Name\_\_\_Jackson Baldwin\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

For my project, I am in charge of NPC’s (or Non-Player Characters) and Sound Design.

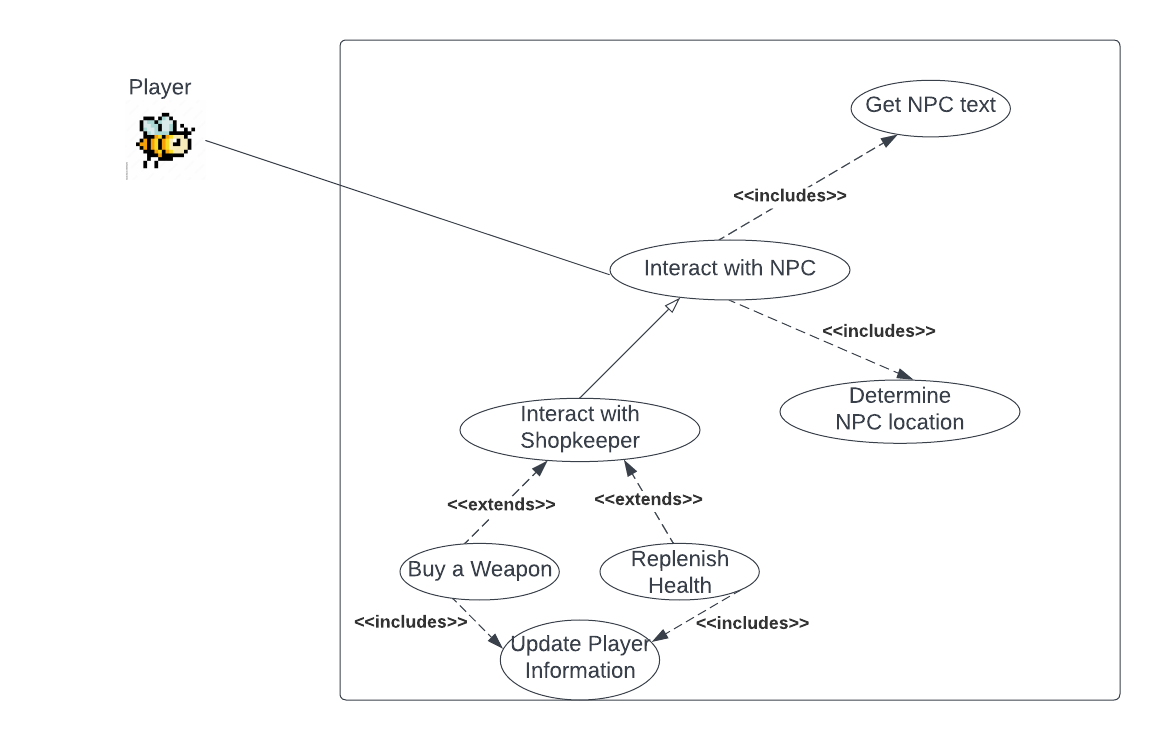
Our game will include several hostile NPC’s, but I will specifically be in charge of friendly NPC’s. This will include interactable characters such as the Queen Bee, local townspeople (bee-ple?), and most importantly, the shopkeepers. As a genre, roguelikes depend highly on the ability to buy or upgrade weapons as the game progresses.

Because the game will be procedurally generated, and due to the fact a shop can be generated at any random time along the path, I will need to be sure the shopkeepers will be able to pull random weapons from the pool of weapons available, as well replenish health for the player.

Sound design is highly dependent on work outside of coding. Other than assigning when a sound plays (player interactions, enemy attacks, boss themes, etc.) , my primary focus will be designing original music for the game, as well as organizing groups to perform the works live. I am hoping to get one of my ensembles, the Silver Saxophone Quartet, to get these pieces recorded, and give the game a unique sound. Since this consists of works outside of this class, I will be focusing more on NPC interactions for this document.

