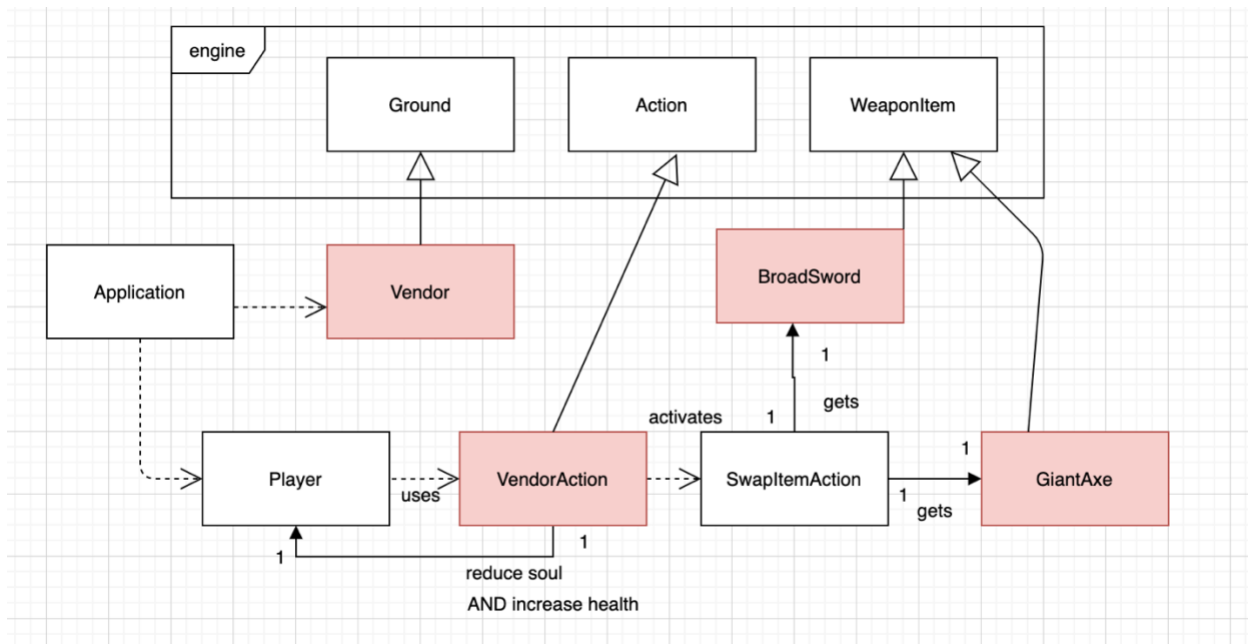


Lab 3 Team 4
Design Rationale
Yo Kogure

Requirement 8: Vendor



The souls can be traded to buy a new weapon and to upgrade the Player's attributes (stats) through a vendor. When the Player buys a new weapon, the weapon in the current inventory will be automatically replaced with it. Replacing the weapon will cause the old weapon to be removed from the game(drawn from the specification sheet).

The class diagram of Vendor is pretty simple. I have made Vendor to be inherited from Ground, so it will sit as a type of ground in the Firelink Shrine. Then, player calls something called VendorAction which inherits from Action. Well, it could have been something called VendorManager just like ResetManager, but I wanted it to use the current template that Action class has, so selected this method.

