REQ3: Godrick the Grafted

The diagram represents an object-oriented design for requirement 3 of assignment 3 and displays the necessary classes and their relationships with each other in order to fulfil the needs of the overall system. These new classes make use of pre-existing code from the engine and game, allowing us to easily extend our system.

This requirement was mainly focused on implementing a two-phase boss fight with Godrick the Grafted; hence, the majority of classes that have been added to the game are associated with and support the implementation of said feature.

The first notable class added to the game is the GodrickTheGrafted class itself. This class extends Enemy and implements AoECapable. Since Godrick the Grafted is himself hostile to the player and generally behaves similarly to other enemies in the game, it makes sense to allow this class to extend the Enemy abstract class and benefit from the code reuse. This complies with the DRY, open closed principle, and Liskov substitution principles, as we do not repeat code unnecessarily, extend instead of modify old code, and our GodrickTheGrafted objects are substitutable with generic Enemies. Additionally, since Godrick is capable of performing AoE attacks with both his weapons, we force this class to implement AoECapable, allowing it to benefit from the preexisting AreaAttack action system developed in Assignment 2. The code that allows Godrick to switch phases has been implemented by overriding his playTurn method, allowing us to assess his HP before running a behaviour check.

Next, we have added two new classes that represent the weapons used by Godrick the Grafted in phases 1 and 2, respectively: the AxeOfGodrick and the GraftedDragon. The AxeOfGodrick is a weapon that can be purchased from finger reader Enia and is capable of performing an AoE attack on surrounding enemies. As such, I have chosen to allow this class to extend the abstract Sword class. This allows the AxeOfGodrick to be traded as Sword extends TradableWeapon while at the same time providing it access to the SpinningAttack action, which is a generic AoE attack action used by both Grossmesser and Scimitar. This allows us to avoid creating custom classes to implement this weapon into the game and also satisfies the DRY, open closed principle, and Liskov substitution principles. An alternative approach would have been to allow the AxeOfGodrick to extend TradableWeapon directly and then create a custom Action for its AreaAttack; however, there is no real reason to do so as the majority of code from the spinning attack would be applicable to this custom class, creating unnecessary repetition. The GraftedDragon class, however, does follow this approach. The reason for this is due to the fact that the weapon is not a slashing or cutting weapon, so the message printed by SpinningAttack would not make sense. Hence, a custom AreaAttack called Flamethrower has been created to avoid this.

The next major class that has been added is the RemembranceOfTheGrafted It is an Item dropped by Godrick upon death and can be traded for either of his weapons from the trader “Finger Reader Enia” or sold to any trader for 20000 runes. Remembrance of the Grafted being an item and not a weapon led to SellingAction being changed to SellingWeaponAction and a new Action called SellingItemAction being created. Additionally, new action called PurchaseWithItemAction was created so the Remembrance of the Grafted could be traded for Godrick’s weapons, since PurchaseAction only takes an integer cost.

The last class added for this requirement is GoldenRune, which is a consumable item that gives the player a randomly generated amount of runes when consumed. This class extends a new class, ConsumableItem (which extends Item and implements Consumable interface). This adheres to SRP (the class only has one responsibility; it has all functionality required to support that), OCP (extends

and adds new functionality without modifying or breaking the original engine code), LSP (can be used wherever an Item instance would be expected) and ISP (fully implements and utilises the Consumable interface).