Dax Collins Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

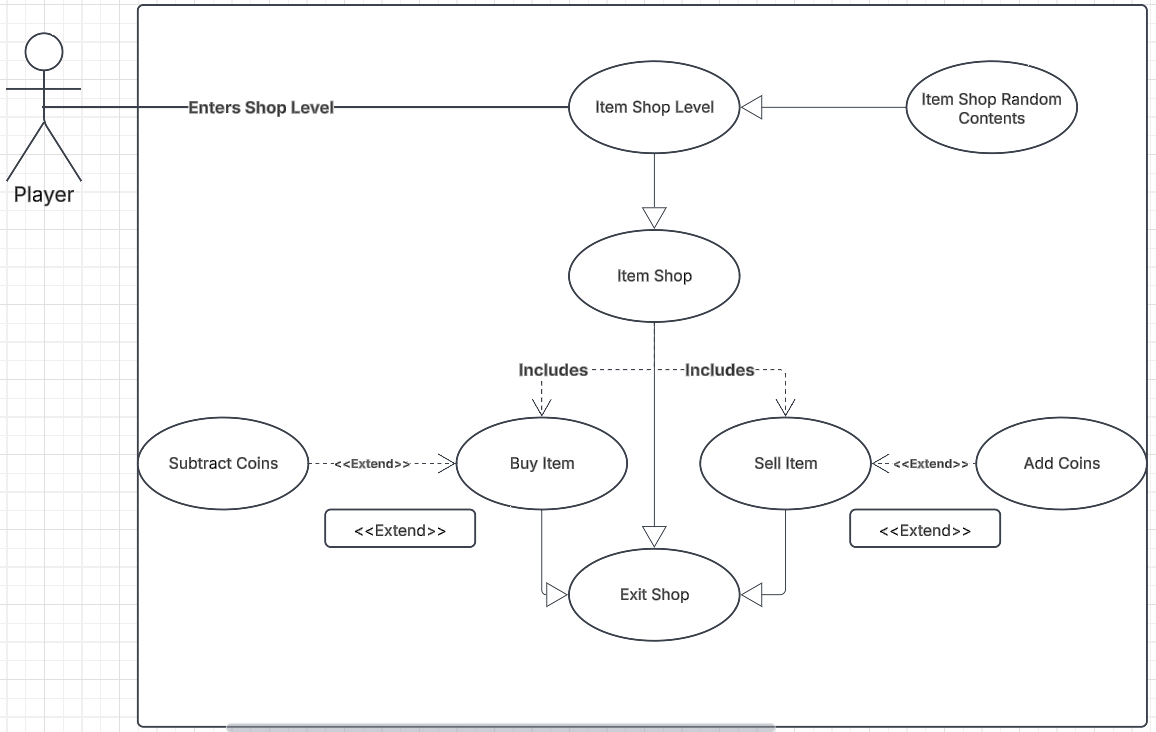
My feature for InBetween is an item shop system, the level itself, and the items available at the shop.

The player will work to collect coins through defeating enemies, opening chests, and selling items. After collecting coins over the course of the game, the player will be able to purchase items such as weapons, armor, potions, and artifacts from the item shop. The store itself is its own level that spawns after every five completed levels and a final spawn before the last boss level. Additionally, there will be a pool of items that can spawn in the item shop and the ones available for purchase will be randomly selected. If an item has already been purchased by the player, it will not show up in the shop again. Furthermore, every time the player visits the shop level, the items available will be different from the last. The Item shop will have a spot for armor, weapons, potions, and artifacts with a certain number of each item purchasable. The items for sale will have set prices. Moreover, there will also be a spot for the player to sell items that they want to exchange for a small number of coins.

Lastly, all of the items within the shop, their sprites, and possible animations will also be a part of my feature.

## Use case diagram with scenario \_\_14

### Use Case Diagram



### Scenarios

**Name:** Shop System

**Summary:** The player collects coins to purchase items such as weapons, potions, and armor from the randomly generated item shop.

