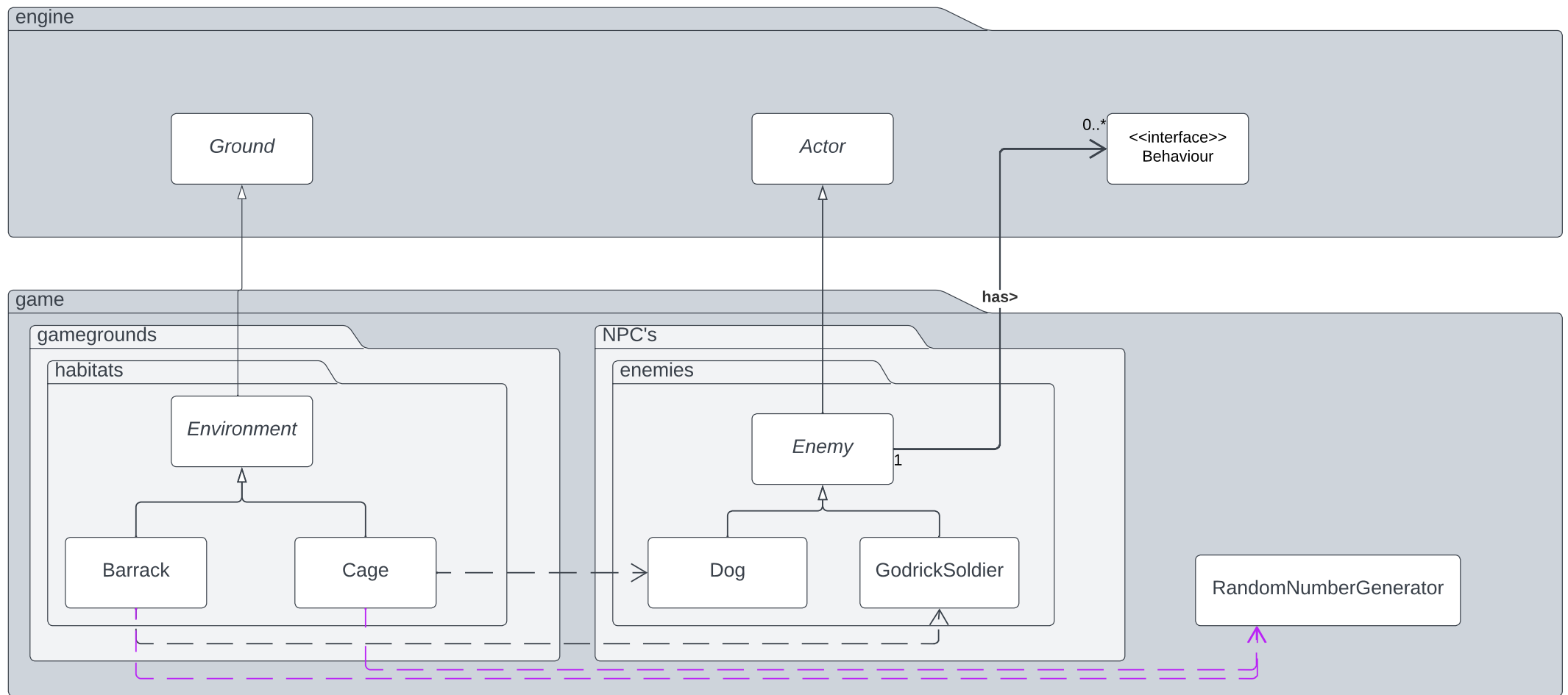


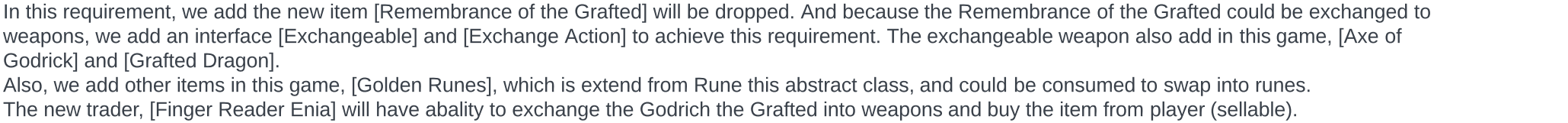
In this requirement we add two new ground types, Cliff and GoldenFogDoor. Both new ground types extend from the Ground class like all other previously created habitats keeping the system extensible.

