**Class Diagram Supporting Document**

This document is created to explain how the newly created classes are associated with other classes shown in the class diagrams.

Other details (e.g. functionality of a class, methods in a class, purpose of a class, etc) are mentioned in design rationale.

We used different colours for classes in class diagrams to distinguish between original given classes and new added classes.

Yellow box - Original class

Blue box - New class

Our team has decided to have a total of 7 class diagrams on the basis of the package. The packages are as follows-

1. Item Package
2. Action Package
3. Actors Package
4. Behavior Package
5. Ground Package
6. Weapon Package
7. WeaponSkills Package

The components of each of the packages are as follows-

1. **Item Package**

* Item
* CindersOfALord
* TokenOfSouls
* EstusFlask
* PortableItem
* RetrieveSoulAction
* DrinkEstusFlaskAction

This package has all classes in the game package that extends from Item class, while CindersOfALord, TokenOfSouls, and EstusFlask are newly created classes. There are two other classes that are associated with the classes in the item package are RetrieveSoulAction and DrinkEstusFlaskAction. Class TokenOfSouls and EstusFlask are dependent on them respectively, as they need them to carry out actions.

1. **Action Package**

* Action
* DrinkEstusFlaskAction
* ResetAction
* RetrieveSoulAction
* EstusFlask
* TokenOfSouls
* Bonfire
* ResetManager

This package has all classes in the game package that extends from Action class, while DrinkEstusFlaskAction, ResetAction, and RetrieveSoulAction are newly created classes. As mentioned above, some classes in this package are associated with Item classes. Besides that, ResetAction class is associated with class Bonfire and ResetManager. It is because we want Bonfire to have the function to reset the game. Hence, it is associated with the ResetAction class that has a ResetManager instance so that it can gain ‘Reset’ function and features.

1. **Actors Package**

* Actor
* Capability
* LordOfCinder
* Skeleton
* Undead
* Player
* Cemetery
* AttackAction
* DieByChanceAction

This package has all classes in the game package that extends from Actor class, while Skeleton class are newly created classes. Even though Undead class is already existing in the given program, it is then associated with 2 newly created classes in other packages, which are class Cemetery (from Ground package) and class DieByChanceAction (from Action package). It is linked with these 2 classes as according to the requirements, Undead objects will spawn from Cemetery, and it also has a 10% chance of dying instantly. Hence, we can say that those 2 classes are specially created for Undead class so that it can execute the mentioned added features.

1. **Behavior Package**

* Behavior
* Ground
* FollowBehaviour
* EnrageBossFollowBehavior
* BurningGround
* LordOfCinder

This package has all classes in the game package that extends from the Behavior class, while EnragedBossFollowBehavior is a newly created class. The class itself is associated with BurningGtound class and LordOfCinfer. The main responsibility of the class is to handle a part of the enraged behavior of the first boss of the game, where the surrounding tiles of the boss will deal damage to the Player by changing the tiles to BurningGround class.

1. **Ground Package**

* Ground
* Vendor
* Valley
* Cemetery
* Bonfire
* Application
* ResetAction

This package has all the classes which extend the Ground class, which are the newly created classes Cemetery, Bonfire and Vendor as well as the pre existing class Valley. The package also contains the Application and ResetAction classes. The Vendor, Valley, Cemetery and Bonfire classes are in dependency with the Application class as they need it to carry out their actions. The ResetAction class is in association with the Bonfire class as we want the bonfire to have the functionality of resetting the game.

1. **Weapon Package**

* Item
* WeaponItem
* MeleeWeapon
* Broadsword
* Yhorm’s Giant Machete
* StormRuler
* GiantAxe
* ChargeAction
* WindSlash
* SpinAttackAction

This package has all classes in the game package that extends from the preexisting MeleeWeapon class, while Broadsword, Yhorm’s Giant Machete, StormRuler, GiantAxe is the newly created classes. Some of the classes are associated with an active skill class such as ChargeAction. These classes’ main responsibility is to handle the stats and skills of each weapon respectively.

1. **WeaponSkills Package**

* Action
* ChargeAction
* WindSlashAction
* SpinAttackAction
* StormRuler
* GiantAxe

This package has all classes that are extended from the Action class, which are the newly created classes - ChargeAction, WindSlashActionAction and SpinAttackAction as well as the newly created StormRuler and GiantAxe classes which extend the MeleeWeapon class. The ChargeAction, WindSlashAction and SpinAttackAction classes extend the Action class so that they can act as active skills and be cast on command by the user. The StormRuler and GiantAxe classes extend the MeleeWeapon class so that the weapons can be held by the Player