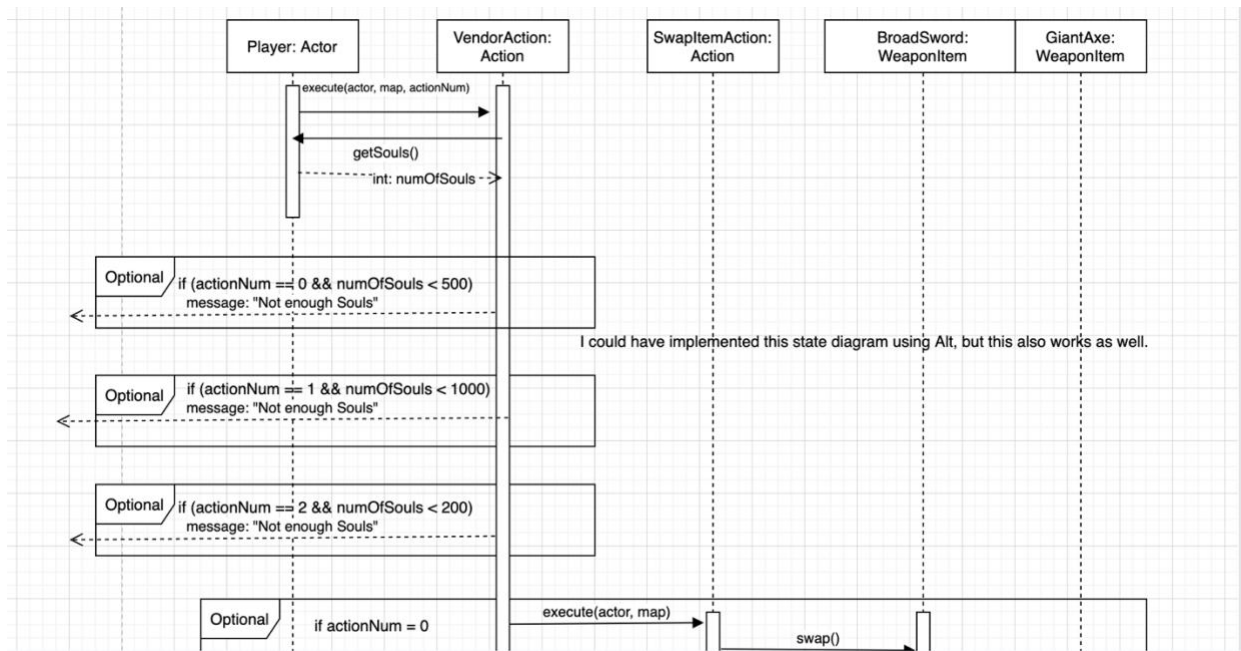
Vendor is just an entity on the ground.

Application will depend on Player to trigger VendorAction. Detailed coding later on, but I will override execute() method in VendorAction, to accept another argument called integer actionNum.

actionNum 0 will represent purchasing BroadSword.

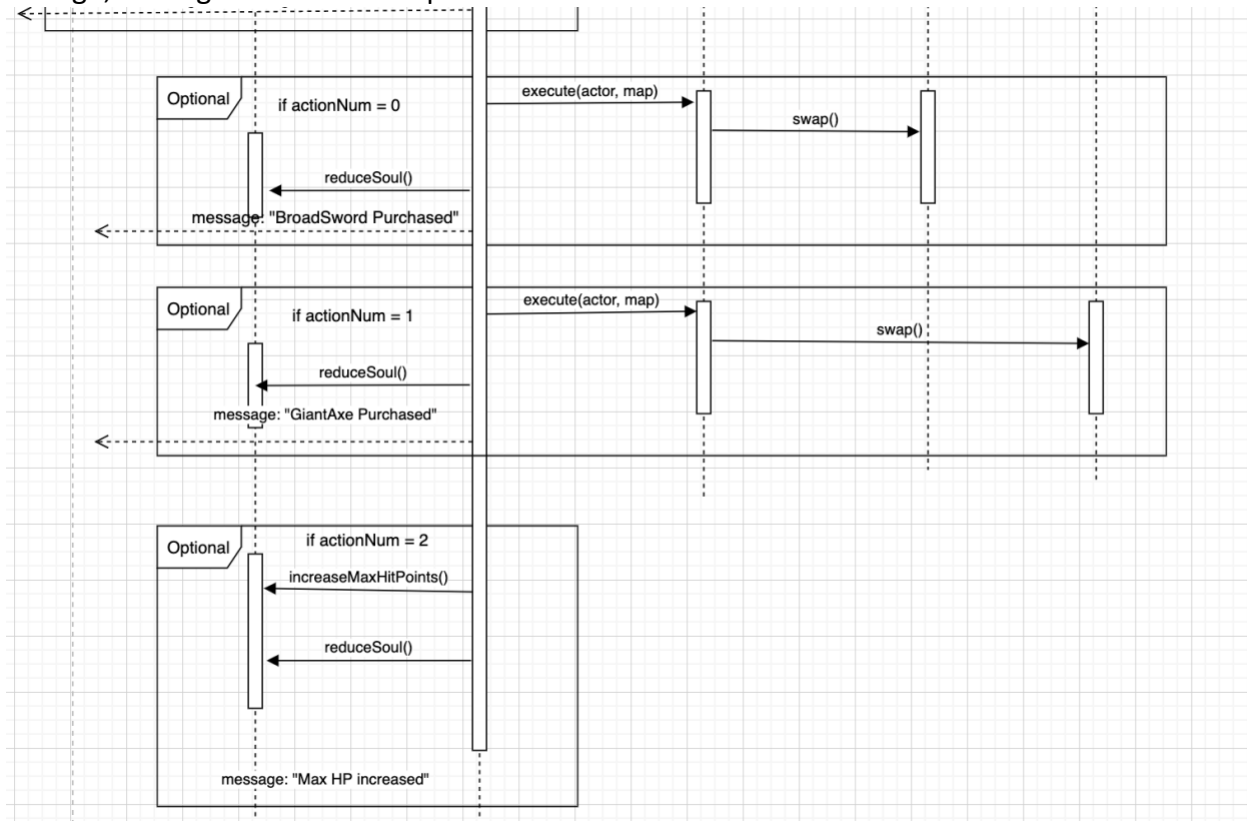
actionNum 1 will represent purchasing GiantAxe.

actionNum 2 will represent purchasing maximum Hit Points of a player. There is no option to buy StormRuler and Yhorm'sWeapon, as they are not inside the Vendor options.



