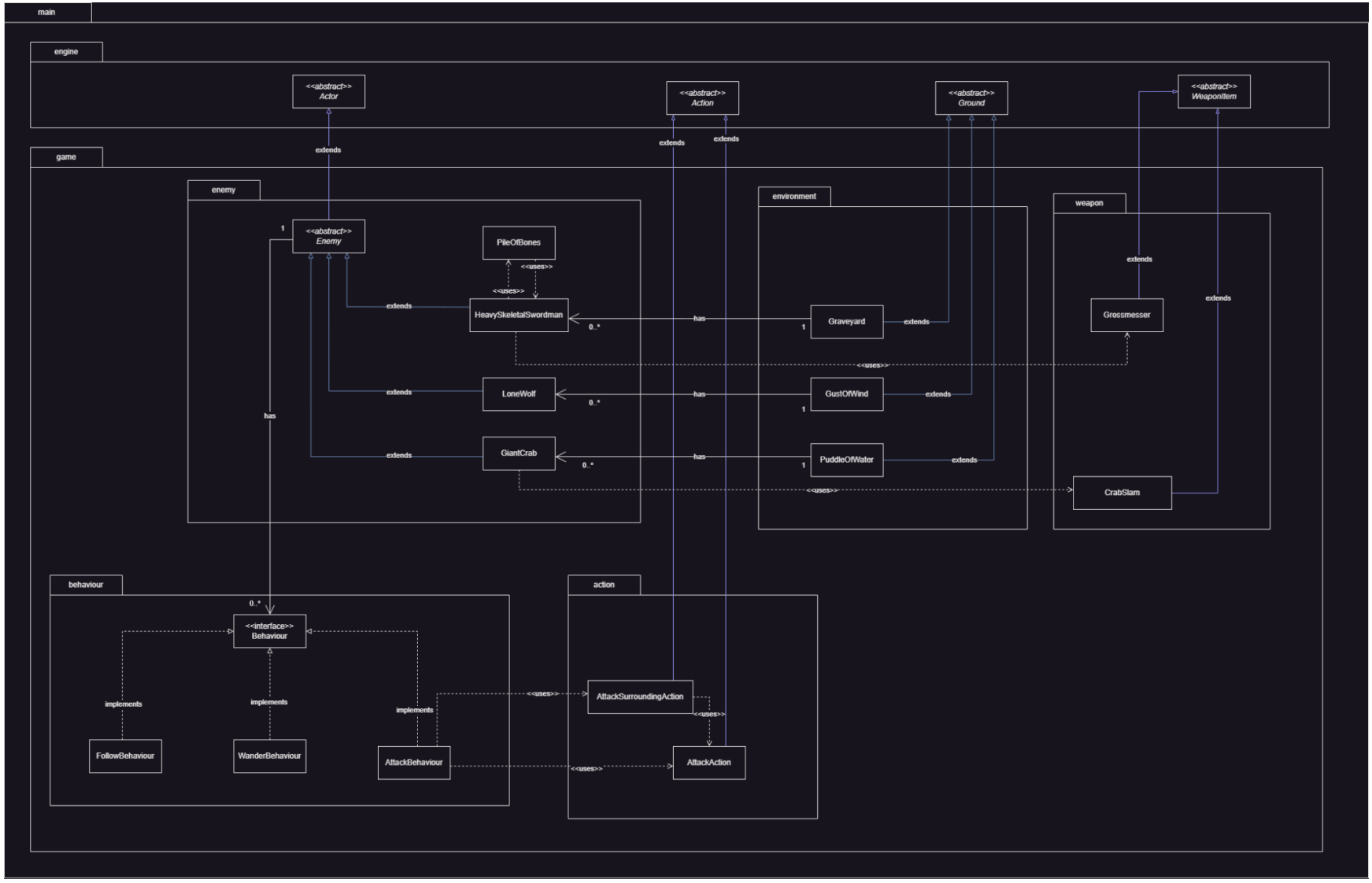
FIT 2099 Group 3
Assignment 1

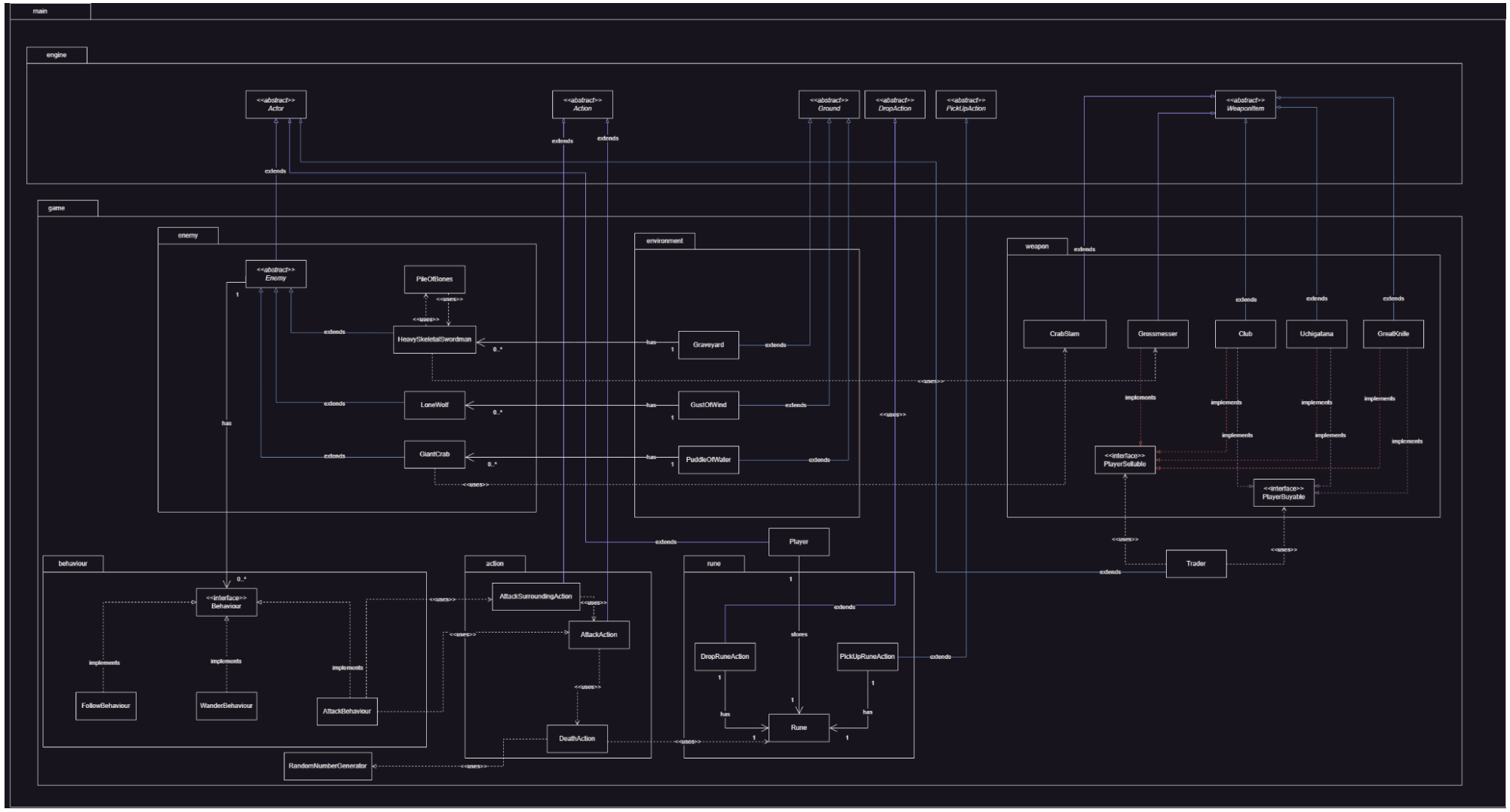
Group Member:

Lee Sing Yuan (32203632)
Loo Li Shen (32619685)
Yeoh Ming Wei (32205449)