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.



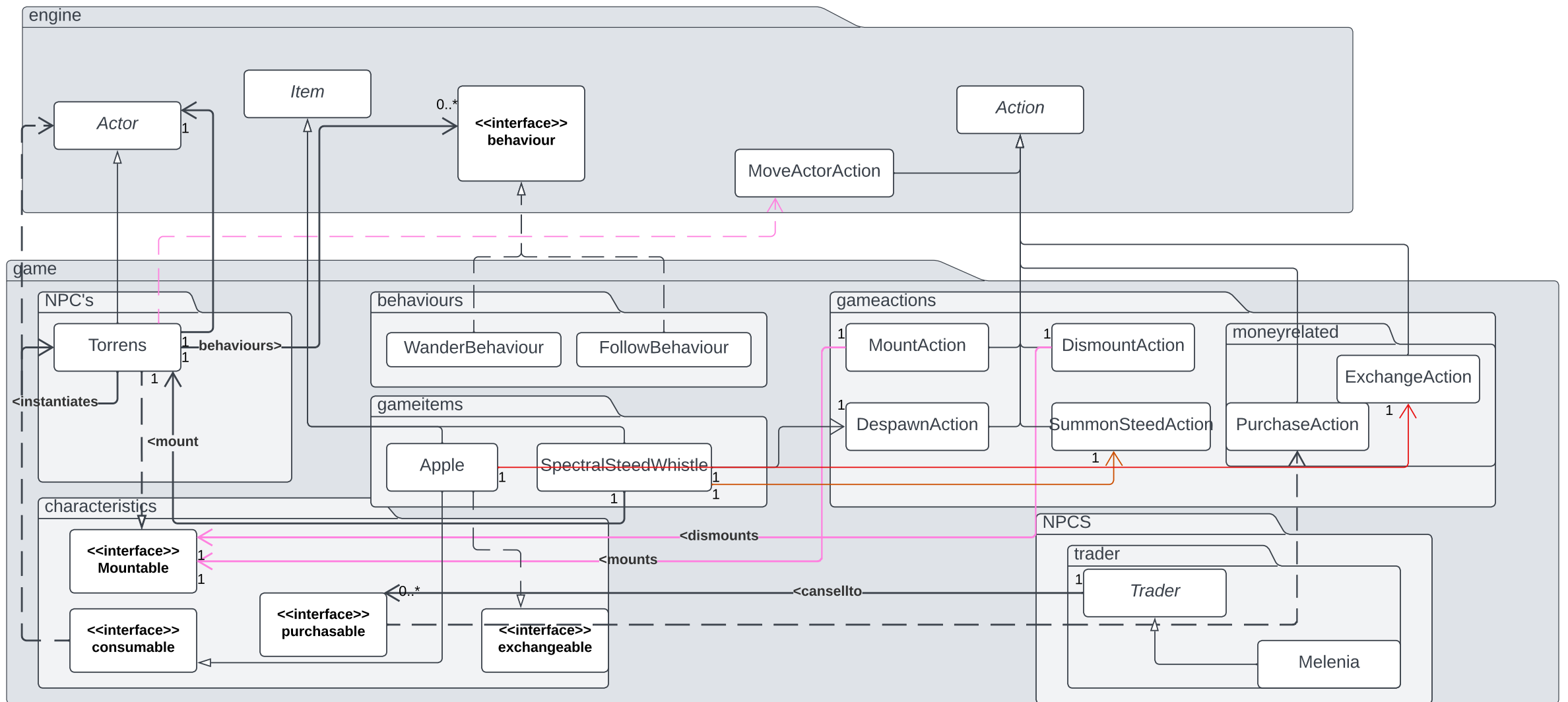
In this requirement, we essentially just added two more enemies and two new spawning environments for these enemies. The two new enemies were Dog and GodrickSoldier, spawning from Cage and Barrack respectively.

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.



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





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 utilised 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 exchangeable for the whistle but also a consumable, 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.