Design Rationale REQ4: The Burial Ground

Introduction

In this section, I will provide a detailed explanation of the design and implementation of the classes: locked gate, lockedgateBG, OpenLockedGateAction, OpenLockedGateBGAction and Oldkey. The classes Application and WanderingUndead was refactored to cater the requirement too.

Design Implementation

1. Refactored the Application class

   This class was refactored to add the burialgroundmap to the world and give the nature of the groud.
   This class was also added with the lockedgates to switch from maps.

2. Created the LockedGate class

   This class inherits the ground as this is a feature of the ground. This gate is added to the ancientwoods map.
   The player can step or enter only if the player has the oldkey.
   When the gate is opened the map changes from ancientwoods to the burialground hence the players next exit will be in burial ground.

3. Created the LockedGateBG class

   This class inherits the ground as this is a feature of the ground. This gate is added to the burialGround map.
   The player can step or enter only if the player has the oldkey.
   When the gate is opened the map changes from burialgorund to the ancientwoods hence the players next exit will be in ancientwoods.

4. Created the OpenLockedGateAction class

   This class is inherited by the action class.
   When of the players exits is the locked gate and the locked gate creates a new instance of this class after checking whether the player has the key. This gate changes the capability of the gate to unlocked, and description is printed in the console.

   This class is used when the lockedgate is opened.

5. Created the OpenLockedGateActionBG class

   This class is inherited by the action class.
   When of the players exits is the locked gate and the locked gate creates a new instance of this class after checking whether the player has the key. This gate changes the capability of the gate to unlocked, and description is printed in the console.

   This class is used when the lockedgateBG is opened.

6. Created the Oldkey class

   This class is inherited from the item class, with a displaychar of '-'.
   This class overwrites the pickupaction to tailor the method to it. When the item is picked up the status of the player changes to "HAS_KEY"
   This has no dropaction as once pickedup it cannot be removed.

Design Rationale REQ4: The Burial Ground

7. Refactored the wanderingundead class

   This class is implemented by a interface as deathrewarder.
   when the player kills the actor, a random counter makes the probability of 25% and drops a old key according to this probability.

   Then the actor becomes unconscious.

Summary

In summary, the detailed design rationale encompasses various object-oriented principles, including encapsulation, the SRP, the Open-Closed Principle, the Interface Segregation Principle, and modular and flexible design practices. The implementation details considered in each design decision ensure code maintainability, code reusability, and overall system flexibility.