## Brief introduction \_\_/3

My features for *Tower of Annihilation* include:

* Creating a player manager.

The Player manager keeps track of everything that has to do with the player, such as their HP, movement speed, inventory, or any sort of effects that are on the player.

* Creating a player movement script.

What the user uses to move the player in our game.

* Creating NPC / Player health bars

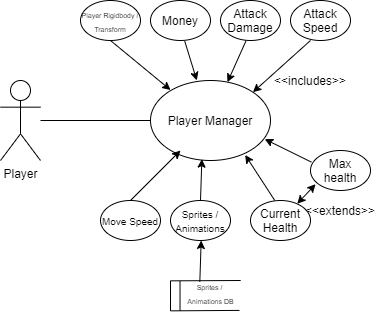
A displaying method for the NPC and Player Health bars. This is something that displays the values that are stored in their managers.

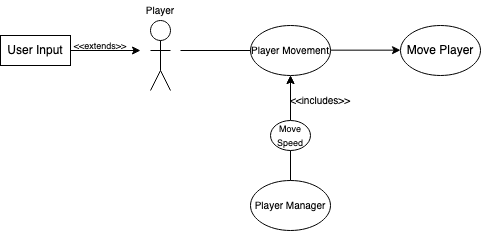
* Game design aspects (Animations / Sprites / Level Tile set / SoundFX - Music)  
  How our game is going to look and sound so it is somewhat uniform and everything goes together

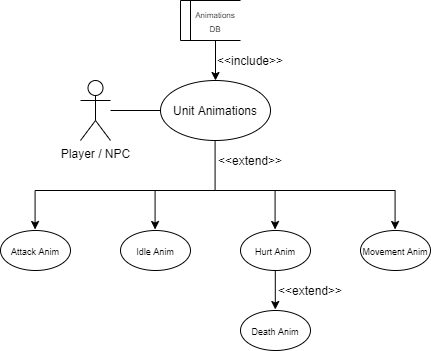
Additionally, I have taken it upon myself to make sure all of the code that we implement in to our project is up to the best practices with unity and C#. I go through most (if not all) of our code base and make small edits, to make sure everything is to how it should be (after talking with the person who implemented it.)

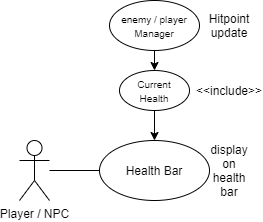
## Use case diagram with scenario \_\_/14

### Use Case Diagrams









### Scenarios:

**Name:** Player manager

**Summary:** Initializing a player gameobject and making sure it has everything that the object will need throughout the game.

**Actors:** Player

**Preconditions:** Player is initialized, exists.

**Basic sequence:**

**Step 1:** User starts instance of level, and player is loaded into game.

**Post Conditions:** The player exists in the current scene.

**Priority:** 1\*

**\***Priorities are: 1 = Must Have, 2 = Essential, 3 = Nice to have.

**Name:** Assigning movement to player objects.

**Summary:** The player movement script adds a way for the user to use various inputs (WSAD) to move the character around the scenes in our game. Using the games 2D grid and positive X and Y movements to make everything smooth and simple.

**Actors:** Player

**Preconditions:** The user inputs a valid input (or inputs).

**Basic sequence:**

**Step 1:** Grab the users input and get its correct X and/or Y value.

**Step 2:** Move the player in the desired X and/or Y direction.

**Exceptions:**

**Step 1:** Check negating inputs are negating values (W: y = 1, S: y = -1 A: x = -1, D: x = 1)

**Step 2:** Ignore both of negating values until one is one is no longer selected.

**Post Conditions:** The player moves in the desired selected input direction(s).

**Priority:** 1\*

**\***Priorities are: 1 = Must Have, 2 = Essential, 3 = Nice to have.

**Name:** Players and NPCs Animations

**Summary:** The Players and NPC pull all of their animations from the same database, they need to be running all of the correct animations when they need to be ran.

**Actors:** Player, NPCs

**Preconditions:** There is a call for an animation by the player or NPC

**Basic sequence:**

**Step 1:** Player Inputs a basic action (movement / attack)

**Step 2:** Correct animation plays for whatever animation that we may need.

**Exceptions:**

**Step 1:** The player does no inputs, and the idle animation plays forever.

**Post Conditions:** The correct animation is playing

**Priority:** 2\*

**\***Priorities are: 1 = Must Have, 2 = Essential, 3 = Nice to have.

**Name:** Player and enemy Health bars

**Summary:**  Players and enemies have current health point values in their managers, the health bars go into these managers and display the current health at real time.

