

Rationale FIT2099 Assignment 3

Unified Modified Language(UML)

Requirement 1: New map & fog door (new ground class)

A new class named FogDoor is created and it should extend the abstract ground class. The class should extend the ground class because it is a ground type that the player can interact with. When it is interacted with, it will teleport the Player to a new map which is called Anor Londo from the original map, Profane Capital. The fog door that is placed at furthestmost South in Profane Capital should teleport the player to the furthestmost North in Anor Londo with the same x coordinates. Since enemies are not allowed to pass through the fog door, we added a capability to enter the fog door to the Player from the abilities enumerator class. This ensures that if the enemies were to follow/wander near the fog door they would not be able to enter through the canActorEnter method.

As for the new map, we created a new map in the application class. Modifying the original map, which means the map size is still the same, with different content. After the map is initialised it needs to be added to the world class' arraylist of game maps. The fog door is also instantiated at the application class since it requires both the game maps to move the player from profane capital to anor londo. There is no other way to do this as adding the game map into the world class' arraylist of map is only accessible in the application class. If the game map is declared anywhere otherwise, we would have no way to initialise the game map's actor location, which can only be updated in the world class' addGameMap method. In other words, there is no other way to do it without modifying the engine code.

Requirement 3: Aldrich the Devourer (new class)

Aldrich the Devourer is a new boss in the Anor Londo map, it is similar to Yhorm the Giant. Aldrich is also considered a Lord of Cinder, the same as Yhorm. Therefore, the new class AldrichTheDevourer extends the LordOfCinder abstract class since it is one of the lords of cinders. Aldrich would drop 5000 souls and Cinders of a Lord when he is defeated. Cinder of a Lord is an item which can be traded in with the vendor to get the weapon of the fallen Lord of Cinder. For example if we take Aldrich the Devourer's Cinder of a Lord to the vendor, it can be traded in for a Darkmoon Longbow (which is the weapon of Aldrich). More on trading Cinders of a Lord and the item itself in the subsequent parts.

Aldrich the Devourer holds a weapon named Darkmoon Longbow which is a different weapon from the existing ones, in a sense that it is a ranged weapon instead of a melee weapon. As usual, each weapon has its own class due to their own uniqueness, and since a ranged attack behaves differently from a melee attack, a new attack action class is also created. The exit of the chamber in which Aldrich is in is covered with floors so that it would not be able to escape since it does not have the TOENTERFLOOR ability. Aldrich will also not be active unless engaged by the player, which is why the RangedAttackAction is added only if the player is within Aldrich's adjacent squares. More on its weapon and ranged attack In its subsequent parts.

Requirement 3: Darkmoon Longbow & RangedAttackAction (new classes)

Darkmoon Longbow is the weapon used by Aldrich the Devourer and it is a ranged weapon which behaves differently which is why a new RangedAttackAction class is added. These 2 classes are pretty straightforward, Darkmoon Longbow would inherit the weapon item class to initialise it's characteristics such as weapon damage, hit rate and its verb.

RangedAttackAction is the attack action that is given to Aldrich to attack the Player when it is within 7x7 squares of Aldrich himself. This ranged attack action would implement the weapons passive skills, such as the 15% chance of double damage (critical hit) and also the ranged weapon passive. The ranged weapon passive will be implemented in a separate method called getAction. This getAction method will be called in AldrichTheDevourer's playTurn method, it first checks if the rangedAttackAction is within the list of behaviours, and if it is then it will return the RangedAttackAction. RangedAttackAction is mostly similar with AttackAction in terms of execution. The checking of 7x7 squares is done in the getAction method where the NumberRange engine class is used to iterate through all squares to check if the target is within range of firing the bow. If there is a wall blocking within the trajectory, it would return the action as well, but it would be considered a missed attack.

Requirement 4 (Chest/Mimic)

I created a Chest Class which extends/inherits Ground abstract class so that the player can interact with it in the adjacent square of the chest. To interact/open the chest, an action is needed to add to the Chest Class. Therefore, I created a new OpenChestAction class which inherits Action class. The Chest has OpenChestAction, so the player will get the action if the player is on the "exit" of the chest. Hence, the player can interact with the chest. Since there is 50% chance to spawn Mimic, so I have created a new Mimic Class which inherits Enemy Class. The OpenChestAction has Mimic Class because at the execute part of the OpenChestAction, I use a random number generator to make sure that there's 50% chance to summon a Mimic. So there's 50% chance, at the execute part, I will create a new object of Mimic and replace the Chest. There's also 50% chance to have token of soul, so the token of soul is also associated with OpenChestAction, so that at the execute function, the new objects of token of soul can be created and replace the chest. The Mimic has FollowBehaviour Class so that it can follow the player. It also associated with AttackAction so that the player can get the attack action and attack it if the player is at the adjacent square of the Mimic.

Requirement 5 (CindersOfTheLord and TradeCindersOfTheLordAction) (Trading CindersOfTheLord with FireKeeper)

I created a new class CindersOfTheLord which inherits PortableItem since its portable and droppable. YhormTheGiant and AldrichTheDevourer are associated with CindersOfTheLord, a new object of CindersOfTheLord are added to each of their inventories. So that when they are killed by the player, the CindersOfTheLord will drop and replace them. The constructor of CindersOfTheLord also takes in the LordOfCinder parameter so that we can use the LordOfCinder at the CindersOfTheLord class and differentiate whether the CindersOfTheLord is dropped from Yhorm The Giant or from AldrichTheDevourer. It is important because each of them can be used to trade with the firekeeper to exchange different weapon.

Since it is different to BuyAction Class created, I have created another TradeCindersOfTheLordAction. This action is added to the FireKeeper Class, so FireKeeper has TradeCindersOfTheLordAction so that the player can get this action after checking through his inventory. If he is at the adjacent squares of the vendor and a CindersOfTheLord object is found at his inventory, then this action will be added to the action list hence the player can get TradeCindersOfTheLordAction to trade for the weapons corresponding to the cinders of the lord. The class variable lordOfCinder which will also be used to check the origin of this cinders of the lord, if it is from Yhorm The Giant, the player can only trade for Great Machete, if it is from Aldrich The Devourer, then the player can only trade for DarkmoonBow.