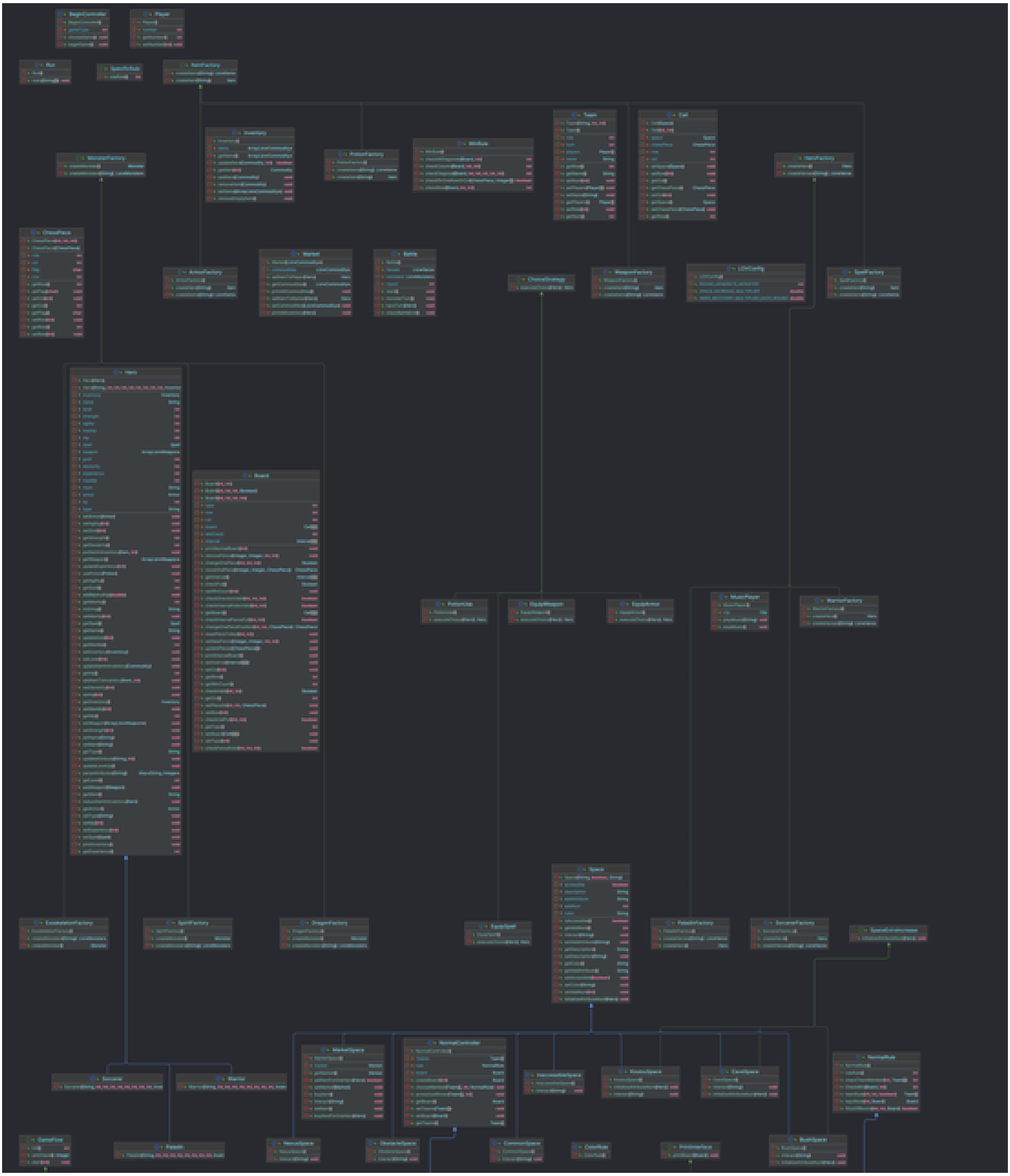
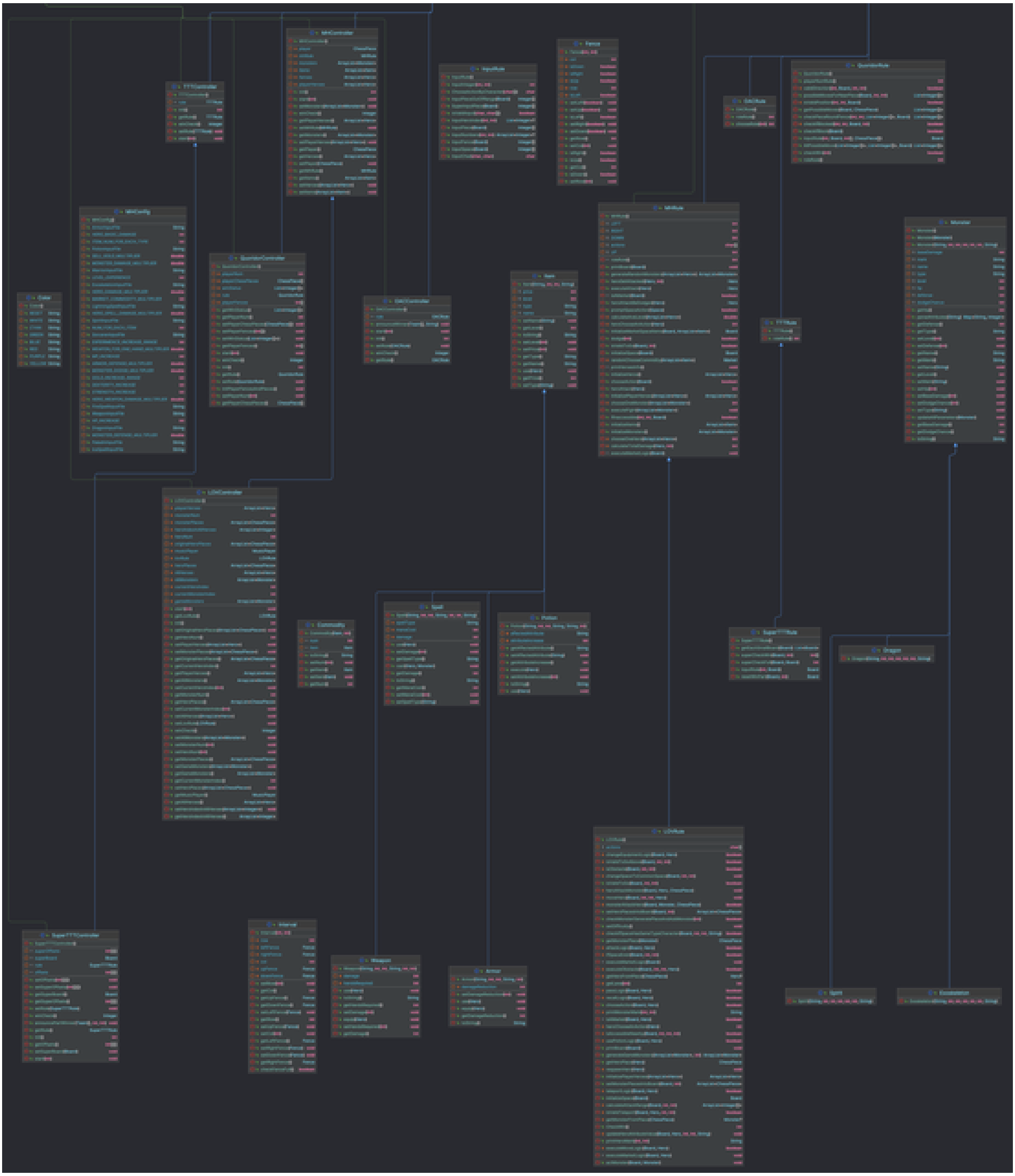
PA5 Design Document

