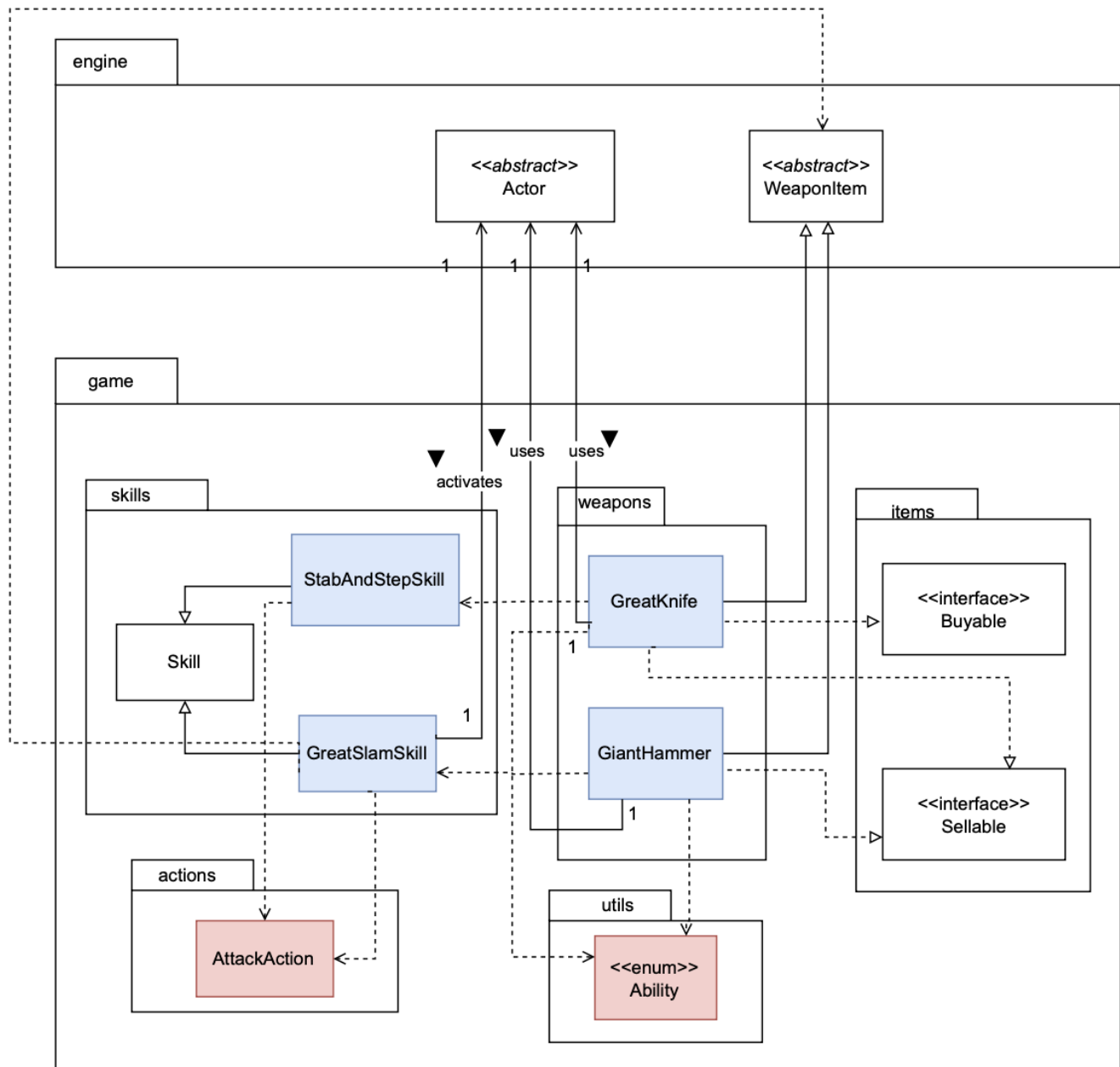


Design Rationale (REQ 4)



Changes Made from Assignment 1

No major changes were made from Assignment 1. All the code used follows the design principles, and expandable for the requirements in Assignment 2.

Implementation

Classes Added

1. Class GreatKnife extends WeaponItem
2. Class GiantHammer extends WeaponItem
3. Class StabAndStep extends Skills

4. Class GreatSlam extends Skills

Design Principles

Class GreatKnife, GreatHammer

These classes was made to represent particular weapon items (GreatKnife, GreatHammer).

Design Principles Followed

1. Single Responsibility Principle (SOLID)

- Both the GreatKnife and GreatHammer classes represent specific types of weapon item. They encapsulate specific functionalities of these weapons, such as their when to create attack action, its skill action, and its buying/selling prices if applicable.

2. Open-Closed Principle (SOLID)

- Both classes are open for extension (can be subclassed to create variations) but closed for modification. The actions provided to the player at different circumstances can be extended by overriding their methods, but its structure remains the same.

3. Liskov Substitution Principle (SOLID)

- The GreatKnife and GreatHammer classes extend the WeaponItem class, therefore they are substitutable with their super class. Their overridden methods are consistent with the behavior expected from the WeaponItem class.

4. High Cohesion

- High cohesion is maintained in both classes. All methods and properties within each class revolve around a single concept: defining the actions provided to the player and storing/returning their attributes related to transactions with traders.

Class StabAndStepSkill, GreatSlamSkill

These classes was made to represent particular weapon skills.

Design Principles Followed

1. Single Responsibility Principle (SOLID)

- Both the StabAndStepSkill and GreatSlamSkill classes represent specific skills performed by the player with a weapon. They encapsulate specific events upon the execution of these skills.

2. Open-Closed Principle (SOLID)

- Both classes are open for extension (are subclassed to create variations) but closed for modification. The implementation of each skill did not require any modifications within its super class, but was achieved solely by method overrides.

3. Liskov Substitution Principle (SOLID)

- The StabAndStepSkill and GreatSlamSkill classes extend the WeaponItem class, therefore they are substitutable with their super class. Their overridden methods are consistent with the behavior expected from the WeaponItem class.

4. High Cohesion

- High cohesion is maintained in both classes. All methods and properties within each class define their display description and the events to be executed upon the use of skills.

Pros of the Design

1. Extensibility

- The skills added (StabAndStep and GreatSlam) operate completely independent from the weapons. This means that any weapons can have these functionalities by adding the skill instance to the allowable actions, without having to make changes in either classes.