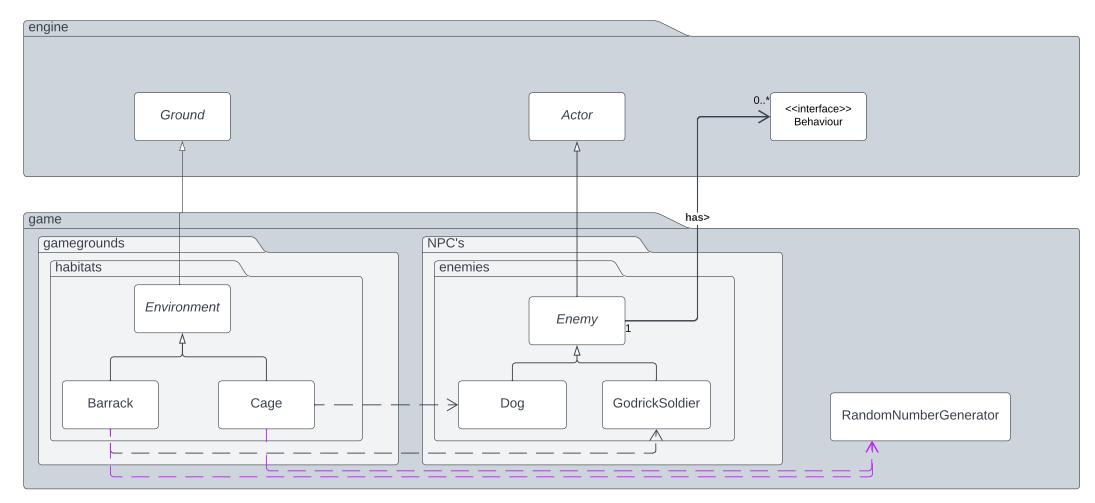


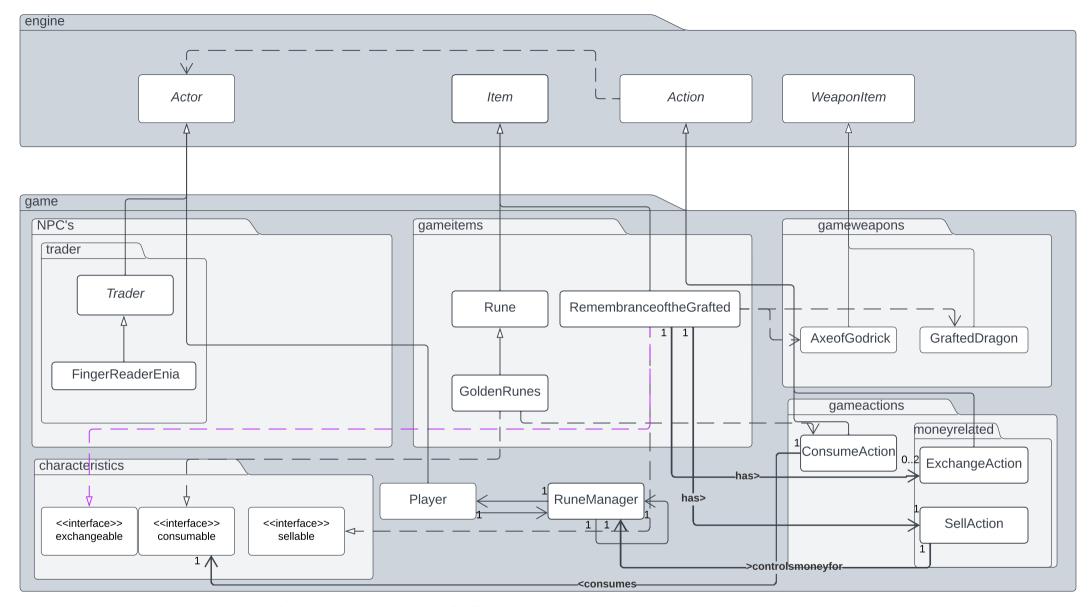
In this requirement we add two new ground types, Cliff and GoldenFogDoor. Both new ground types extend from the Ground class like all other previously created habitats keeping the system extensible.

