Assignment 1 Design Rationale: CL_AppliedSession10_Group1

Requirement 2 – Monster spawning, AlienBug, SuspiciousAstronaut

The goal of this requirement is to implement two new hostile creatures in our game 'static factory' while adhering closely to relevant design principles and best practices.

To implement these hostiles creatures, Alien Bug and Suspicious Astronaut have been implemented as Monster classes from assignment 1 as this parent class encompasses all necessary functionality of a hostile creature. However, since alien bug does not attack, all behaviours has been removed from the parent class. This decision reduces repetition and gives each class its own responsibility. However, statuses and behaviours must be added for each actor individually which may cause repetition such as for wander behaviour.

Another design was for Alien Bug to generate its name using a static method. This is beneficial as only the class needs to know how to generate a name for each instance.

Alien Bug has complex behaviour and it was decided that each behaviour will implement the behaviour parent class and follow a detect then execute process. The alien bug will follow a specific target using a FollowBehaviour(target) when a target is within the surroundings of the alien bug and will move in a direction using the MoveActorAction from engine. The alien bug with pick up behaviour will pick up an item on the Actor's location using a pickup action. This approach follows the single responsibility principle where the behaviours detect the appropriate situation while the actions execute the actual functionality. This also follows open-closed principles as the actor to follow can be changed using the injected status.

An alternative design for alien bug's behaviours would be to have it aware of its own location and allow it to check for items or followable players in its surrounds however this would violate the open-closed principle as if any behaviour or detection needed to be changed, the alien bug class would need to change. Additionally if this behaviour was to be repeated in a new creature, the code would need to be repeated which violates DRY.

Overall this design provides a flexible, extensible implementation of the two hostile creatures. It carefully considers the requirements within the bounds of our design principles while also laying the groundwork for future enhancements or optimizations.