

### Documentations (Bonus Features: GameShop & Gauntlet)

In the game, there is a GameShop on Earth which sells HP potions and Gauntlets. The functionality of the items sold are as follows: -

Items	Description
HP Potion	Allows player to heal itself by different healpoints
Gauntlet	Allows player to snap finger to cause the enemies within 5 squares apart from it to disappear

The items at the GameShop can be bought using different amount of coins. Each enemy will drop a small pouch of coins that contains different amount of coins after being defeated. The player is given an empty coin pouch at the start of the game. The GameShop seller can only look at the coins in player's coin pouch. This means that the player have to add the coins dropped into its coin pouch. If player do not have the coin pouch with it (e.g. player dropped the coin pouch), it can still pick up the coins, then add it to the pouch later. Player can only buy one item each turn.

#### *Coins and coin pouch*

A CoinPouch class is created to implement coin pouch object. It inherits Item class and has an instance variable of type integer to keep track of the amount of coins in the pouch. It also contains public methods that can return the amount of coins, add coins and withdraw coins.

In GameManagement, all enemies are given a small pouch of coins to be dropped after they are defeated. A small pouch of coins is an inventory Item object that is added with AddCoinsAction in its allowableActions. YugoMaxx's small pouch contains 10 coins, DrMaybe's contains 5 and other enemies' contain 3 coins. The amount of coins in each small pouch is specified when constructing the AddCoinsAction. When player performs AddCoinsAction on a small pouch of coins, all the coins in the small pouch will be transferred to player's pouch, then the empty small pouch will disappear. If player has no coin pouch, player will fail to do so.

### *GameShop*

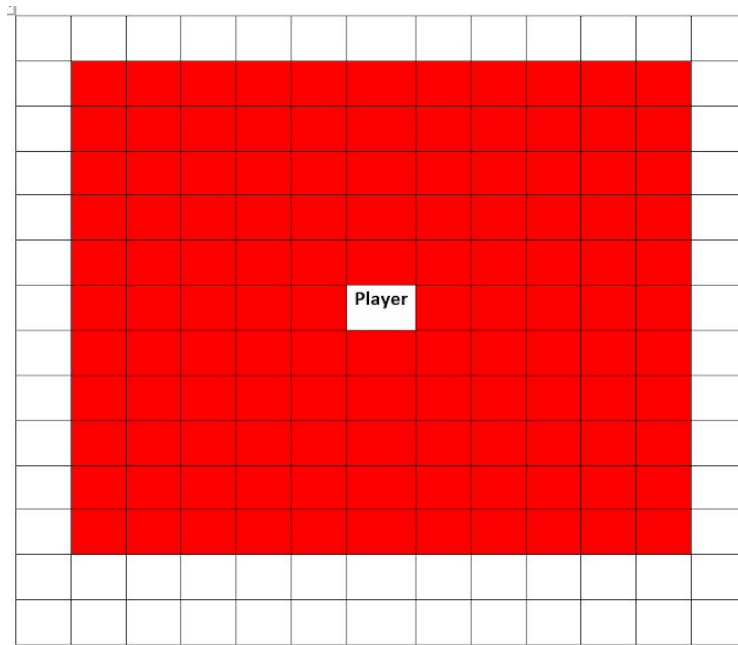
A GameShop is a furniture Item object that allows Actors to perform VisitShopAction. In VisitShopAction's constructor, a hashmap that maps Item to Integer will be added with Item objects to be sold as the keys, and the amount of coins required to buy the item as the associated data. When player performs this action, a menu will be printed. The menu shows the total amount of coins in player's coin pouch and the list of items sold with the required amount of coins. Player will have to enter a valid character key to buy the item. Player also has a option to leave the GameShop without buying anything. If player choose to buy an item, VisitShopAction will first check if player contains enough coins to buy the item, then only deduct the amount from player's coin pouch and add the item to its inventory. Once the item is sold, the item key will be removed from the hashmap that maps item to cost. This means that the GameShop will run out of stock if all items are bought.

### *Potion*

Potion is an Item object which is created and added into the hashmap in VisitShopAction that maps the potion item to its price. The price for every potion varies based on the potion's heal point. Each potion allows the player having it to perform UsePotionAction to restore the player's hitpoint by the potion's healpoints. UsePotionAction has instance variable of type Item to store the reference to the potion that this action uses, it also has an instance variable of type integer that records the potion's heal points. During the execution of UsePotionAction, it will restore the player's hitpoint by the potion's heal point and remove this potion from player's inventory. The player's hitpoints will never exceeds its maximum hitpoints after using the potion.

### *Gauntlet*

Gauntlet is also an Item object which is created and added into the hashmap in VisitShopAction that maps the potion item to its price. The price for one gauntlet is 25 coins and it allows player to perform SnapFingerAction. During the SnapFingerAction's execution, it will search for all the locations that are within 5 squares from the player (See example in diagram below) and check if the actor at any of those locations (if any) is an enemy, if it is an enemy, the enemy will be knocked out and removed from the map. The gauntlet will then disappear from player's inventory at the end of the execution of this action.



For example, when the player in the diagram above performs SnapFingerAction using the gauntlet, all the enemies that are in the red zone will be knocked out and removed from the map.

## Class Diagram

The interactions between new classes added for bonus feature and existing classes are shown below:

