Name\_\_\_\_\_\_\_Jacob Alderink\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

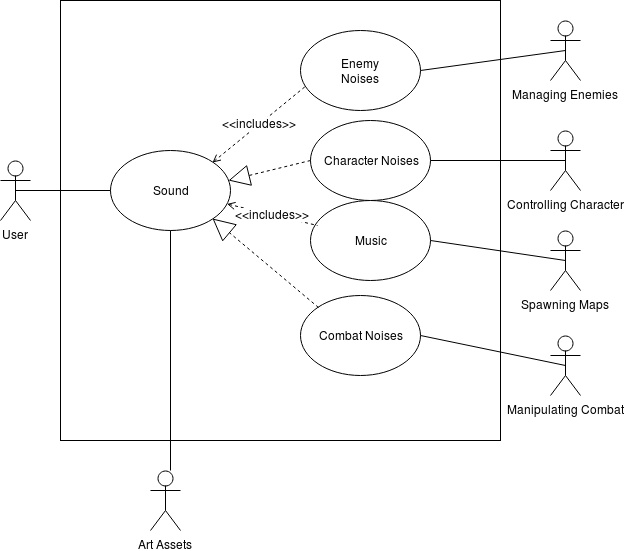
[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

My feature is sound design in the game. I will champion this and my part of the game will be focused on Creating and maintaining sound.