1.UML

You can see UML picture in code dictionary as well.





2.Design Pattern

2.1 Factory Pattern

In the generation of hero, monster, item, and space, I utilized the **Factory Pattern because the construction of instances of these classes is repetitive, similar to how products are manufactured in a factory.**

2.2 Strategy Pattern

In the process of the user changing equipment, **I used the Strategy Pattern because the user needs to replace different items**. Although the detailed logic of the replacement varies, the essence of the operation is still the same—replacement. This makes it very suitable for the Strategy Pattern.

2.3 Decorator Pattern

When constructing a Space, some special Spaces provide certain bonuses to specific attributes of the hero. **In this case, a regular space can be constructed first, and relevant methods can be implemented within it to enhance the specific attributes of the hero who enters the space. This makes it very suitable for the Decorator Pattern.**

3.Legends of Valor Gameflow Control

1. We controlled the entire game flow by using the **GameFlow** interface, **achieving a unified implementation of game initialization, game progression and termination, as well as victory condition evaluation**.
2. LOVController class **implements GameFlow interface and extends MHController class**, which is controller class for Monsters and Heroes, because many variables can be reused.
3. LOVRule class **implements GameFlow interface and extends MHRule class**, which is rule class for Monsters and Heroes, because many methods can be reused. **Rule class will check validation for every action.**

4.Code Reuse

We reuse a lot of code in previous games, **including Board class, InputRule class for checking input validation, Color class, all classes and interfaces in Monsters and Heroes, etc.**