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

**Name:** Interact with NPCs

**Summary:** The user interacts with an NPC to buy a Weapon

**Actors:** Player

**Preconditions:** Shop Room has been entered

**Basic sequence:**

**Step 1:** Walk up close to the shopkeeper.

**Step 2:** Interact with the shopkeeper and read the text.

**Step 3:** Click/select “Buy a Weapon”

**Step 4:** Update the Player Information with New Weapon and respective money

**Exceptions:**

**Step 1:** “Buy a Weapon” or “Replenish Health” are selected without the necessary funds: Display “Insufficient Funds”

**Post conditions:** Player Information is Updated (money, weapons, and heath included)

**Priority:** 2\*

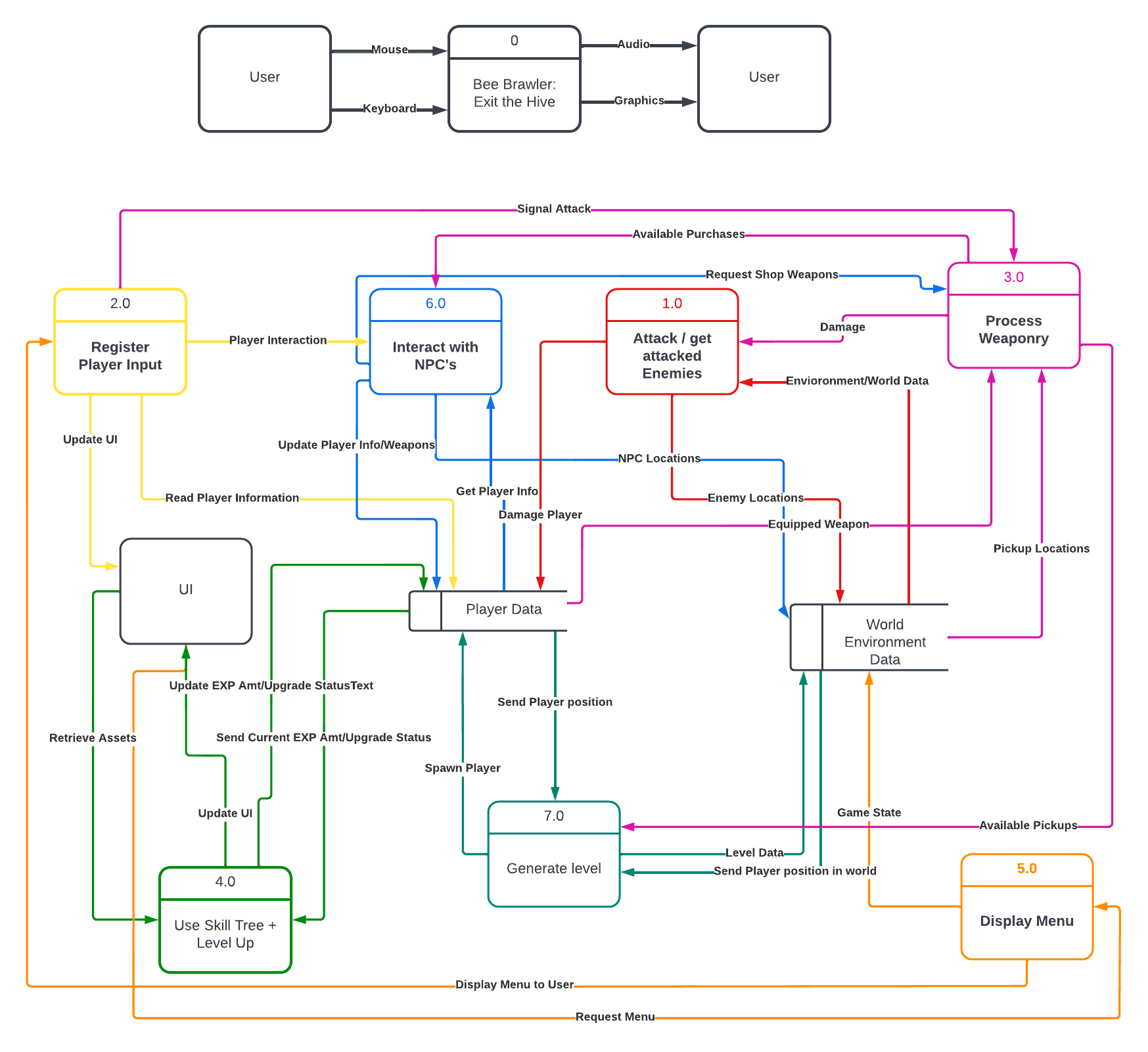
**ID:** C01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

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

### Data Flow Diagrams

Use Case Diagram and Data Flow Diagram 0



### 

### Process Descriptions

**6.1: Talk to NPC**

Get player.position;

Update Location to Player Data;

**6.2: Talk to Shopkeeper**

Get player.position;

Get player.info;

Load Shop();

Get Available Items from Weaponry();

IF player wants to buy weapon

call BuyaWeapon();

Update Location to Player Data;

ELSE IF player wants to buy health

call BuyHealth();

Update Location to Player Data;

ELSE

Update Location to Player Data;

**6.3: Buy a Weapon**

IF(player.gold >= item)

Buy Weapon;

Update player.weapons;

Update player.gold;

ELSE

Insufficient Funds Error

**6.4: Buy Health**

IF(player.gold >= item)

Buy Health;

Update player.health;

Update player.gold;

ELSE

Insufficient Funds Error

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

To test the shop and NPC capability, many tests can be performed.

