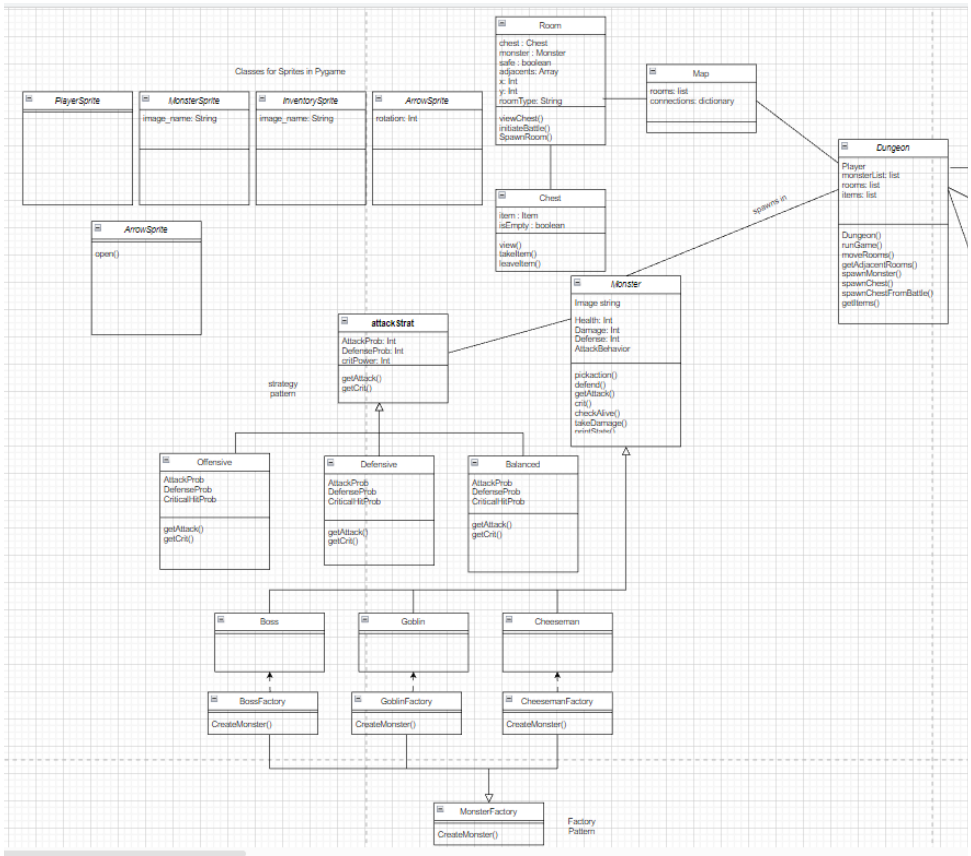
Implemented Patterns -

- Object pool - In order to manage items, we have implemented the object pool pattern to limit the number of total items available to the player. All randomly spawned items in chests are drawn from the pool. This ensures that the player receives a balance of items.
- Strategy - Each monster is instantiated with a randomly generated strategy that determines how it behaves in battle. There are three options: aggressive (which is most likely to attack), defensive (which is most likely to use healing spells), and balanced (a combination of aggressive and defensive).
- Factory - The creation of the monster is not handled through the monster class but rather specific factories for each monster. To create one, the game calls create through one of the factories.

**Class Diagram:**
**https://drive.google.com/file/d/1x8PwVCxOtDwCFdSjgD8FVYPjWmEfL6GE/view?usp=sharing**

There are two diagrams - the top is our full diagram and the bottom is what we've implemented so far for Project 6. See screenshots below.



**Plan for Next Iteration:**

- Fully functional battles between players and monsters, including the final boss fight which ends the game
    - Player can use spells or change weapon during battle
    - Battle ends when either the player or monster's health is less than 0 or when the player runs away
- Full inventory system
    - Allow a player to view their inventory at any time
    - Player can change their weapon or use an item in a safe room
    - Player's stats (health, attack, defense) display along with inventory
- Full implemented player design with health / stats
- Observer pattern for notable events
- UI complete for battles, inventory, and player design
- Unit tests through Pytest