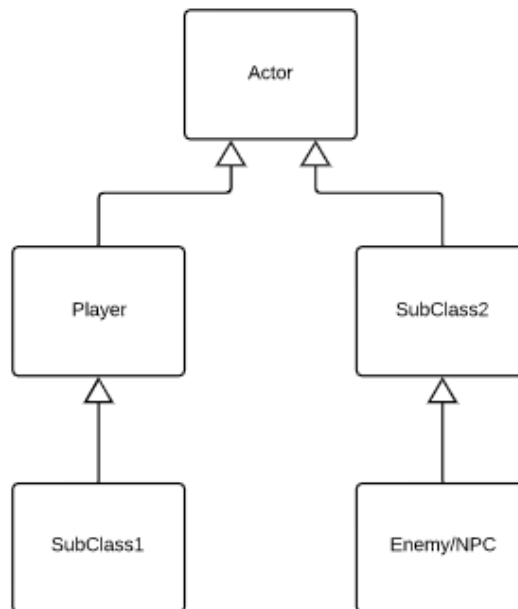


Recommendations for change to the game engine

Perceived problem 1

Repeating code when trying to find a specific item from actor's inventory. Was thinking to create a subclass to resolve this problem, however Player is already inherited from Actor. Which means we would need to create 2 subclasses, one for Player, another one for Actor, to resolve this problem with a repeated modification (we could not change the code in game engine). Below is a visualization of such solution without modifying the game engine.



Proposal on design changes

Add a new method for Actor class which is to check whether a specific item is inside of actor's inventory. This method would take one parameter, which is the name of item (String) and would return the first occurrence of item object with the same name if it exists in actor's inventory else it would return null.

Advantages of proposal

It follows the principle we have learnt in this unit, 'Don't Repeat Yourself'. Which would make our life easier and have a cleaner and more readable code. The proposed method returns an item reference from the actor inventory, so when we would like to remove an item from actor's inventory, we do not have to recode to find the item itself.

Example:

```
Item key = actor.contains("Key");  
If (key != null) {  
    actor.removeItemFromInventory(item);  
}
```

Disadvantages of proposal

Using the name of item (String) as the parameter of the proposed method "smells" like it has a "code smell" because there could be a scenario where items with different name.'

Perceived problem 2

Must create a new class for enemy with a damage that is base on the basic enemy (Grunt). Such as Goon, Yugo Maxx and Doctor Maybe.

Proposal on design changes

Create a public static final attribute inside of Actor class for storing the base damage (intrinsic weapon) or base hit points (HP). Have tried to record the base hit points inside of our Enemy class and it succeed. However, when I was trying do the same thing to base damage, it failed due to you can not store the return's value of method inside of public static final attributes.

Example:

```
public static final int BASE_DAMAGE = super.getIntrinsicWeapon();
```

Advantages of proposal

Easier to retrieve the value when defining a new value which is based on it.
Reduced the number of classes whilst increase the variety of enemy.

Disadvantages of proposal

If there is enemy with different hit points/damage that has different behavior or functionality, it would not reduce the number of class that need to be created as we would still need to override the `getAllowableActions()` or other parent's method.