**Actors:** Player

**Preconditions:** Player has made it to the shop level.

**Basic sequence:**

**Step 1:** User Enters Shop Level

**Step 2:** User Accesses Item Shop

**Step 3:** User Exits Item Shop

**Exceptions:**

**Step 2.1:** User Buys an Item

**Step 2.2:** User sells an Item

**Post conditions:** Coins are added or subtracted to the player’s total count.

**Priority:** 2\*

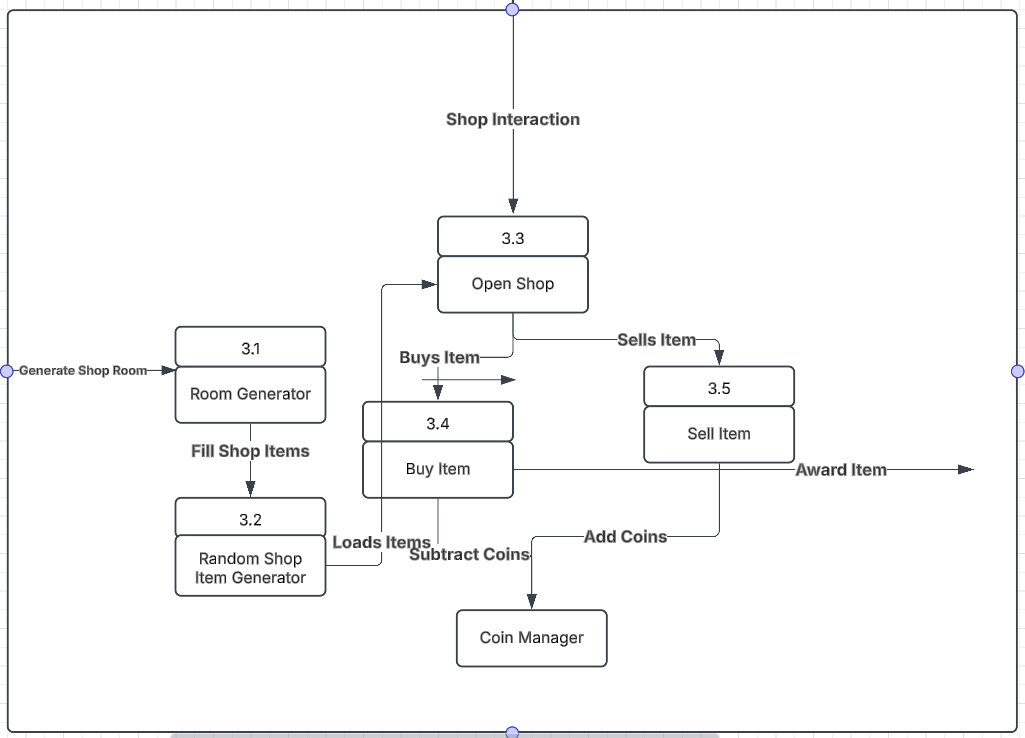
**ID:** C03

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

A diagram of a game

Description automatically generated

### Level 1 Diagram, Section 3: Item Shop



### Process Descriptions

**Room\_Generator ()**

Instantiate (ItemShop)

**Random\_Item\_Generator ()**

for( Int i = 0; i < 10; i++ )

item\_i = random\_Item ()

instantiate item

set item price = Item\_Set\_Value

**Random\_Item ()**

Int x = rand()

Return (Item ID = x)

**Buy\_Item ()**

OnButtonPress purchase item

CoinManager(item\_Set\_Value)

Inventory.AddItemPurchased

**Sell\_Item ()**

OnButtonPress Sell Item

CoinManager(item\_Sell\_Value)

**CoinManager ()**

Enemy\_Kill += coin\_Amount

Purchase\_Item -= coin\_Amount

Sell\_Item -= coin\_Amount

## Acceptance Tests \_\_\_\_\_\_\_\_9

Testing the Item Shop works to ensure that the level itself loads properly, random items are loaded, items purchased or sold add or subtract the right number of coins, and items are properly added or removed from the player’s inventory. To test everything in the level, there are a number of tests to run. Firstly, the player needs to enter the level to test that it loads properly. Following, the player needs to open the shop multiple times to check if items are loading properly. Next, the player needs to purchase every item in the shop to ensure the proper number of coins are subtracted, that the player cannot go into negative coin amounts, and that items are put in the player’s inventory. Lastly, the player needs to sell every item to test that proper coins are given back to the player and that items are removed from the player’s inventory.

**Input/output examples**

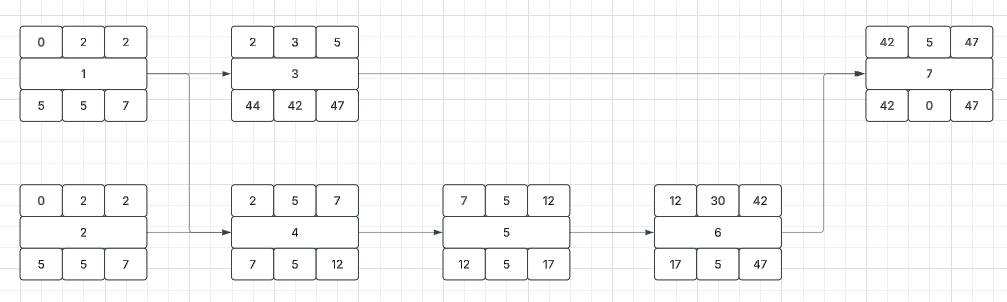
|  |  |  |
| --- | --- | --- |
| Input | Output | Notes |
| Player Enters Item Shop Level | Item shop level loads and everything is in place | Level should be fully loaded with no missing features |
| Player Opens Item Shop | Items within the shop are filled with random items from the item database | Items should only be loaded one time – no duplicate items, items should be different from the previous item shop level, and items should have the correct values |
| Player Buys an Item | Item is added to player’s inventory and coins subtracted from total coins | The correct item should be added to the player’s inventory and the correct number of coins should be subtracted |
| Player Sells an Item | Item is removed from the player’s inventory and coins added to total coin amount. | The correct item should be removed from the player’s inventory and the correct number of coins should be added. |
| Player Tries to buy item without enough coins | Item should not be given to the player, and no coins should be subtracted, and player is notified they do not have enough coins. | Ensure no coin numbers are altered and that the player does not receive the item. |

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (hrs) | Predecessor Task(s) |
| 1. Design item shop level | 2 | - |
| 2. Program random shop item generator | 2 | - |
| 3. Design and create item shop UI | 3 | 1 |
| 4. Program item shop | 5 | 1, 2 |
| 5. Plan items for item shop | 5 | 4 |
| 6. Create item sprites, animations, and attributes. | 30 | 5 |
| 7. Testing and debugging | 5 | All |

### Pert diagram



### Gantt timeline