So, Actor will first execute VendorAction which inherits from Action. it will call `getSouls()` to obtain the number of souls that player has. Right after that process, we will have to perform checks to see if player has sufficient souls. All of these are written as if statements, depending on which `actionNum` the player chose. Also note that `actionNum` is always either 0, 1 or 2 and it is made sure in the earlier part of the process when executing.

If player has insufficient souls, it will return corresponding error message, and if they have enough, it will go on to the next part.

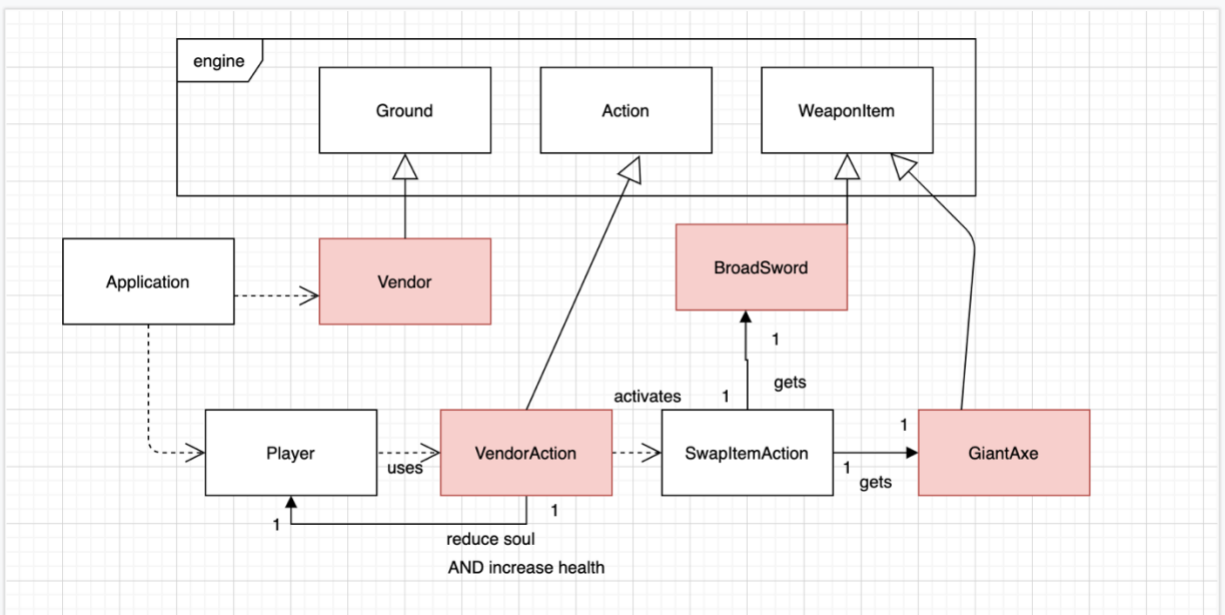


If actionNum = 0, purchase BroadSword is selected, since I have made sure they have enough souls, it will execute() another action called SwapItemAction. This is an action defined already in game file, and is used to obtain new weapon/replace old weapon with new one. Therefore, it will have an association 1 to 1 from SwapItemAction to BroadSword and GiantAxe. After performing weapon swap by swap()(detailed method will be implemented later), it will call reduceSoul() to reduce corresponding soul price as a payment(500). It will also give out message that it has been purchased successfully.

Very similar interactions for the case if actionNum = 1, except it is for buying GiantAxe of price 1000. Since if-else statements and if statements have no difference in this specific interaction diagram, I have decided to apply Optional, to help us visualize better for 3 conditions.

Now if actionNum = 2 is selected, it will increase the maximum hitPoints of the player, and also reduces 200 souls as a payment. Lastly, it will message completion message and concludes the

end of Vendor interaction diagram.



All associations in this part is 1 to 1 as each one of their object only associates with one of the other. VendorAction also associates with Player, because it will have to call its method to reduce soul and also to increase health.