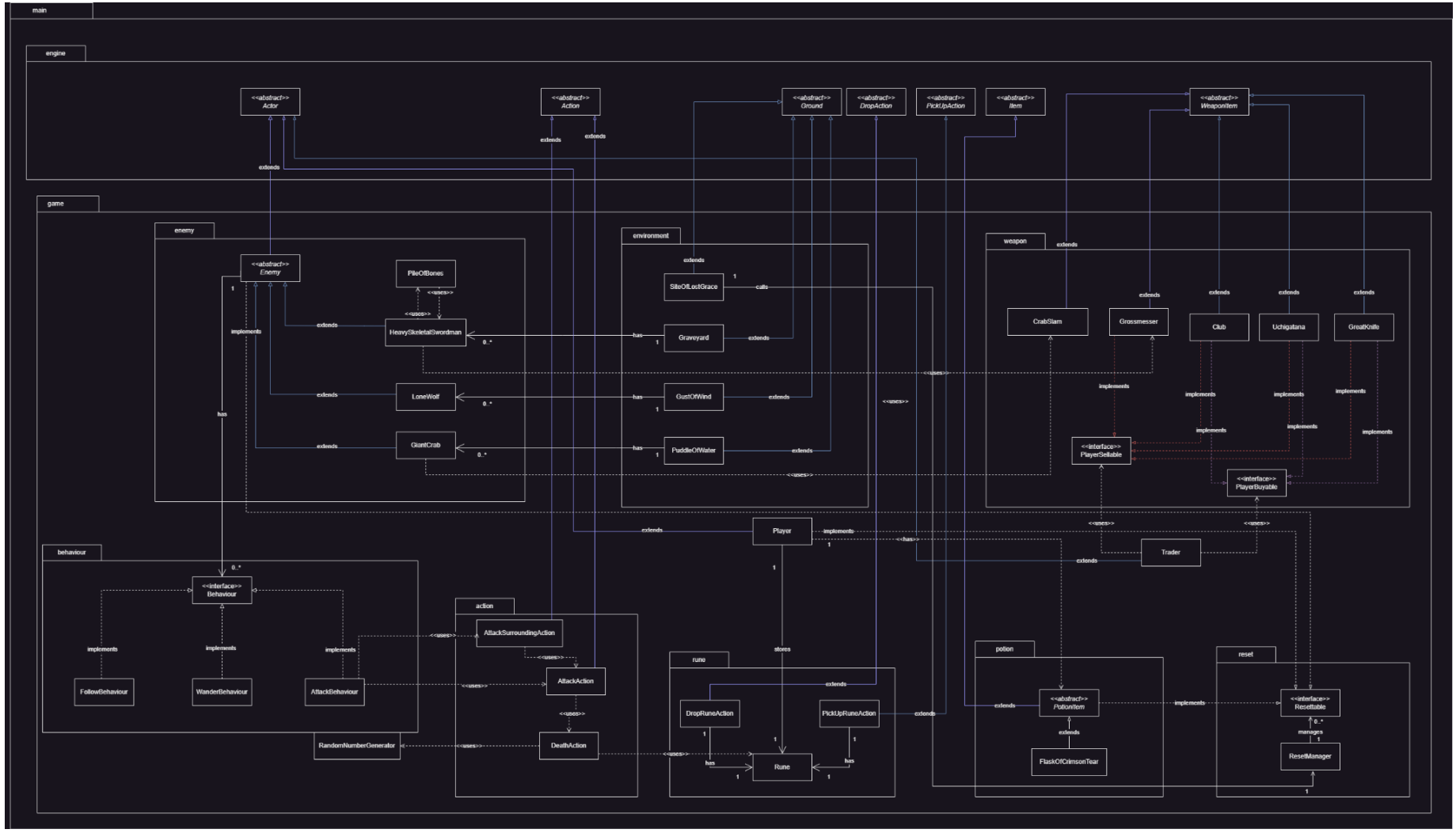
REQ 1:UML


main
engine
<<abstract>> Actor
<<abstract>> Action
<<abstract>> Ground
<<abstract>> WeaponItem
extends
extends
extends
game
enemy
1
<<abstract>> Enemy
PileOfBones
<<uses>>
<<uses>>
extends
HeavySkeletalSwordman
0..*
has
1
Graveyard
extends
<<uses>>
extends
LoneWolf
0..*
has
1
GustOfWind
extends
has
extends
GiantCrab
0..*
has
1
PuddleOfWater
extends
<<uses>>
CrabSlam
environment
weapon
extends
Grossmesser
extends
behaviour
0..*
<<interface>> Behaviour
implements
implements
implements
FollowBehaviour
WanderBehaviour
AttackBehaviour
action
<<uses>>
AttackSurroundingAction
<<uses>>
<<uses>>
AttackAction

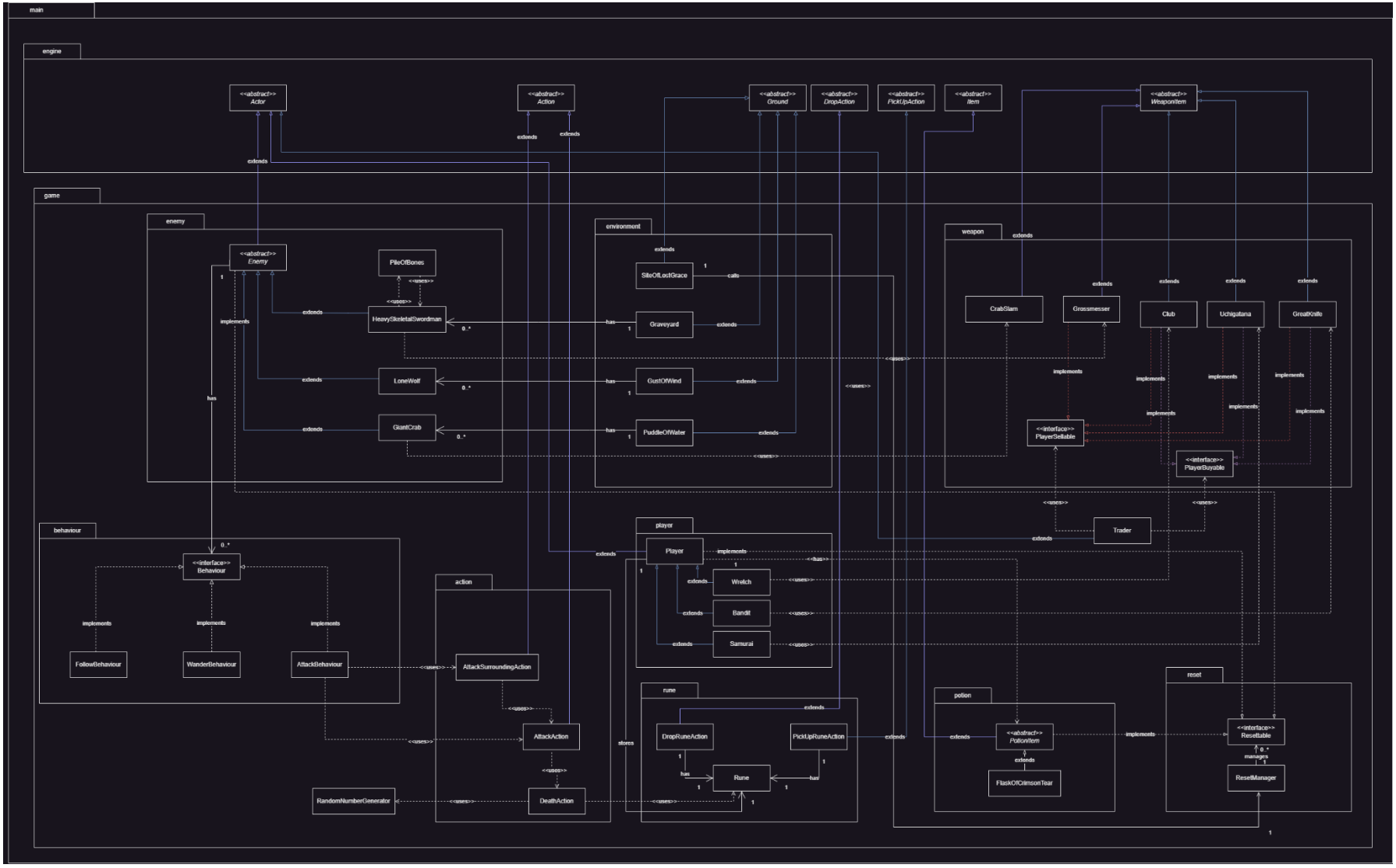
REQ 2: UML


main
engine
<<abstract>> Actor
<<abstract>> Action
<<abstract>> Ground
<<abstract>> DropAction
<<abstract>> PickUpAction
<<abstract>> WeaponItem
extends
game
enemy
<<abstract>> Enemy
PileOfBones
<<uses>>
HeavySkeletalSwordman
LoneWolf
GiantCrab
has
1
0..*
environment
Graveyard
GustOfWind
PuddleOfWater
weapon
CrabSlam
Grossmesser
Club
Uchigatana
GreatKnife
implements
<<interface>> PlayerSellable
<<interface>> PlayerBuyable
Player
Trader
behaviour
<<interface>> Behaviour
FollowBehaviour
WanderBehaviour
AttackBehaviour
action
AttackSurroundingAction
AttackAction
DeathAction
rune
DropRuneAction
PickUpRuneAction
Rune
stores
RandomNumberGenerator

