Name Kyle LeDoux Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

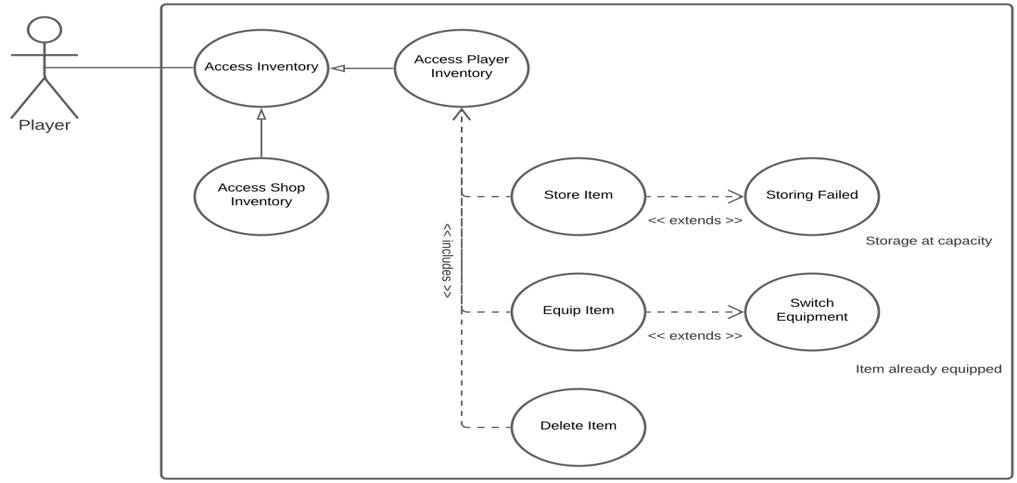
## Brief introduction \_\_/3

My feature for the Stymphalian video game is an inventory system. There are two types of inventories included in this: the player’s inventory and the shop inventory.

The player’s inventory has three functions: storing, equipping, and deleting items. The inventory will be a separate display that can be accessed using a key. The shop inventory will be a set list of items that they player may purchase using gold coins. This will be in a set location and the player may access the shop inventory when interacting with the shop NPC. Currency is also displayed in the player’s inventory, along with any collectible items the player picks up throughout their journey.

## Use case diagram with scenario \_\_14

### Use Case Diagram



### Scenarios

**Name:** Access Inventory

**Summary:** The player can access the player’s inventory or the shop’s inventory to interact with items in a variety of ways.

**Actors:** Player (Telephus)

**Preconditions:** Item class has been initialized and developed.

**Basic sequence:**

**Step 1:** The player will access the superclass inventory.

**Step 2:** Inventory will display all available items.

**Step 3:** If the player inventory is what’s being accessed, the user can choose to store, equip, or delete an item.

**Step 4:** If the shop is accessed, only perform the same characteristics the inventory superclass performs but with shop’s items.

**Exceptions:**

**Step 1:** Inventory is accessed in fight scene.

**Step 2:** Display inventory can’t be completed.

**Step 1:** Player adds item to inventory when inventory is full.

**Step 2:** Inventory asks user to cancel action or drop (delete) an item.

**Step 1:** Player tries equipping weapon when they already have a weapon equipped.

**Step 2:** Switch item from inventory with item that’s equipped.

**Post conditions:** New equipped item will now be in player’s hands, the item is now stored in inventory for player to access later, or the player no longer has an item in their inventory (dropped or deleted).

**Priority:** 2\*

**ID:** C01

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

### Data Flow Diagrams

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

### Process Descriptions

Manage Player Inventory

If item needs to be stored

Access item collectible Id

If inventory isn’t full

While( search through all inventory items )

If item is in inventory

Increase item ID’s count plus 1

Break

Add item to inventory

If item needs to be equipped

Is item already equipped (i.e., a weapon) by the player

If yes, copy id from equipped state and delete the item

Add new item to equipped, delete from inventory

Add old item to inventory using copied ID

If no, delete item from inventory and add to equipped state

If item needs to be dropped

Delete item from inventory

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

Input List:

* Add 1000 random items to inventory
* Delete 1000 items from inventory
* Add 1000 items to equipped state

Output List:

* Function performed
* Number of items in inventory after action
* Number of items per item object (display how many objects are stacked for that item)
* Result of equipped state after action

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Object (ID) | Function | # items in inventory | # items with stacks | # items in equipped | Equipped state | Result |
| 001 | Item.add( 001 ) | 1 | 0 | 0 | Empty | Success |
| 002 | Item.add( 002 ) | 2 | 0 | 0 | Empty | Success |
| 001 | Item.add( 001 ) | 3 | 1 | 0 | Empty | Success |
| 001 | Item.equip(010) | 3 | 0 | 1 | Full | Success |
| 005 | Item.remove( 005 ) | 3 | 0 | 1 | Full | Failed |
| 002 | Item.remove( 002 ) | 2 | 0 | 1 | Full | Success |
| 002 | Item.equip( 002 ) | 2 | 0 | 2 | Broken | Failed |
| … | … | … | … | … |  | … |

Continue chart until add, remove, and equip actions are completed 1000 times each

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Items List | 6 | - |
| 2. Item Struct | 6 | 1 |
| 3. Animations | 3 | - |
| 4. Artwork | 6 | - |
| 5. Inventory Display | 5 | - |
| 6. Player Inventory Actions | 15 | 2,5 |
| 7. Shop Inventory Actions | 5 | 2,5 |
| 8. Testing | 5 | 6,7 |

### Pert diagram

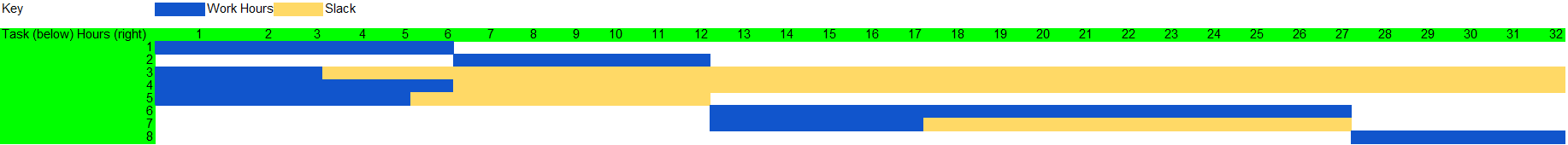
A picture containing electronics, black, keyboard

Description automatically generatedA picture containing electronics, keyboard

Description automatically generated

### Gantt timeline

(Full View)



(Split View)

Chart, bar chart

Description automatically generated

A picture containing chart

Description automatically generated