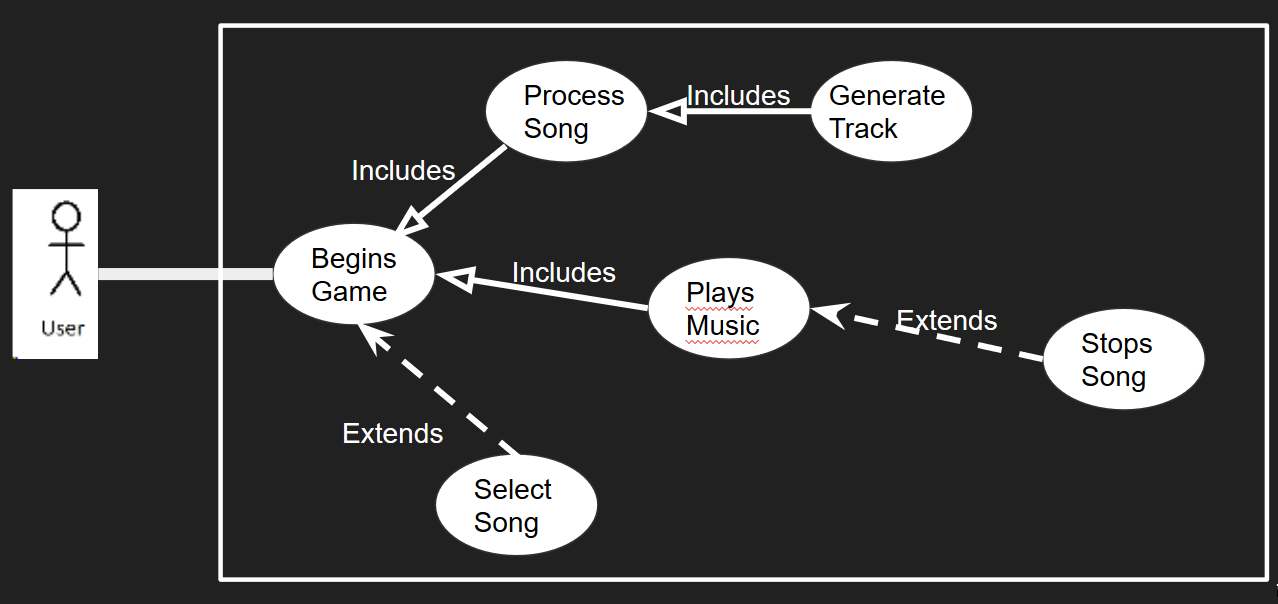
Name: Jaidin Medina Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

The Sound Manager is responsible for processing the song selected by the user, and passing off the processed song so that a track can be generated. It is also responsible for playing music while the game is running, and ending it when an obstacle is hit, or the song ends.

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

### Use Case Diagrams



### Scenarios

**Name:** Begins Game

**Summary:** The user begins the game, which plays music until the game is over

**Actors:** User (Player)

**Preconditions:** Game has been started

**Basic sequence:**

**Step 1:** Game Begins

**Step 2:** Processes song

**Step 3:** Plays music until song ends

**Exceptions:**

**Step 1:** Song can be changed from default. In this case, the processed song is the

chosen one

**Step 3:** An obstacle is hit: song is stopped

**Post conditions:** None

**Priority:** 1

**ID:** C05

## Data Flow diagram from Level 0 to process description for your feature \_\_\_\_\_/14

### Data Flow Diagrams

### 

Song List

5.2

Play Song

Chosen Song

5.3

Stop Song

5.1

Process Song

Music Manager

### Process Descriptions

Process Song:

Take in a song

Store timestamps of beats within the song within an array

IF difficulty is easy:

WHILE any pair of consecutive timestamps are within 0.5 seconds, delete the second one.

IF difficulty is hard:

WHILE any pair of consecutive timestamps are within 0.25 seconds, delete the second one.

Return timestamp array

//This method will be used by object manager to generate the track

Play Song:

Take in a song

Begin playing song

IF song ends OR obstacle hit:

// Will check time elapsed against song length, and will receive obstacle // hit from Obstacle Manager

Call Stop Song

Stop Song:

End playing current song

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

**Process Song**

A test suite of 3 songs will be processed, each of which will be also processed by hand. The two arrays will be compared, and checked for any discrepancies.

A test suite of 100 songs will be processed for easy, and 100 for hard. The timestamps will be output to a file, and checked to ensure that for easy no timestamps are within 0.5 seconds, and for hard, no timestamps are within 0.25 seconds. The timestamps will also be checked to ensure that they are in sequential order (they will be sorted and the sorted array will be compared to ensure it matches the unsorted).

**Play Song/Stop Song**

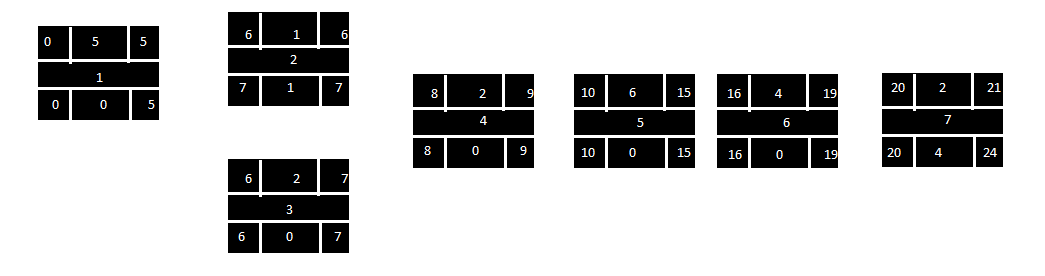
A test suite of 10 songs will be ran through the game, forcing an obstacle collision to occur at specific times, ensuring that the song ends at that point. In addition, a run with no obstacle collisions will be repeated 5 times, checking at random intervals during the song to ensure it is still playing. In addition, 5 songs of various lengths will be run to completion, to ensure that the song stops at the end of its duration.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 5 | - |
| 2. Output/Input Design (Data Transfer) | 1 | 1 |
| 3. User Documentation | 2 | 1 |
| 4. File Read/Write | 2 | 2, 3 |
| 5. Programming | 6 | 4 |
| 6. Testing | 4 | 5 |
| 7. Finalizing Production | 2 | 6 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |