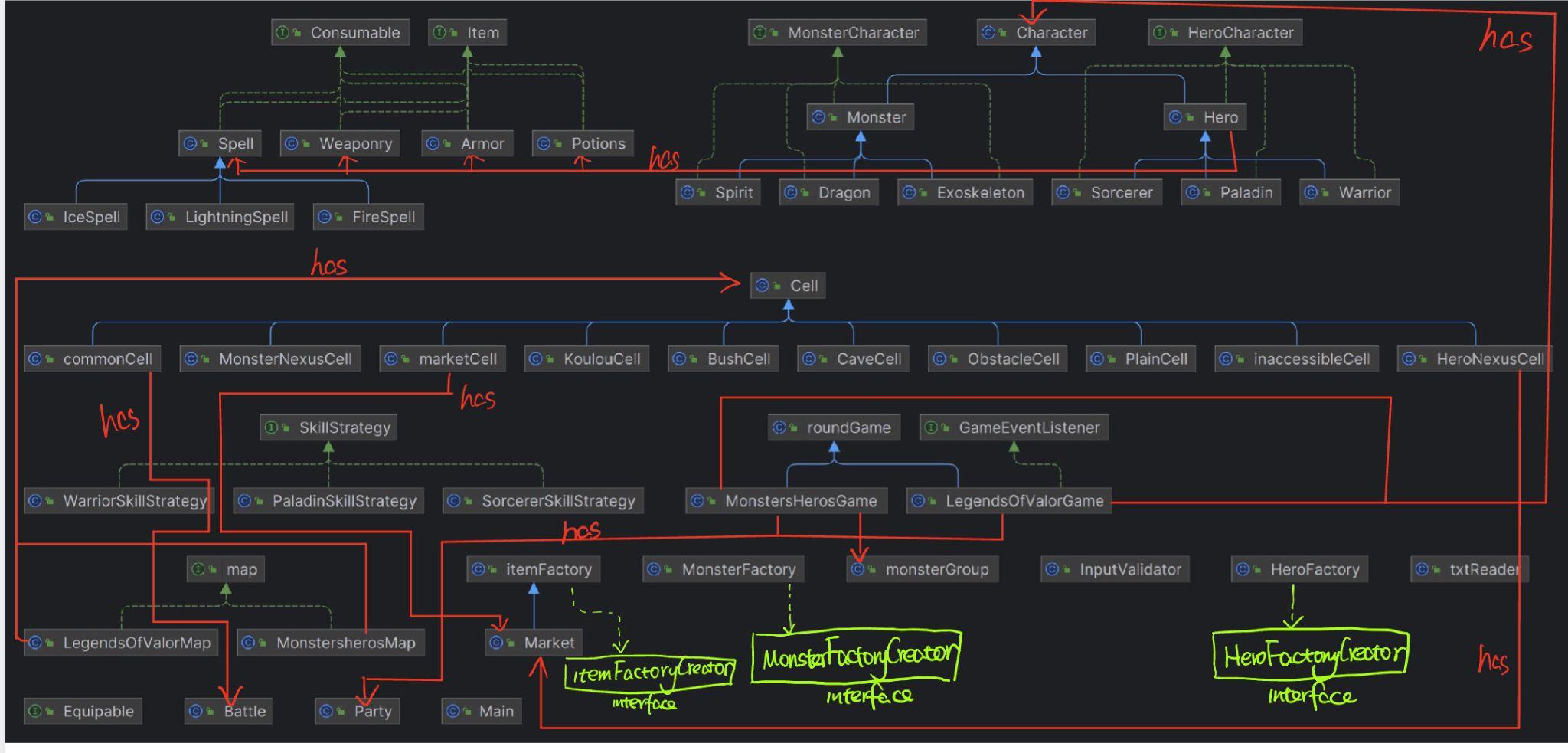
Design Document



For this assignment - Legends of Valor Game, we utilized Jingyi’s structure from the last assignment as a foundation. We first addressed an issue in her market cell setting, where markets in each cell were not distinctly separated. This was rectified, and now the market class is efficiently used in the Hero Nexus Cell. Her previous framework was both extendable and scalable. In the main class, players have the option to choose between playing Heroes and Monsters Game or Legends Of Valor Game. We introduced a new game class, `LegendsOfValorGame`, extending the same abstract round game class as the Heroes and Monsters Game. Similarly, the new game map, `LegendsOfValorMap`, implements the same map interface. We added new cell classes to the cell package, allowing both previous and new cells to be utilized in the game. The inaccessible cell is a feature in both games, showcasing the possibility of creating a new game that incorporates cells from both assignments.

For characters, we've added new hero action methods (such as teleport, move, recall, etc.) and more position methods. This is a change from the last assignment, where movement was based on the whole party.

All character types, item types, and space types are distinct classes, allowing for specific modifications to be made in their respective classes without affecting others.

Patterns Used:

- Factory Patterns: Implemented to create new heroes, monsters, and items.

- Observer Pattern: Utilized to handle event -- a hero reaching the monster's Nexus or vice versa. The game class acts as an observer, monitoring hero and monster movements. When a hero or monster reaches a Nexus, a notification is triggered, and the game class determines the game's outcome.

- Strategy Patterns: Employed to provide skill increases to different types of heroes upon leveling up. Each hero type, extending the superclass `Hero`, requires a unique implementation of `skillIncrease()`. We have developed a set of strategy interfaces or classes representing diverse skill-gaining behaviors.

Project Features:

1. Players can quit the game at any time by entering 'q'.

2. A spell can be used up to 3 times.

3. A potion is designed for single-use.

4. When a hero or a monster is killed, their body remains on the map for that round. Monster bodies disappear after the round ends, while heroes respawn.

5. Game statistics are provided at the end of each game.

6. Using two hands to hold a weapon increases attack points.

7. Monster actions include moving downward unless encountering an Obstacle space (turn left or right) or a hero (triggering a fight).

8. New monsters spawn every 8 rounds.

9. Every round, each surviving hero regains 10% of their HP and mana.

10. Upon death, monsters drop gold and experience points, calculated as 500×monster\_level and 2×monster\_level, respectively, and distributed to all heroes.

11. Heroes cannot teleport to a space within the same lane.

12. Heroes in their nexus are prompted to visit the market first, followed by selecting an action.

13. If a hero makes an invalid action, they must choose an action again within the same round.

14. Players can access information about all heroes and monsters during a hero's turn.

15. Restrictions on hero teleportation include not moving ahead of the teleported hero, avoiding spaces occupied by other heroes, and not teleporting behind a monster in the lane.

In summary, designing a new game using similar board maps, characters, items, and spaces is straightforward in our framework. Additionally, it is effortless to introduce new characters, items, and space types into our games.