Design Patterns

We used creational and behavioral design patterns to create our project.
Creational: Builder
Behavioral: Iterator, Mediator, Strategy

Builder: Builders separate the construction of a complex object from its representation, allowing the same construction process to create various representations.

- In our project, we had one character builder which could be used to create different character types (namely Enemy and Player). This was useful because player characters and enemy characters all had the same functions, with the main difference between them being whether the user controlled them or not. This allowed us to create similar objects that could behave differently based on whether they were enemies or players.

Iterator – Iterators provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

- In our project, we used iterators to cycle through player and enemy action loops. This was necessary because each team's turn consisted of each character on that team taking their action. Iterators were needed to track which character was currently acting and to transition to the opposing team's turn once the current team has finished acting with each of their characters.

Mediator – Define an object that encapsulates how a set of objects interact. – Mediator promotes loose coupling by keeping objects from referring to each other explicitly, and it allows their interaction to vary independently.

- While not an object, functions in game.js, namely, playerAction and enemyAction, acted as a mediator in our program. These two functions allowed for player and enemy characters to interact with each other without explicitly referring to them. This allowed for loose coupling and more freedom in what each character can do.

Strategy – Define a family of algorithms, encapsulate each one, and make them interchangeable. – Strategy lets the algorithm vary independently from clients that use it.

- The playerAction and enemyAction, described above, also allowed for user (and "AI") strategy. Each action that a character had allowed them to choose between multiple options (attack, heal, AOE (area of effect) attack, use item) and multiple targets to act upon. This allows the user to play the game how they want and allows for increased game variety. Strategy is implemented in a different fashion for the enemy (AI) characters. While they can select the same four actions characters can and can also act upon different targets, their choices are determined randomly.