Assignment 3 - Rationale

Actions package

New classes:

LightBonfireAction

This class is created for the player to light a bonfire that is not yet activated. It extends from Action class since we want to add an available action for the player to perform when the player is around a non-activated bonfire. It removes the non-activated ability from the bonfire as well when this action is executed to reduce duplication of lightning action on the same bonfire in the future. Since this class represents only one action that can be performed, it follows the Single Responsibility Principle when it is created.

TeleportAction

This class is created for the player to teleport to a destination. It works the same way as MoveActorAction, except for the description of this movement, which changes to “Player moves to Anor Londo”, for example, instead of “Player moves Anor Londo”. Hence, extending this class from MoveActorAction helps in achieving DRY principle due to similar functionality. Since this class represents only one action that can be performed, it follows the Single Responsibility Principle when it is created.

RangedAttackAction

This class was made to handle attacks made from a range with a ranged weapon. This action is created in the previously mentioned methods AttackBehaviour and the players playturn method when the ranged weapon is tested for. The ranged attack action will also check for the weapons passive. Currently there is only one ranged weapon so it will detect for the Darkmoon Longbow and randomly increase its damage.

OpenChestAction

This class allows the player to interact with chest objects. When the player discovered a chest at his adjacency positions, a description will be shown on the console and the player can open the chest. This class obeys the Single Responsibility Principle as it has only one responsibility which is allowing the player to open the chest. TradeCinderofLordAction This class allows the player to trade the Cinder of Lord to Vendor to get the corresponding Lord of Cinder’s weapon in return. It works similar as the BuyBroadswordAction, BuyGiantAxeAction and

IncreaseMaxHpAction. An instance of this class will be added to the Vendor only when there is a CinderofALord object in the player's inventory.

Grounds package

Existing classes:

Bonfire

There are two new attributes added to this class, including bonfireManager with type BonfireManager, and name with type String. Since the bonfireManager should store and manage all bonfires instantiated throughout the whole game in Application, it is passed in as an argument of the constructor of a bonfire instance to ensure that different bonfire instances share the same bonfireManger instance. Changes were made to allowableActions() as well, where it no longer returns only restAction, but getActions() of bonfireManager is called instead to return several available actions. Since this class represents only one specific ground type, it follows the Single Responsibility Principle when it is created.

New classes:

FogDoor

This class is created to represent a fog door that allows teleportation of the player to a new map. It extends from Ground class since we want to add a new ground type that can be displayed in the game. In order to recognize all destinations that the player can teleport to using a specific fog door, a hash map, which is the attribute's type of this class, is the most suitable choice to store this information due to its ability to store different types of value as a key (fog door' name) and a key's value (fog door's destination). Since this class represents only one specific ground type, it follows the Single Responsibility Principle when it is created.

Chest

This class representing the chest object that we can found on the map displayed as "?". It inherits the ground class so that the player is able to interact with it (game rule, the player can interact with current ground). In this class, it overrides the allowableActions method so there will be a new OpenChestAction object created every time it being called. This allowed the player to interact with the chest object using the open chest action.

Weapons Package

New Classes:

RangedWeapon

The class is created to be a parent class for all ranged weapons in the game. It extends the WeaponItem class since RangedWeapon still counts as a weapon. It will automatically add the RANGED capability to any weapon extending this class. Each ranged weapon would also have to have a range from where the wielder can attack from.

DarkmoonLongbow

The class is created to represent the ranged weapon Darkmoon Longbow. This weapon allows the wielder to attack from 3 tiles away and has a passive crit chance. This weapon is always wielded by the boss “Aldrich the Devourer” and can be obtained by the player by trading Aldrich’s Cinder of Lord to the vendor. The enemies held weapon are tested in AttackBehaviour, while the player is checked in their playturn method. This weapon is also blocked by any ground type that can block projectiles such as walls.

Enemies Package

New Classes:

AldrichTheDevourer

The class is created to represent the second boss in the game. This boss will always wield the Darkmoon Longbow. Since this boss will always attack from a range, the follow behaviour wont be added to him like usual. So I had to add the follow behaviour to all enemies during the attack behaviour if the player was on the edge of their range.

Mimic

This class is created to represent the new enemy that the player might run into when opening a chest. This class applied the Single Responsibility Principle as it only has one responsibility which is representing the Mimic enemy. In this class, it overrides the resetInstance method from its parent class and create new token objects when the mimic is died. It has a possibility to drop 1, 2 or 3 tokens of souls and each token is worth 100 souls.

Behaviour Package

Existing classes:

AttackBehaviour

This class was edited to test if the actor/enemy was either unarmed, using a melee weapon or using a ranged weapon. If they were unarmed or using a melee weapon then it would run like usual. However if they had a ranged weapon then it would loop through an area around them based on the weapons range.

Unclassified classes

New classes:

BonfireManager

This class is created to represent a bonfire manager that stores and manages all bonfires instantiated throughout a whole game, thus allowing easier maintenance of available actions that the player can perform when the player is around a bonfire. In order to recognize all bonfires and their locations, a hash map, which is the attribute's type of this class, is the most suitable choice to store this information due to its ability to store different types of value as a key (bonfire instance) and a key's value (location bonfire instance). Since this class represents only one specific ground type, it follows the Single Responsibility Principle when it is created.