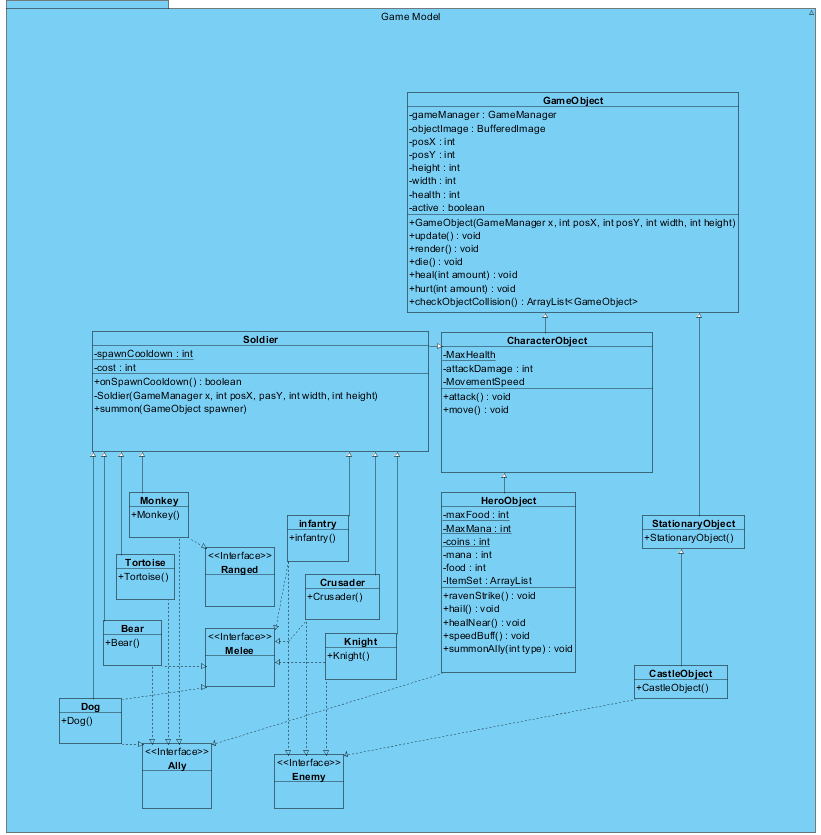
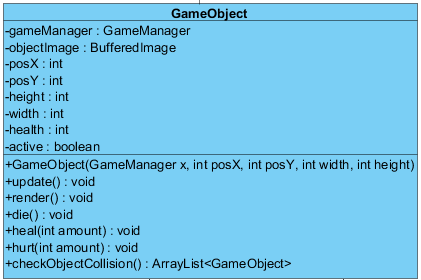
# **3. Subsystem Services**

**Game Model Subsystem**

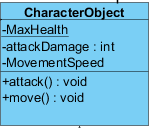
“Game Model Subsystem” describes the entities that are used while the game is running. It has describes many crucial objects including all the enemy and ally soldiers, the hero itself and the enemy castle. It also has interfaces such as “Ally”, “Enemy”, “Melee” and “Ranged”.



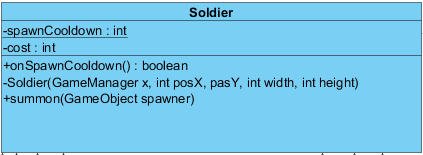
GameObject Class:

“GameObject” is an abstract class that is the parent of all game objects including HeroObject, all Soldier objects and CastleObject. This class has attributes and methods that all entities should have such as position information declared as “posX” and “posY”, image of the entity declared as “objectImage”, width and height of the object and so on. Operations such as “update()”, “render()”, “die()” and “hurt()” are defined as abstract operations since different types of objects might need to act differently to these operations. The “render()” and “update()” methods are periodically called by the ObjectManager on active GameObjects.

CharacterObject Class:

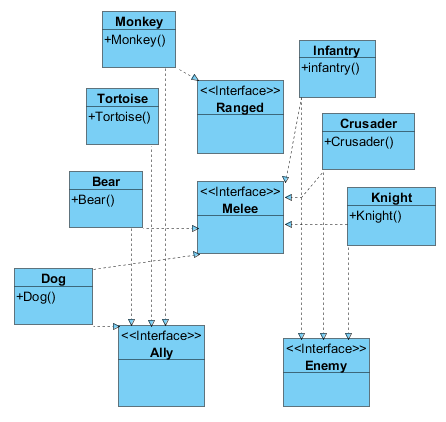
 “CharacterObject” is also an abstract class that is the parent of all character objects like enemy and ally soldiers. The “attack()” and “move()” operations are defined as abstract operations since every character can have different attack and move values/patterns.

Soldier Class :

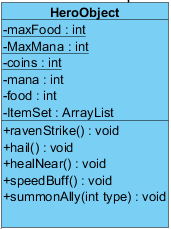


“SoldierClass” is yet another abstract class that has the common attributes of all the soldier type objects. They have a private constructor and they are intended to be constructed only by the “summon()” operation given the conditions are met.

Soldier Classes and Interfaces:

Classes like “Monkey”, “Tortoise”, “Bear” and “Infantry” are all child classes of the “Soldier” abstract class. Their hostility is indicated using the “Ally” or “Enemy” interfaces. And their attack type is indicated by the “Melee” and “Ranged” interfaces. Note that Tortoise class implements neither the “Ranged” or the “Melee” interface, since it is a defensive soldier that does not attack.

HeroObject Class:



“HeroObject” class defines the character which the player controls. The “coins”, “mana” and “food” values are also held in this object.“ravenStrike()”, “hail()” , “healNear()” and “speedBuff()” methods all correspond to skills the player can cast using mana. The player can also summon an ally soldier using food, thanks to the “summonAlly()” method.