**Actors:** Player, enemies

**Preconditions:** Initialization, when the player/enemy current health amount is created and a health bar is made

**Basic sequence:**

**Step 1:** Player takes damage from an enemy.

**Step 2:** The player manager current health value updates.

**Step 3:** The health bar displays the new number in the health bar.

**Exceptions:**

**Step 1:** The enemy attack is not in range.

**Step 2:** No damage is taken, the health bar does not change.

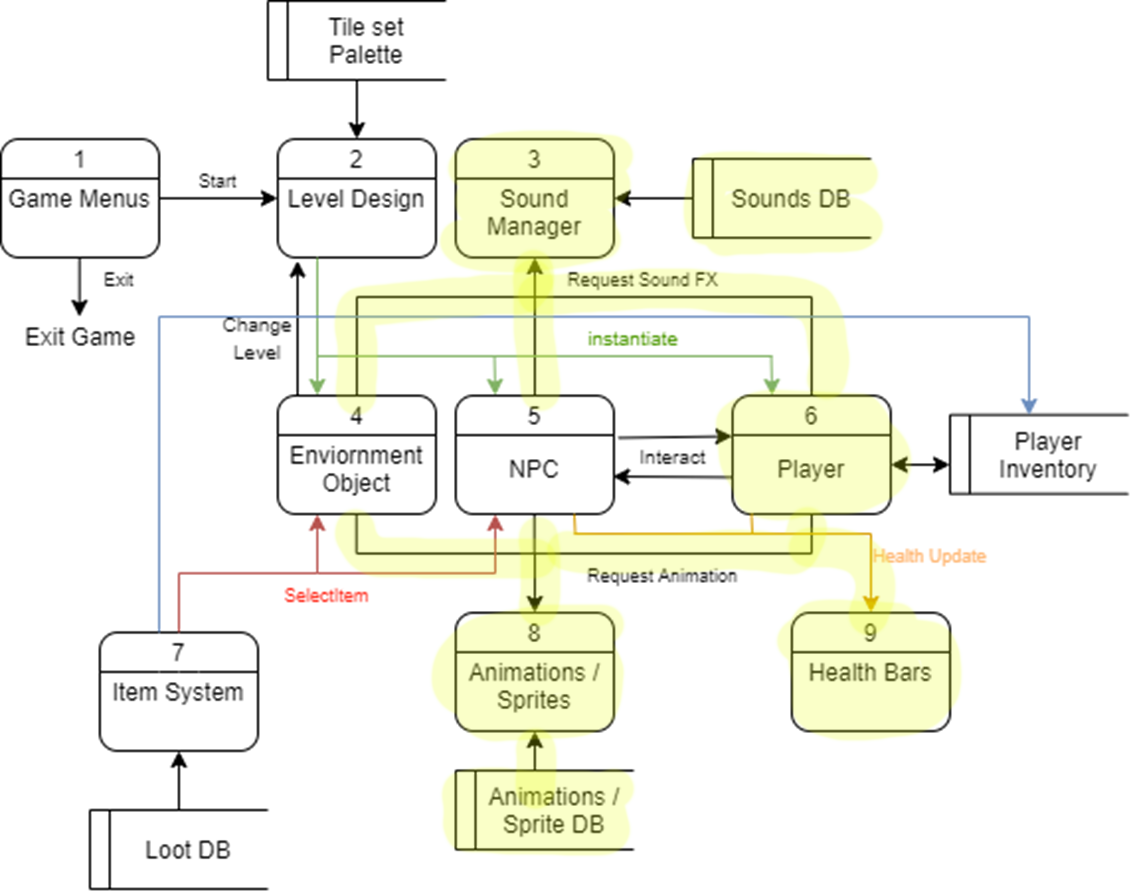
**Post Conditions:** Correct health bar amounts

**Priority:** 2\*

**\***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

### Level 0



### Data Flow Diagrams

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated Diagram

Description automatically generated

## 

### Process Descriptions

**SoundFX(3.1):**

If SoundEffectTrigger(gameobject)

Play the desired gameobject soundfx

**Music(3.2):**

Just play the game music somewhat quietly from the background.

**Player Health(6.1):**

If the player interacts with an NPC or object, by either getting attacked by an enemy or buying a potion from a shopkeeper. The players hitpoint value can change, and the changes are kept track in the player manager.

**Player Movement(6.4):**

If the user inputs a movement key, the player will move in the desired direction based on the key(s) that were pressed.

**Player Animations(8.1):**

**NPC Animations(8.2):**

Depending on what animations are needed there are certain triggers that activate when something in particular happens. If the player takes damage there is a hurt trigger that is ran in the animator that plays the hurt animation and then continues back to the idle, as long as there is more than 0 hp, or else it plays the death animation

**Player Health Bar(9.1):**

**NPC Health Bar(9.2)**

Displays the NPC or player current health in the enemy or player manager with the current amount.

## Acceptance Tests \_\_\_/9

**Player Movements:**

Have an absurd amount of movements inputted into a bot and make sure that the player follows the inputs correctly.

If we feed the bot plenty of inputs that the player can follow, the output should be:

* Where the players transform position is.

**Player Manager / Health Bars:**

Create 1000 instances of a player and make sure they have unique values, even if one of the players is being attacked.

The output should be like the following:

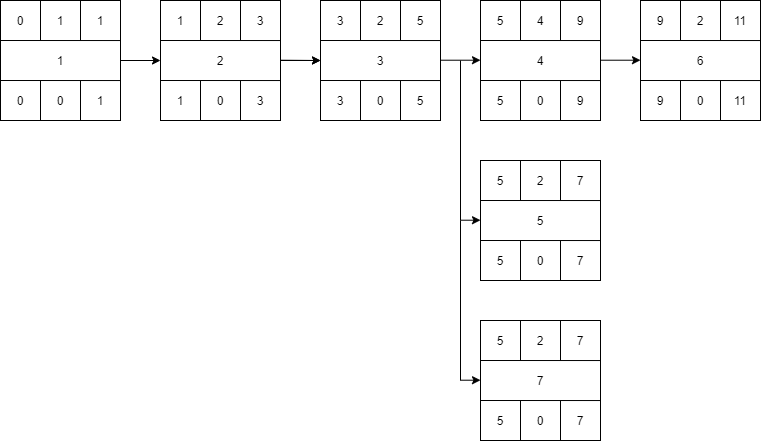
One of the 1000 players should have their hp value changed and none of the others should.

## Timeline \_\_\_/10

### Work Diagrams

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Repo + Git Setup | 1 | - |
| 2. Unity Project setup | 2 | 1 |
| 3. Player Manager | 2 | 2 |
| 4. Sprites / Animations / Tileset | 4 | 3 |
| 5. Sound Manager | 2 | 3 |
| 6. Coins / Power-up Integration | 2 | 4 |
| 7. Implement Health Bars | 2 | 3 |

### Pert Diagram



### Gantt Diagram

