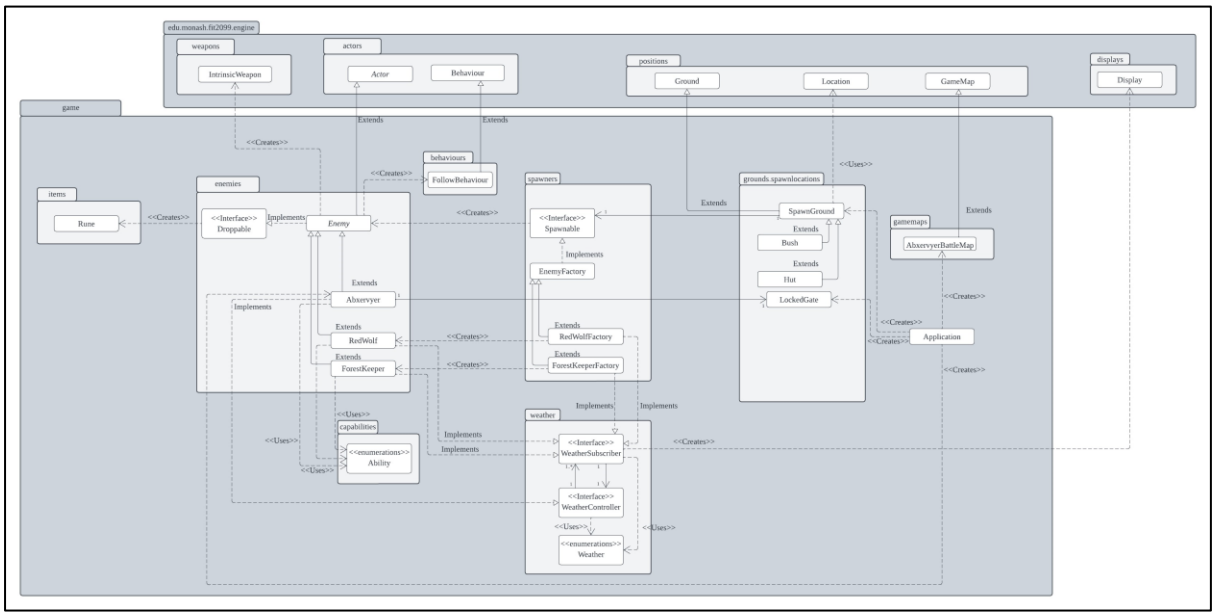
FIT2099 Assignment 2 Design Rationale

Requirement 5:



- The boss is added by extending from the Enemy class which implements the Droppable interface. This helps to ensure that the drop method will be implemented in the boss class as it may drop items after being defeated by the players. By extending from the Enemy class, we can avoid code repetition (SRP).

- Since the boss can control the weather of the map, instead of containing a map in the boss class (which may help to manage the effect of weather for the spawning location and the actor), we introduce two new interfaces, WeatherController and WeatherSubscriber. WeatherController basically represents a remote control for the weather and it can manage those that will be affected by the weather, and hence it is implemented by the boss. Meanwhile, WeatherSubscriber is implemented by those game entity that may be affected by the weather. By separating the two interfaces, we can follow ISP as each interface consists of methods that are only specific to the responsibility they managed.

- By using the two interfaces, we can also achieve OCP principles as it is extensible without needing to modify the code. If there is a new enemy that can control the weather, they can implement WeatherController while if there is a new game entity that is affected by the weather, they can implement WeatherSubscriber.

- The enemies that implement WeatherController will consist of a list of game entities that implement WeatherSubscriber. By having this association, this makes the WeatherController to be easier to notify the weather to the subscriber and call the weather effect to act. For those WeatherSubscriber, they may have reacted differently for different weather. By implementing it as an interface, it enforces the implementation of the weather effect to be done in the class that implements it. This also helps to achieve SRP as the WeatherController is responsible to control the weather and report the current weather only while the WeatherSubscriber will do the action based on the weather effect.

Pros:

- Achieve OCP as it is easier to extend without needing to modify the existing code.
- Each class manages their own responsibilities causing a high cohesion and hence will be easier to manage and maintain.

Cons:

- There is only two weathers for now and the weather is implemented as an Enum. The weather subscriber uses if-else to check the weather and apply the respective effect. So, if there is a new weather introduced, the effect of the weather might need to be explicitly coded out in every class that implements WeatherSubscriber.

Future extensions support:

- Implements WeatherController for new enemy that can control the weather. The details implementation will be forced to complete in the class as well.
- Implements WeatherSubscriber for game entity that is affected by weather. However, the effect of the weather on the game entity needs to be coded out explicitly as different game entity may react differently to the same weather.
- For a new weather introduced, add the weather into the Weather Enum.