## Use case diagram with scenario \_\_14

### Use Case Diagram



### Scenarios

**[You will need a scenario for each use case]**

**Name:** Sound Creation

**Summary:** The user receives sound from various noises in the game

**Actors:** User

**Preconditions:** Game has been started

**Basic sequence:**

**Step 1:** Recieve character noise.

**Step 2:** Recieve music input.

**Step 3:** Receive Enemy noise input.

**Step 4:** Receive Combat noise input.

**Step 5:** Concatenate the sound to the user.

**Exceptions:**

**Step 1:** No enemies are present: Don’t produce enemy noise.

**Step 2:** No combat is happening: Don't produce combat sounds.

**Post conditions:** Sound is produced to the user

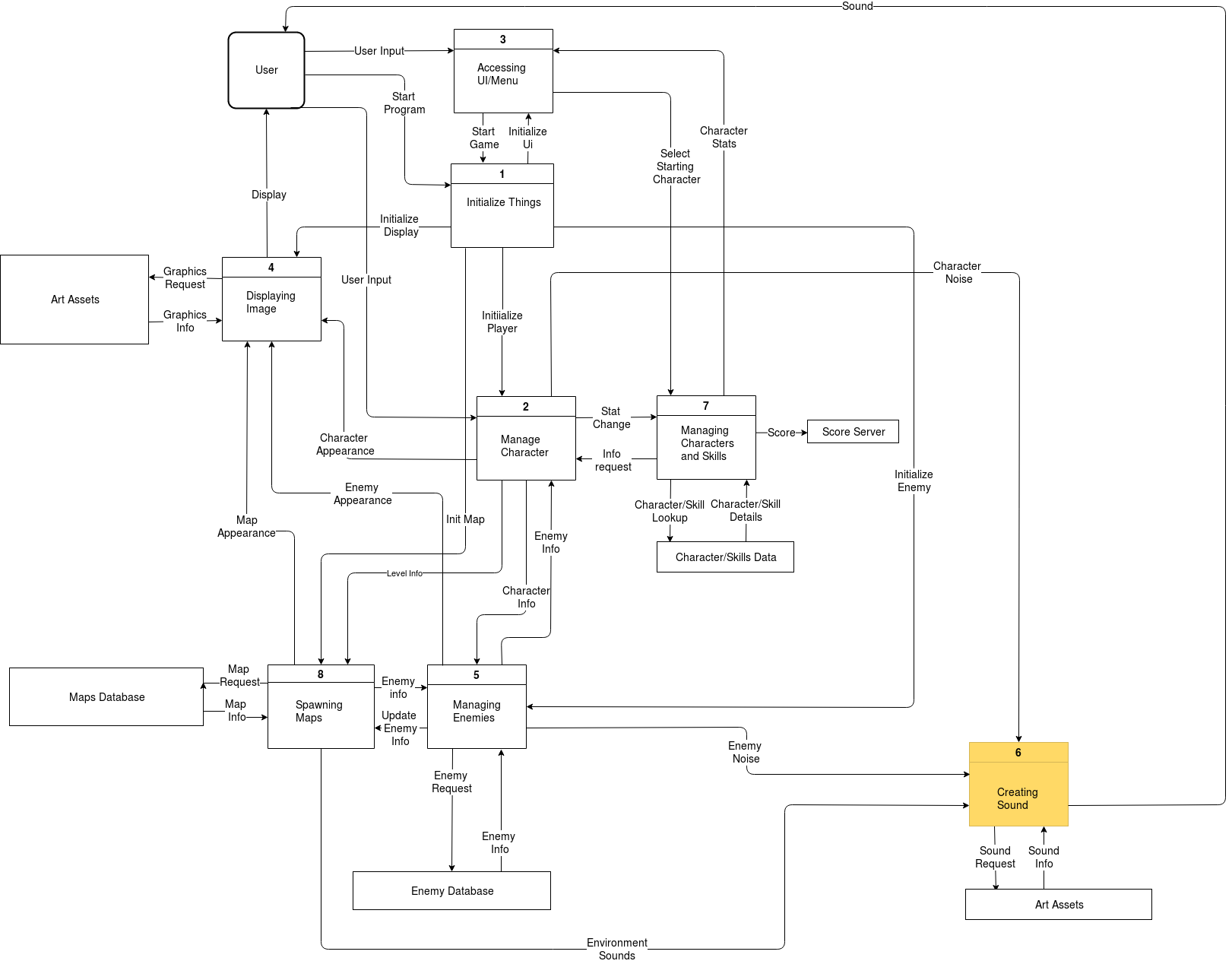
**Priority:** 2

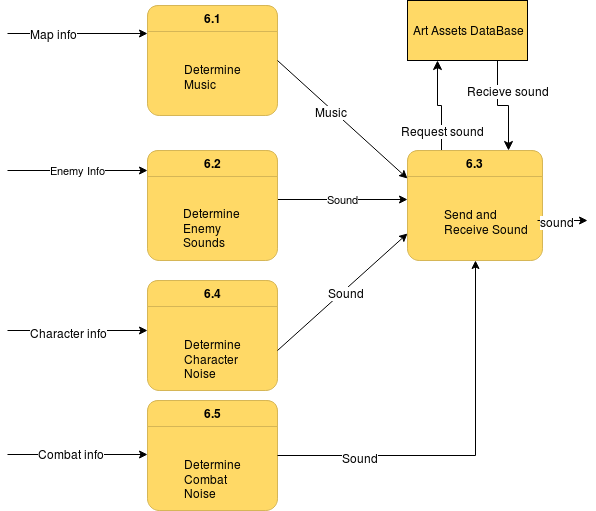
**ID:** 6

\*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





### Process Descriptions

Determine music:

Receive map info. Based on level that is currently being played, play that specific song.

Determine Character Noise:

Receive Character info, based on his location, his health, and other

aspects play certain noises from the character that are dependant on

these things.

Determine Combat Noise:

Receive information regarding the characters location, and the enemies

location. As well as who was hurt and how were they hurt as well as any

combat related noises and determine the different sounds this.

Determine Enemy Noise:

Receive information from enemy regarding location as health and other

info, and determine from there the type of noise to be made.

Send and Receive Sounds:

Receive every noise input from the previous processes and and send

them each to the user through his speakers, or audio device after

retrieving them from the database.

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

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

**Enemy Noise Feature:**

Run feature for each type of enemy in the game, as well as at different healths. Test the audio to make sure that the noise they make is spaced out over time, and is actually the correct noise that the enemy is making.

**Character Noise Feature:**

Run feature for each type of character in the game, as well as at different healths and different locations in the game. Test the audio to make sure that the noise they make is spaced out over time, and is actually the correct noise that the character is making.

**Combat Noise Feature:**

Run feature for each combat scenario in the game. First have the character fighting with his weapon. Make sure the weapon being fired or “activated” is making the correct noise. Next check to see that when it hits or misses the two different noises are made. Next check to make sure the the enemy for each type of their possible weapons/ abilities is also making the correct noise/sound.

**Music Feature:**

Check that for each map type that we decide to add into the game, that the correct music type that was chosen is playing.

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

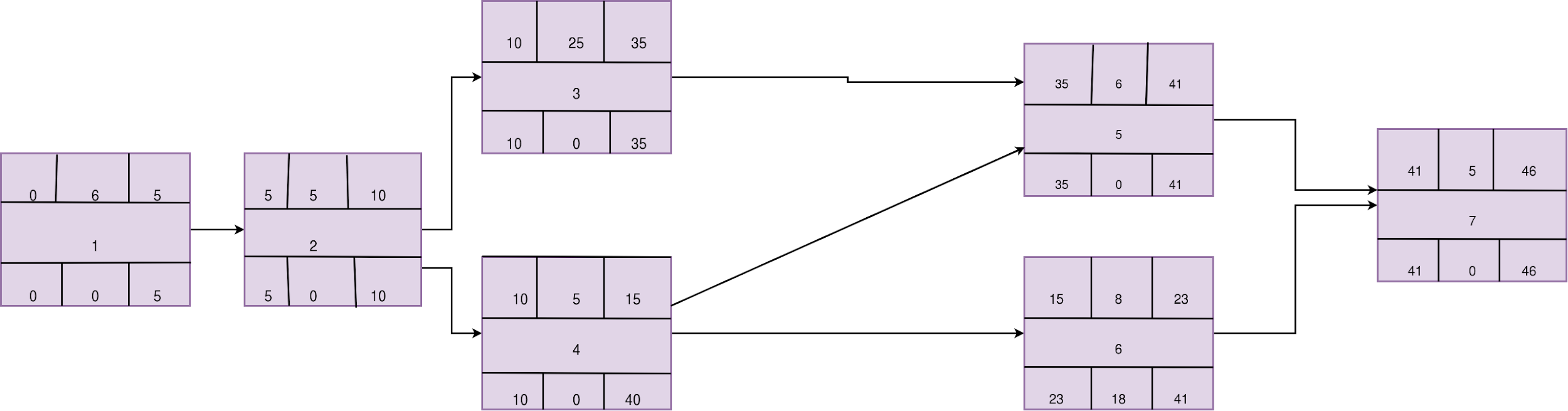
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Requirements Collection | 6 | - |
| 2. Sound Design | 5 | 1 |
| 3. Sound Recording | 25 | 2 |
| 4. Database Construction | 5 | 2 |
| 5. User Documentation | 6 | 3,4 |
| 6. Programming | 8 | 4 |
| 7. Testing | 5 | 6,5 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |