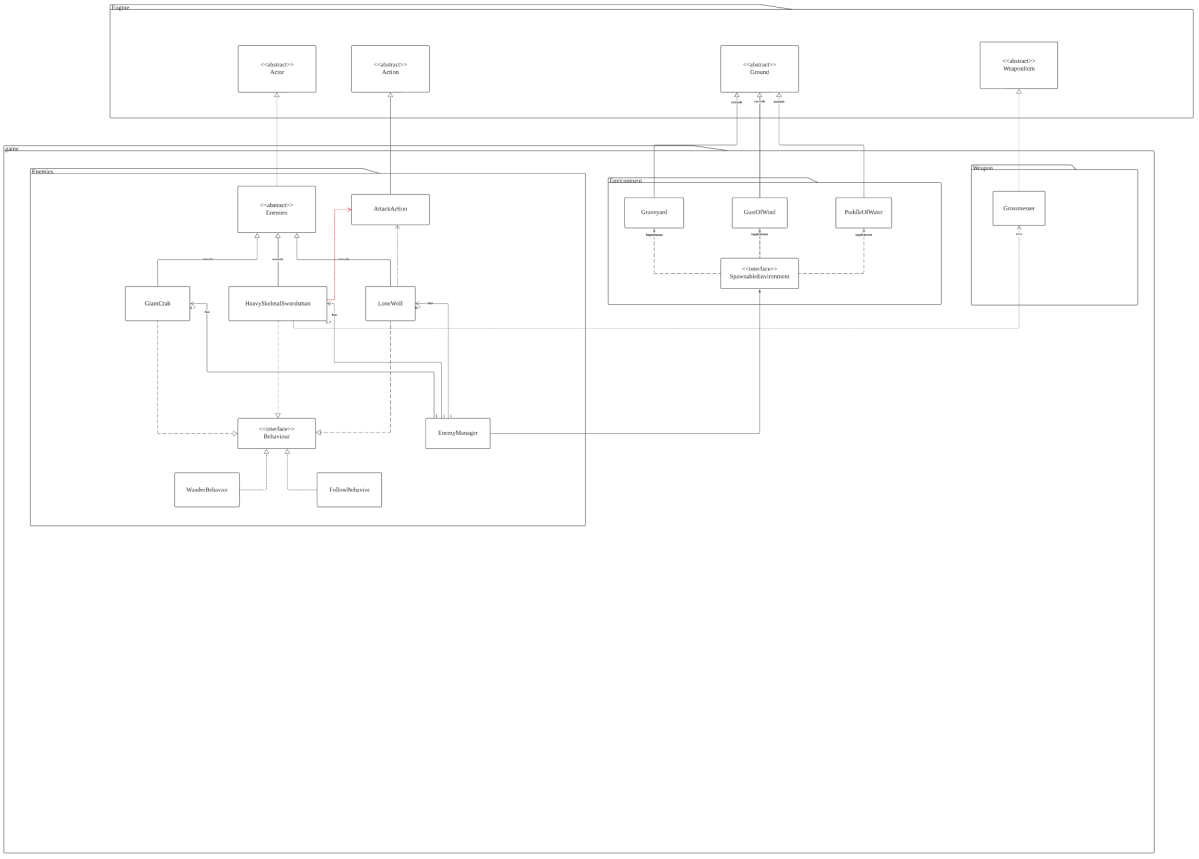
FIT 2099 group 3 assignment 1

Lee Sing Yuan
Li Shen
Yeoh Ming Wei

REQ 1:UML


Engine
<<abstract>>
Actor
<<abstract>>
Action
<<abstract>>
Ground
<<abstract>>
WeaponItem
game
Enemies
<<abstract>>
Enemies
AttackAction
GiantCrab
HeavySkeletalSwordsman
LoneWolf
<<interface>>
Behaviour
EnemyManager
WanderBehavior
FollowBehavior
Environment
Graveyard
GustOfWind
PuddleOfWater
<<interface>>
SpawnableEnvironment
Weapon
Grossmesser

Design Rationale

To help us understand how your system will work, you must also write a *design rationale* to explain your choices. You must demonstrate how your proposed system will work and ***why* you chose to do it that way**. Here is where you should write down concepts and theories you've learnt so far (e.g., DRY, coupling and cohesion, etc.). You may consider using **the pros and cons** of the design to justify your argument.

The design (which includes *all* the diagrams and text that you create) must clearly show the following:

- what classes will exist in your extended system
- what the roles and responsibilities of any new or significantly modified classes are
- how these classes relate to and interact with the existing system
- how the (existing and new) classes will interact to deliver the required functionality

You are not required to create new documentation for components of the existing system that you *do not* plan to change.

Goal:
How was it implemented:
Why this method?:
Alternative:
Pro:
Con:

REQ 1:Rationale

1.A) interface for spawnable environments

Goal: Reduce amount of inheritance
How was it implemented: interface for environment spawning enemies
Why this method?: 1) Because there already exist a ground parent class
2) Could have more environments that could spawn enemies
Alternative: creating another parent class
Pro: applies DRY
Con: may be hard to debug when we got issue

1.B) enemy manager

Goal: Reducing dependency
How was it implemented: EnemyManager class ,handles arraylist of enemies, spawn and despawns. (for example 1 spawn method and then if conditions to select which enemy to spawn or despawn rather then interface [alternative method])
Why this method?: Because with this it will be easier to spawn and despawn enemies in the future.
Alternative: 1 enemy connect to one environment (something like interface to spawn enemies)
Pro: more direct and easier to debug
Con: hard to manage all enemies since they are all in different classes , a lot of code repetition (for example : in Graveyard , need to have/ implement method which is quite similar to other environments like Puddle but with minor tweeks. Breaks DRY)

1.B.1 and 1.B.3) skeleton skill and crab attack area is coded in class

Goal: achieve DRY
How was it implemented: code the skill in the class and if there is a need later like in **REQ 5**, make this class a parent
Why this method?: Because the skills are unique to them and for example in **REQ 5** the code is exactly the same with no changes (based on the specs at the current time)
Alternative: interface for the skills
Pro: easy to implement (copy paste)
Con: repeated code , breaks DRY

1.C) Grossmaseer inherits weaponItem