The GoldenFogDoor gives the player access to new maps in the game, utilising the new TeleportAction whose sole purpose is to transfer the player between maps, abiding by the SRP.



In this requirement, we essentially just added two more enemies and two new spawning environments for these enemies. The two new enemies were Dog and GodrickSoldier, spawning from Cage and Barrack respectively.

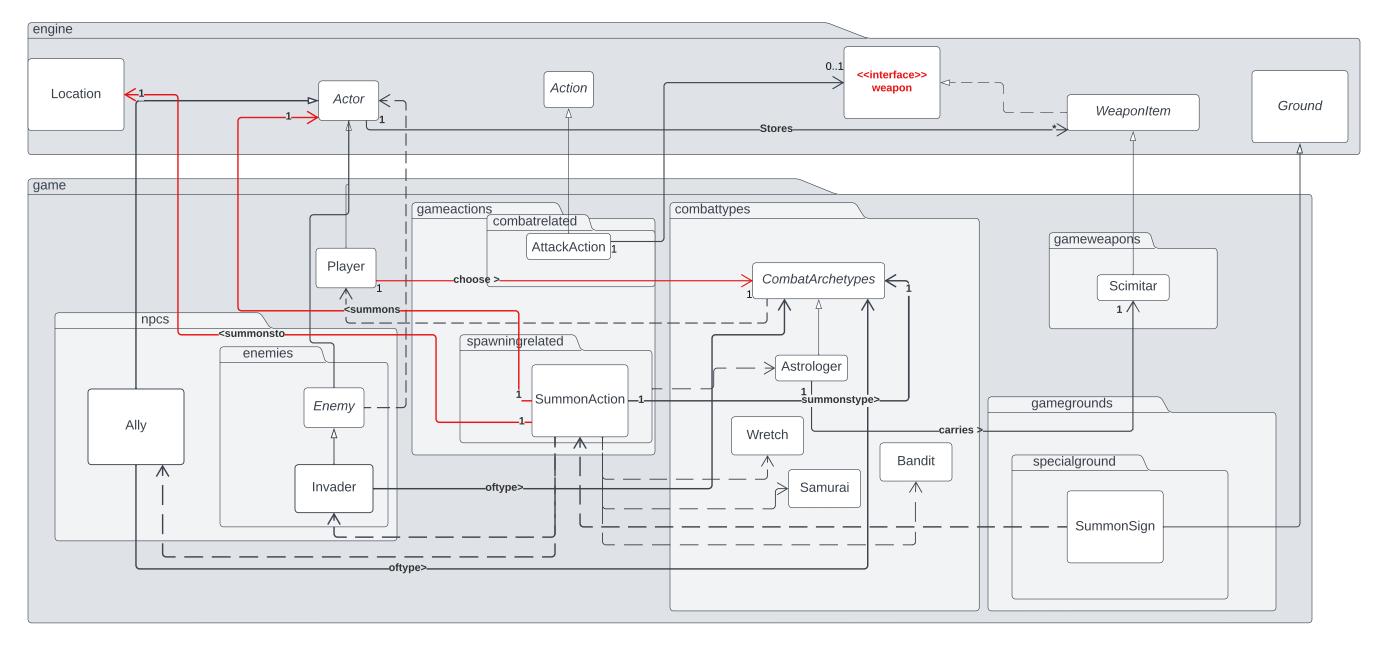
We were easily able to add in new enemies and spawning environments, without changing anything in the existing code, exemplifying how the code follows the open/closed principle.



In this requirement, we add the new item [Remembrance of the Grafted] will be dropped. And because the Remembrance of the Grafted could be exchanged to weapons, we add an interface [Exchangeable] and [Exchange Action] to achieve this requirement. The exchangeable weapon also add in this game, [Axe of Godrick] and [Grafted Dragon].

Also, we add other items in this game, [Golden Runes], which is extend from Rune this abstract class, and could be consumed to swap into runes. The new trader, [Finger Reader Enia] will have abality to exchange the Godrich the Grafted into weapons and buy the item from player (sellable).