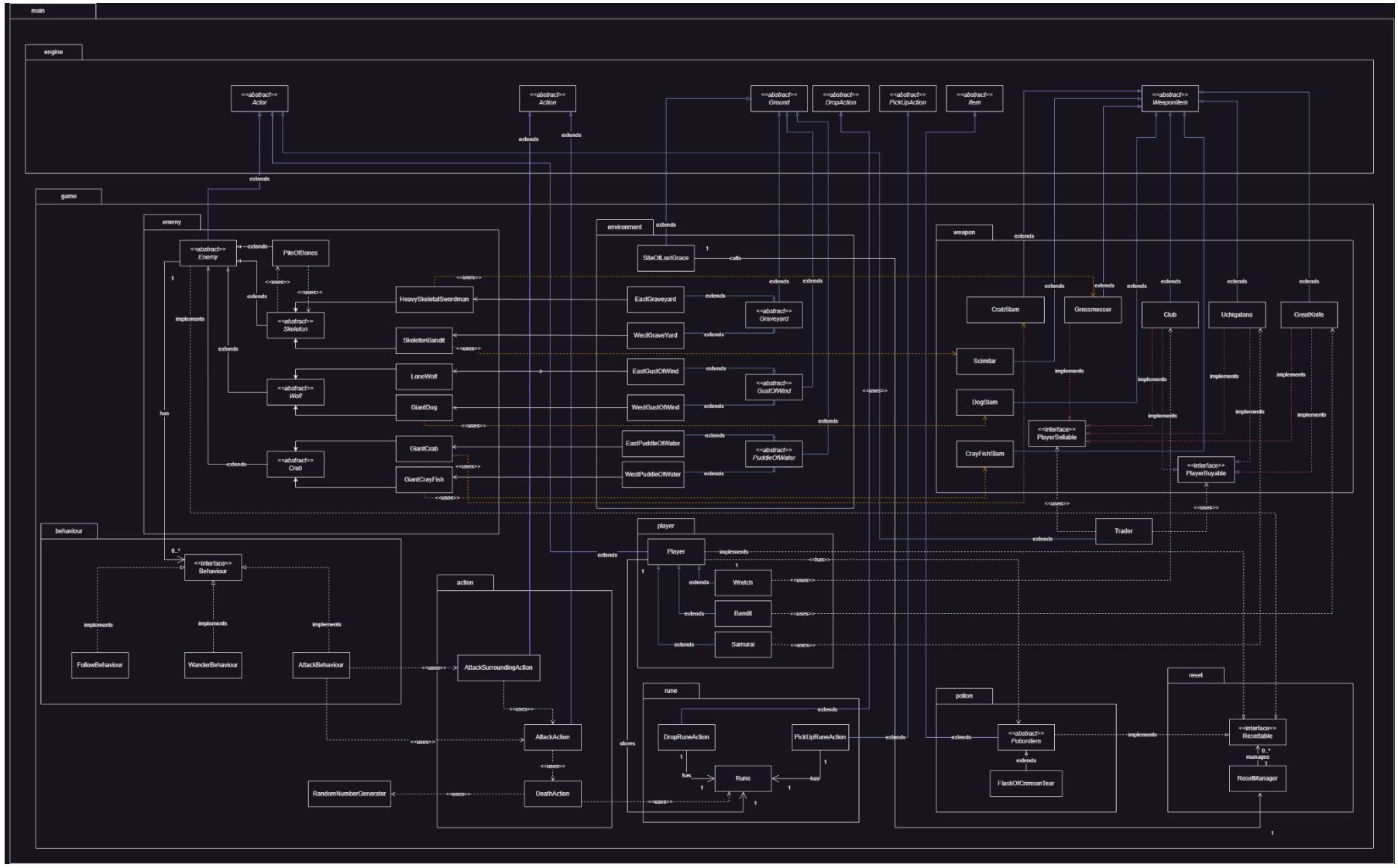
REQ 3: UML


main
engine
<<abstract>> Actor
<<abstract>> Action
<<abstract>> Ground
<<abstract>> DropAction
<<abstract>> PickUpAction
<<abstract>> Item
<<abstract>> WeaponItem
extends
game
enemy
<<abstract>> Enemy
PileOfBones
<<uses>>
HeavySkeletalSwordman
LoneWolf
GiantCrab
implements
has
0..*
environment
SiteOfLostGrace
calls
Graveyard
GustOfWind
PuddleOfWater
weapon
CrabSlam
Grossmesser
Club
Uchigatana
GreatKnife
<<interface>> PlayerSellable
<<interface>> PlayerBuyable
Player
Trader
behaviour
<<interface>> Behaviour
FollowBehaviour
WanderBehaviour
AttackBehaviour
RandomNumberGenerator
action
AttackSurroundingAction
AttackAction
DeathAction
rune
DropRuneAction
PickUpRuneAction
Rune
stores
potion
<<abstract>> PotionItem
FlaskOfCrimsonTear
reset
<<interface>> Resettable
ResetManager
manages

REQ 4: UML


main
engine
<<abstract>> Actor
<<abstract>> Action
<<abstract>> Ground
<<abstract>> DropAction
<<abstract>> PickUpAction
<<abstract>> Item
<<abstract>> WeaponItem
extends
game
enemy
<<abstract>> Enemy
PileOfBones
HeavySkeletalSwordman
LoneWolf
GiantCrab
implements
has
<<uses>>
environment
SiteOfLostGrace
Graveyard
GustOfWind
PuddleOfWater
calls
weapon
CrabSlam
Grossmesser
Club
Uchigatana
GreatKnife
<<interface>> PlayerSellable
<<interface>> PlayerBuyable
Trader
behaviour
<<interface>> Behaviour
FollowBehaviour
WanderBehaviour
AttackBehaviour
action
AttackSurroundingAction
AttackAction
DeathAction
RandomNumberGenerator
player
Player
Wretch
Bandit
Samurai
rune
DropRuneAction
PickUpRuneAction
Rune
stores
potion
<<abstract>> PotionItem
FlaskOfCrimsonTear
reset
<<interface>> Resettable
ResetManager
manages

