

Assignment 3 Design Rationale: CL_AppliedSession10_Group1

REQ1: The Ship of Theseus

The goal of this requirement is to allow the Intern to interact with the computer terminal in order to travel between different locations such as Polymorphia, the Factory, and a new moon called Refactorio. Other than that, the computer terminal also sells Theseus, a teleporter that allows the Intern to teleport to anywhere within the Intern's current location.

A new abstract Moon class extends GameMap to provide a consistent interface for creating and managing different game maps. Currently we have Polymorphia, Factory and Refactorio but in the future more maps can be added without modifying the existing code (Open/Closed Principle). It also supports the Liskov Substitution Principle (LSP), as Moon can be used interchangeably with GameMap.

The Terminal class has been modified to allow the addition of multiple Moon instances and provides travel actions to move between them. A new TravelAction class was added to handle the action of moving the player between moons. The travel logic is encapsulated within the Terminal and TravelAction classes, improving maintainability and reducing the risk of unintended side effects.

In the Application class, the main method is simplified to delegate the world initialization to a new method initialiseWorld. The initialization of the world and game maps has been abstracted into separate methods. InitialiseWorld method sets up the world, including creating and adding game maps (moons) to the terminal. The initialisation of each moon is done in their initialization methods, which is called inside the get methods of the respective moons. This improves code readability and maintainability by modularizing the setup process. It adheres to the Single Responsibility Principle (SRP) as each method now has a single, well-defined purpose. This also helps reduce connascence of name, making it easier to understand and modify the setup logic.

Lastly, a Theseus class is created and implements the Purchasable interface so that it can be purchased from the Terminal. It can also be used by the Intern to teleport around the current map with TeleportAction. This implementation is supported by the TheseusFactory class that implements the PurchasableFactory interface. This is just extending the purchasing system we created in Assignment 2 where each class has a well-defined responsibility (SRP), and is easily extensible (OCP) where new purchasable items are added without modifying the existing code.

The con to the current TravelAction is limited to only moving to a fixed player spawn location. Future implementation could be done by injecting the destination coordinates via constructor injection or setter injection to make the travel feature more versatile. This change would support not just the player teleportation but also allow the teleporting of other actors and enable the travel to multiple spawn locations. This not only improves the flexibility of the code, but reduces the coupling between the TravelAction class and the specific coordinates when using dependency injection.