I want to test the shop's capability of determining if there are any limitations to a player receiving an item. For example, a player should not be able to buy health if their health is full, and a weapon should not be able to be bought if inventory is full (this UI is controlled by another team lead, but this will be necessary for tests and a proper working system).

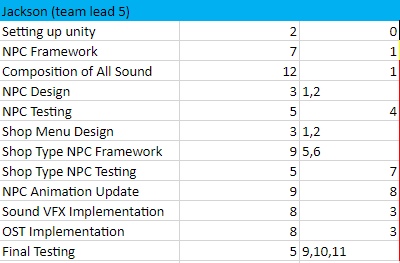
To automate this, I could simulate player info. Health would be randomly chosen between 1-10, as well as weapons 1-5 (I am not sure how many weapons you will be able to hold in this game. For the sake of the experiment, let’s assume a player can hold 5 weapons). In addition, the player would have anywhere between 1-500 gold (or honey).

The test would be simple: create a script that simulates a player buying one random health item and one random weapon item. With random health,weapons, and gold variables, this would show where the shop limits are, and what it can’t handle.

In addition, I will create a way to log the items that the shop selected from the weaponry on each test. After several rounds of automation, I could observe the data and ensure that the shop items are truly randomized.

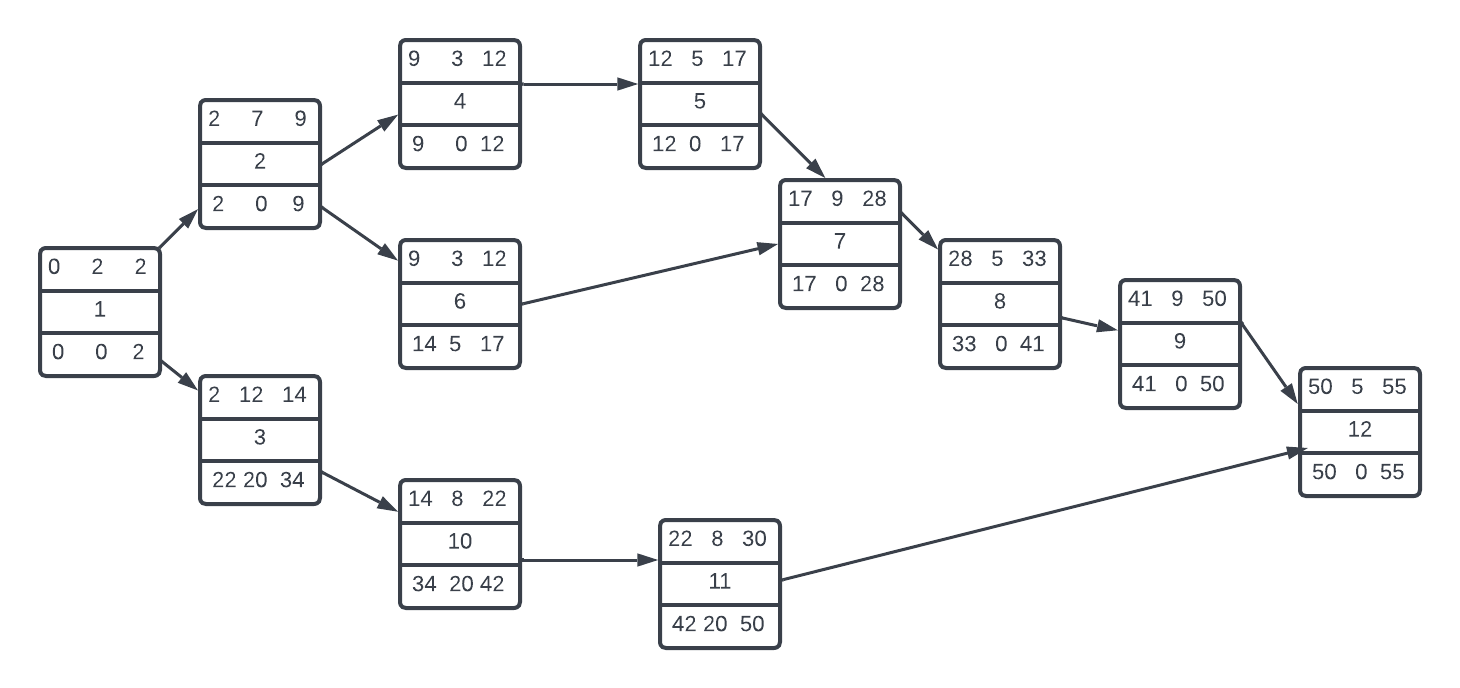
## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items



*note: I recognize that the (1) in NPC Design and Shop Menu Design is redundant. I have redone this chart 5 times and I don’t want to do it again…please don’t mark off points*

### Pert diagram



### Gantt timeline