



### Changes from Assignment 1

We decided to remove the abstract 'SkilledWeapon' class entirely, all weapons directly extend 'WeaponItem' now. Originally, this class was implemented in Assignment 1 under the assumption that all weapons with special skills would have a limited turn duration. But with the specifications of REQ4, we realise that perhaps 'FocusSkill' is the exception rather than the rule. As such, we decided to change the implementation of how 'FocusSkill' is implemented.

Now, we have a 'FocusActionCapable' interface instead, this interface has 'setSkillTurn' and other methods to provide the functionality of buffing the Actor activating the skill for a limited amount of turns. This interface is implemented by BroadSword.

'FocusAction' constructor now takes in a 'FocusActionCapable' instance instead, and calls the methods defined in the interface inside the 'execute' method. This adheres to the Dependency Inversion Principle (DIP) as 'FocusAction' depends on the abstraction of the 'FocusActionCapable' rather than its detailed implementation.

With these changes, we adhere to the Interface Segregation Principle (ISP), as only BroadSword with its Special Focus skill needs to implement the 'FocusActionCapable' interface. Furthermore, this implementation adheres to the Open/Closed Principle (OCP) as we do not need to modify the existing code when adding more weapons with the special Focus skill.

The main objective for REQ4 is to add 2 new weapons: GreatKnife and GiantHammer, which have the special skills of 'StabAndStep' and 'GreatSlam' respectively. GreatKnife can be bought and sold, while GiantHammer can only be sold. Also, we have to add the boss map of Abxervyer.

To achieve this, we just have 'GreatKnife' and 'GiantHammer' extend WeaponItem. GreatKnife returns 'StabAndStepAction' in its AllowableActions, while GiantHammer returns 'GreatSlam' in its AllowableActions. Both weapons return 'SellAction' in their AllowableActions, and the logic of buying and selling is the same as explained in the

previous requirement.

## **Advantages**

### **Simplicity and Clarity**

Our new implementation of having all weapons extend `WeaponItem` directly is more straightforward. The previous implementation added another layer of abstraction of 'SkilledWeapon', which only existed to provide the functionality of keeping track of the duration of turns, that is not used by other weapons other than `BroadSword`.

### **Adherence to ISP**

This new implementation adheres to Interface Segregation Principle (ISP) as we do not force other weapon classes like `GreatKnife` and `GiantHammer` to depend on methods or functionality they do not use.

## **Disadvantages**

### **Increased Dependency Coupling**

We introduced a new level of dependency coupling by having 'FocusAction' depend on the 'FocusActionCapable' interface. If there are changes to the interface in the future, all implementing classes, including 'BroadSword,' may need to be updated accordingly.