REQUIREMENT 4 - Design Rationale

Classes Modified/Created	Roles/ Responsibilities	Rationale
WeatherWizard	The WeatherWizard wanders around and changes the weather. Upon unconsciousness, it explodes and respawns in the same location.	**Solution** • The main purpose of the WeatherWizard is to have the ability to access a method that executes a weather change every turn. This is used through the "playturn" method. • In order for the WeatherWizard to continue changing the weather despite being a killable character, I gave it the ability to respawn when hit. Alternatively, I could have either not given him the "addCapability(Status.ENEMY);" line of code which would have meant the tarnished wouldn't have the option of hitting the wizard. This option was a good one however I wanted the wizard to have a more unique and fun design especially given the character is a wizard. The other (worse) alternative would be just giving the wizard 1 million health, however this would have seemed more like a bandaid solution that doesn't fully resolve the problem **RELEVANT SOLID PRINCIPLES** The WeatherWizard follows the Single Responsibility Principle (SRP) by focusing only on weather changes and respawn logic. It aligns with the Open/Closed Principle (OCP) by allowing new behaviours to be added through the behaviour map without modifying existing code. Lastly, it applies the Dependency Inversion Principle (DIP) by relying on the Atmosphere abstraction for weather management, promoting flexibility and easier testing.
Atmosphere	The Atmosphere class manages weather conditions in the game. It allows for random weather changes and provides the current weather status.	**Solution** I used a random technique in order to make the weather unpredictable similar to how the weather acts in real life. I also decided to give the weather around a ⅛ chance of changing. I feel this allows for more tactical fights as, for instance, given the furnace golem is immune to fire damage, you would want to start a fight with it once the weather turns to rain or snow. Given the low likelihood of the weather switching, you can use multiple turns to approach the golem whilst the weather is good. On the other hand, if the weather had a very high chance of changing every turn, the weather changes wouldn't be as strategic as, in this scenario, by the time the tarnished is within one space of the golem the weather might switch to SUNNY thus making a weapon such as the tall axe less powerful against it due to the fact that the tall axe being sunny means a fire status effect is dealt however, as mentioned before the golem is immune to to this effect **RELEVANT SOLID PRINCIPLES** The **Single Responsibility Principle (SRP)** is applied by having the Atmosphere class solely handle weather-related logic, ensuring clarity and maintainability. **Open/Closed Principle (OCP)** is followed by allowing easy modification of weather-changing logic (e.g., changing probabilities) without altering the class structure. Finally, **Liskov Substitution Principle (LSP)** ensures that any changes to how weather affects the game (like interactions with specific enemies) won't break the Atmosphere class, as it only provides weather status without managing gameplay effects directly.

TallAxe	Represents the TallAxe, a weapon that reacts to weather conditions and applies different effects. Depending on the weather, the axe can burn, poison, or deal extra ice damage to enemies.	For this class I wanted to create a weapon that interacted with the weather. Given the added powers that the weapon holds. I also wanted to make sure that it is difficult to obtain thus I put the strength required on 15, thus the player will have to match the weapons power. In order to make this class work I had to change the BurningStatusEffect class to accept parameters for damage and duration. This is because before this requirement burning damage was only done to those who step on fire, thus changing the damage in order to accommodate the new tallaxe fire effect would also change the damage on takes when stepping on fire. **RELEVANT SOLID PRINCIPLES** The TallAxe follows the Open/Closed Principle by extending functionality with weather-based effects without modifying the core WeaponItem class. It aligns with the Single Responsibility Principle by handling only the weapon's behaviour and delegating status effects to specialised classes. The Liskov Substitution Principle is maintained by ensuring the modified BurningStatusEffect still behaves consistently, even with the new parameters for damage and duration.