All the attribute are extend from the original abstract class, we just give these attribute proper ablity and correct attribute to achieve the requirements.

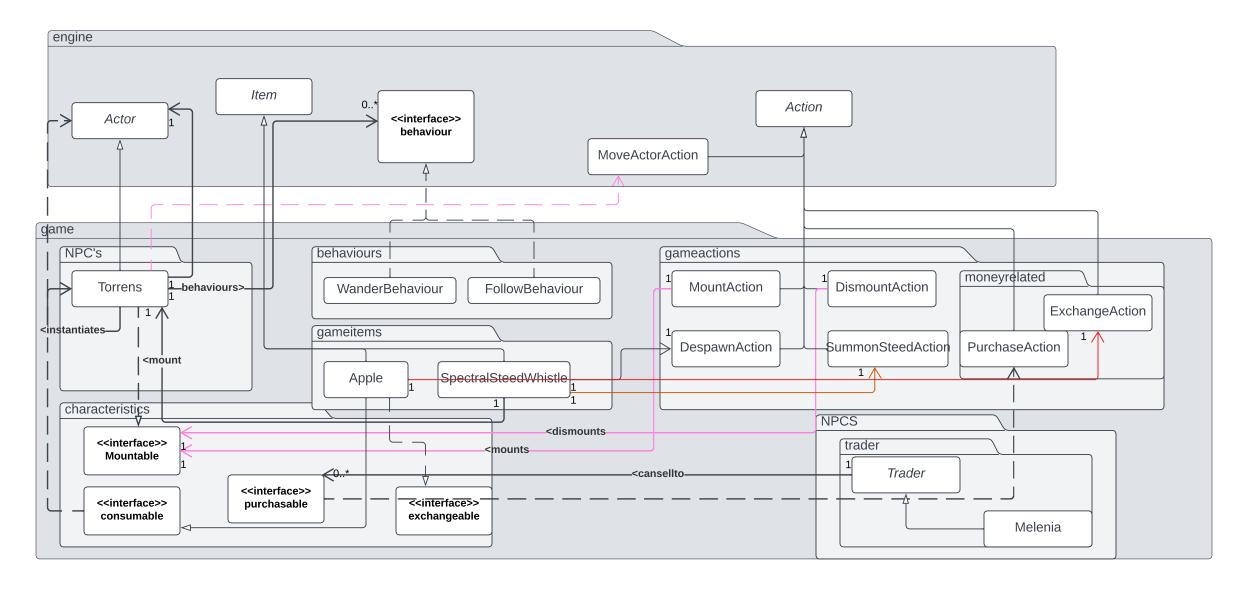


The first part of this requirement was to add a new combat archetype called Astrologer. The Astrologer class was easily added, with no code needing to be modified to accomodate for the new class, showing the code abides by the open/closed principal.

The second part of this requirement was to add in the allies/invaders.

The Invader class was essentially adding in another enemy, so obviously we extended it from the Enemy class again following the OCP. The Ally class however is not an enemy, therefore making it extend directly from Actor was the better choice.

The SummonSign is a new Ground type that can be interacted with. Via the SummonAction class the player can spawn randomly either an Ally or Invader, with a random combat type.



In this requirement we implemented our creative requirement which was just a horse which adds movement abilities.

To mirror the actual game we implemented a singleton class for torrens as after you get the whistle you can summon it whenever and it is always with you as long as you have the whistle.

Torrens is an actor as we want it to basically be an NPC which follows the player and lets them get some movement options.

To implement this design we utilsied the dependency inversion principle by using the mountable interface. This is useful because we can extend it further with similar mountable things and utilise this to have it connect to another actor.

We also implemented an apple which doubles as being exchangable for the whistle but also a consmable, this was to test how extendable our design was as we were able reuse the consumable interface therefore the DIP was used and is helpful in extending the design.

The horse also follows the player whilst on the map unless mounted then it will follow through to other maps as well without needing further summoning.

The design for the mount utilises a lot of actions even if they are small, as this way we can reuse these actions to build more complex ones, as well as abide by the Single Responsibility Principle.