REQ 5: UML


main
engine
<<abstract>> Actor
<<abstract>> Action
<<abstract>> Ground
<<abstract>> DropAction
<<abstract>> PickUpAction
<<abstract>> Item
<<abstract>> WeaponItem
extends
game
enemy
<<abstract>> Enemy
PileOfBones
<<abstract>> Skeleton
HeavySkeletalSwordman
SkeletonBandit
<<abstract>> Wolf
LoneWolf
GiantDog
<<abstract>> Crab
GiantCrab
GiantCrayFish
implements
has
<<uses>>
environment
SiteOfLostGrace
calls
EastGraveyard
WestGraveYard
<<abstract>> Graveyard
EastGustOfWind
WestGustOfWind
<<abstract>> GustOfWind
EastPuddleOfWater
WestPuddleOfWater
<<abstract>> PuddleOfWater
weapon
CrabSlam
Grossmesser
Club
Uchigatana
GreatKnife
Scimitar
DogSlam
CrayFishSlam
<<interface>> PlayerSellable
<<interface>> PlayerBuyable
Trader
behaviour
<<interface>> Behaviour
FollowBehaviour
WanderBehaviour
AttackBehaviour
action
AttackSurroundingAction
AttackAction
DeathAction
RandomNumberGenerator
player
Player
Wretch
Bandit
Samurai
rune
DropRuneAction
PickUpRuneAction
Rune
stores
potion
<<abstract>> PotionItem
FlaskOfCrimsonTear
reset
<<interface>> Resettable
ResetManager
manages

Rationale

To improve the modularity and maintainability of the code, we have applied several principles in our design. Firstly, to avoid code duplication, we have created a parent class for the similar ground types, such as the graveyard, puddle, and wind **(DRY)**. Additionally, we have implemented the **open-close principle** by making it easy to add new types of skeletons. For example, when adding a new type of skeleton, we only need to make it inherit from the Skeleton class.

Furthermore, we have applied the **single responsibility principle** by having the enemy class handle only the existence of the enemy, while the skill,which is assumed that the enemies will have a weapon for that skill in their inventory, is handled through an action class by checking if the required weapon exists.

Moreover, for the skeleton skill, it will be assumed that it is another enemy. So when the skeleton dies, it creates this other enemy to replace it and when the count is completed,it creates a skeleton again. It is done this way, because for future enemies that have the same skill (not limited to skeleton) they just have to instantiate this class without touching any other already existing code. **(Open Close Principle)**

The trader uses the interfaces to handle inventory and selling of weapons. This reduces the number of dependencies to the trader and if there are new weapons added in the future, they have to implement the interface without any changes done to the trader. **(Open Close Principle)**. For the weapons, they have many similar properties. Hence it inherits from an abstract class **(DRY).**

For REQ3, players will drop all their runes after being killed by an enemy and also be able to collect the runes as runes will drop on the ground after the game resets. DropRuneAction class and CollectRuneAction class will be separated methods in which there is only one method for each of the classes and has only one job to do. It allows us to easily manage the classes when there are problems in a specific action class. **(Single Responsibility Principle).** Since DropRuneAction class is similar to DropAction and CollectRuneAction is similar to PickUpAction, it is better to perform an extension to their respective abstract classes so that we do not modify the abstract DropAction and PickUpAction classes and instead only make changes at CollectRuneAction and DropRuneAction. **(Open-Close Principle)**

For the Flask of Crimson Tears, it inherits a potion abstract class so that if in the future, any new potions are added, they just have to inherit from the potion class without changing any code **(Open Close Principle)**

For the Site of Lost Grace, it inherits ground abstract class. The SiteOfLostGrace class should have only one responsibility, which is to manage the player's progress by resetting and allowing them to respawn at the last visited site. The "SiteOfLostGrace" class will call upon the "ResetManager" when the player dies or rests at a site of lost grace. The "ResetManager" class is responsible for managing the reset functionality of the game, while the "Enemy", "Player", and "PotionItem" classes implement the "resettable" interface, which has a single responsibility of providing the reset functionality. By separating the reset functionality into its own class and interface, the design follows the **SRP**, which helps to make the code more maintainable, testable, and flexible.

Upon resetting, the reset manager calls all who implemented the resettable interface. This allows addition of new resettable entities without modifying any existing code **(Open Close Principle).**

The roles or classes for the player, is done by an abstract player class because of the similarities. **(DRY)**