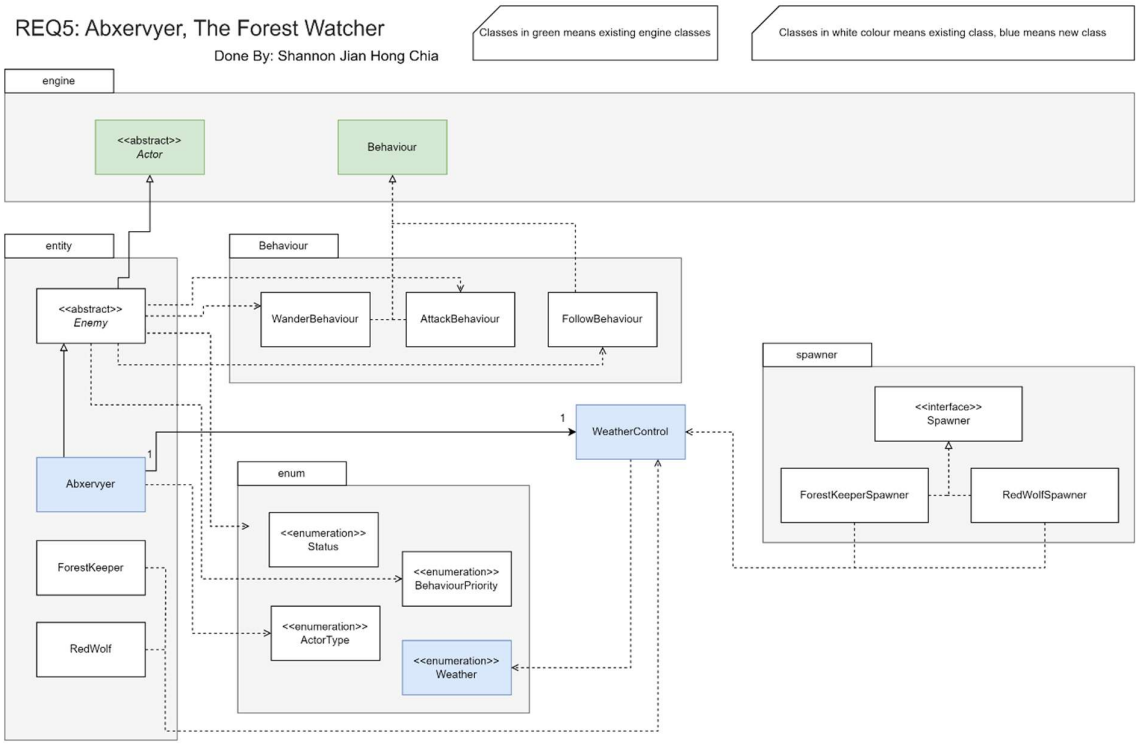
Design Rationale: Req 5 (Done by Shannon)



Brief

The UML diagram represents requirement 5, where 3 new classes were introduced, 2 new concrete class and 1 enumeration.

Within this requirement, there are 2 new major implementations that were carried out. 1st being the new boss – Abxervyer, followed by a new Weather system that can change the weather in the forest, and provide certain buff to other entities depending on the weather.

Weather

Weather

There are 2 types of weather that will be cycled within the game, which are sunny or rainy, and depending on which type of weather, it'll bring certain buff to certain enemies and affect the spawn rate of enemies as well.

WeatherControl

The weather control class is designed to manage the current weather conditions in the game (SRP), allowing for changes between sunny and rainy weather. It also updates the weather periodically based on a specified frequency, which is 3 days. This design emphasizes of simplicity, encapsulation, and ease of extensibility.

There is a getter for the current weather, that is set to a static method, to allow for other classes such as the spawner to access it without needing to create a weatherControl instance (for global game state), a method to update the weather, and a method to change the weather to the other vice versa.

This class also depends on the “Weather” enum, promoting dependency on abstractions rather than concrete implementation (DIP).

Abxervyer

The Abxervyer represents a boss enemy that extends the Enemy class (LSP) and introduces specific behaviours, such as changing the weather every 3 game turns.

This class integrates with the ‘WeatherControl’ to receive updates on weather conditions during its turn, it is done by overriding the ‘playTurn’ method. Upon being defated, it will use the designated gameMap destination to create a new gate instance.

The class is also open for extension, allowing for the addition of new behaviours or features without needing to modify the existing code (OCP).

Buff

Buff such as increase spawning rate, attack buff, or healing buff is done within respective class with the usage of an if, else if condition.

Forest Keeper Spawn Rate	The spawner will call the static getCurrentWeather() method, if it’s sunny, it’ll have this specify spawn rate, else if it’s rainy, it’ll have this other spawn rate
Red Wolf Spawn Rate	
Forest Keeper Heal	Within the ForestKeeper playTurn method, it’ll call the static getCurrentWeather() method, if it’s rainy, it’ll heal 10 hp every round, else no buff
Red Wolf increase hit rate	Within the redWolf playTurn method, it’ll call the static getCurrentWeather() method, if it’s sunny, it’ll have a higher damage, else no changes.