Design Rationale

There are two distinct maps in the game world, the Abandoned Village and the Burial Ground. In the maps are different enemies: Wandering Undead and Hollow Soldier. These enemy characters are placed in the game world and serve as adversaries that the player can attack and loot the spoils that they left when they die, such as Key, Healing Vial and Refreshing Flask. These items or spoils can be picked up by the Player and with it they return a Replenishing Action. When the action is chosen by the Player to be executed, player's health or stamina will be regenerated by a certain amount. However, these items are not guaranteed to appear when the enemy dies, only by chance.

In terms of concepts and theories, the DRY principle has been followed by encapsulating similar functionalities into reusable components. For example, the Replenish Action class is designed to handle health and stamina replenishment for various items. Instead of duplicating code for each item, we have a single Replenish Action class that can be reused by multiple items with different parameters. Replenish Action class also has low coupling because it interacts with the Actor and Item classes without relying on unnecessary external dependencies. Dependency Inversion Principle states that high-level modules should not depend on low-level modules, but both should depend on abstractions. The proposed system follows a design where high-level modules for example, actors interact with low-level modules such as items through abstractions like the Action class.