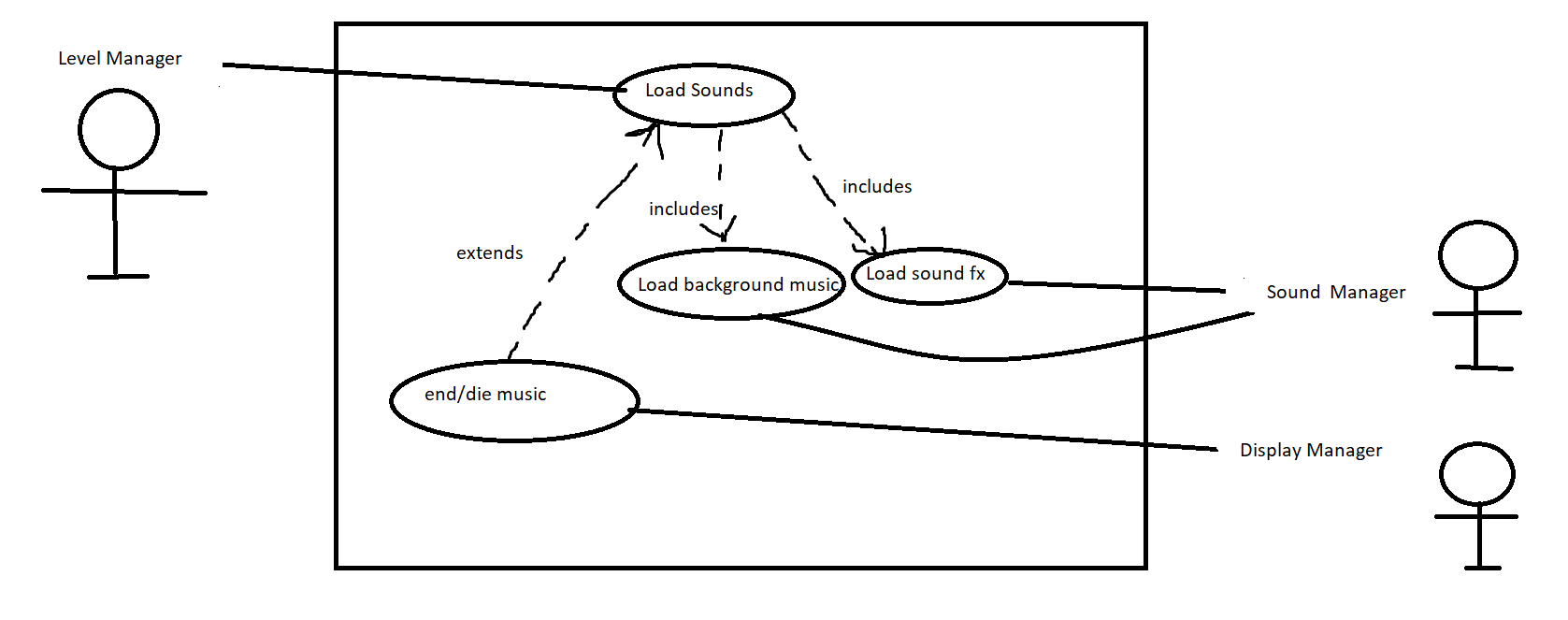
Name: Matthew Mills Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My job is to create a use case that load sound files into a Sound manager object when it is triggered by a level manger object.

## Use case diagram with scenario \_\_14



### Scenarios

**Name:** Load Sound FX

**Summary:** The level manager retrieves audio resource files and places them in the sound manager.

**Actors:** Level Manager, Sound Manager

**Preconditions:** The game has been set up and the Level Manager have been instantiated

**Basic sequence:**

**Step 1:** The level manager creates a level.

**Step 2:** It loads the necessary sounds from audio files and places them in the sound manager.

**Exceptions:**

None, there is no user input, so this process will happen the same way every time the game runs.

**Post conditions:** The sound manager contains the sound fx that will be accessed by the game objects.

**Priority:** 3

**ID:** 3.1

**Name:** Load Background

**Summary:** The level manager retrieves audio resource files and places them in the sound manager.

**Actors:** Level Manager, Sound Manager

**Preconditions:** The game has been set up and the Level Manager has been instantiated

**Basic sequence:**

**Step 1:** The level manager creates a level.

**Step 2:** It loads the necessary sounds from audio files and places them in the sound manager.

**Exceptions:**

None, there is no user input, so this process will happen the same way every time the game runs.

**Post conditions:** The sound manager contains the sound fx that will be accessed by the game objects.

**Priority:** 3

**ID:** 3.1

**Name:** End/Die

**Summary:** The level manager retrieves audio resource files and places them in the display manager.

**Actors:** Level Manager, Display Manager

**Preconditions:** The game has been set up and the Level Manager has been notified that the player stopped playing or died.

**Basic sequence:**

**Step 1:** The level manager creates a level.

**Step 2:** It loads the necessary sounds from audio files and places them in the display manager.

**Exceptions:**

None, there is no user input, so this process will happen the same way every time the game runs.

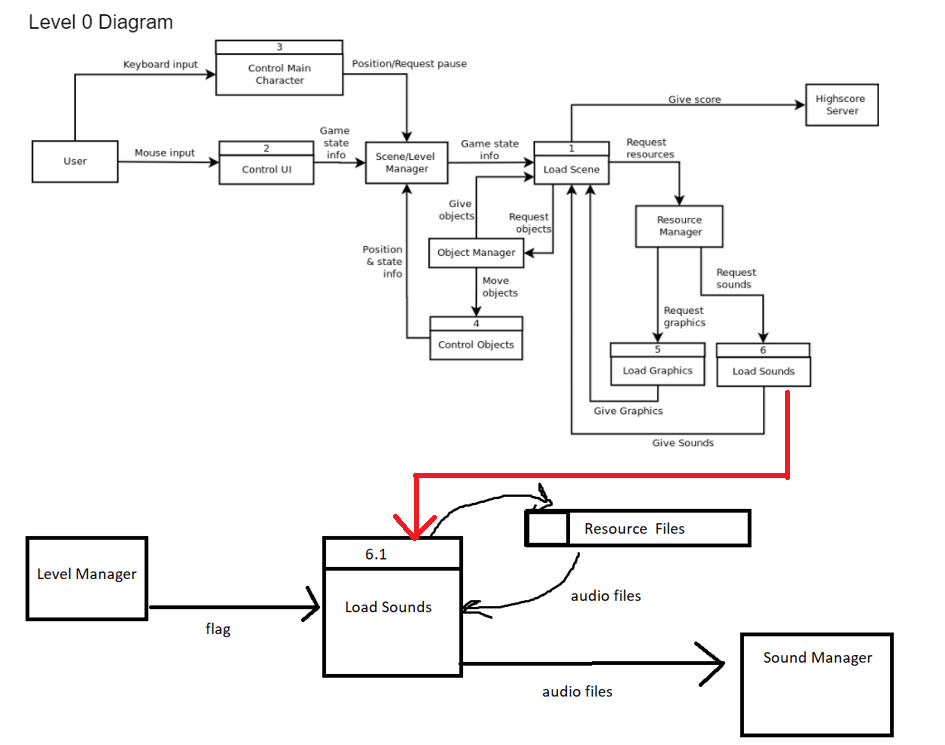
**Post conditions:** The Display Manager contains the end/die music that will it will play.

**Priority:** 3

**ID:** 3.1

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

### Data Flow Diagrams



### Process Descriptions

Load Sounds:

The level manager opens the resource files

It then loads the sound files into the Sound Manager

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

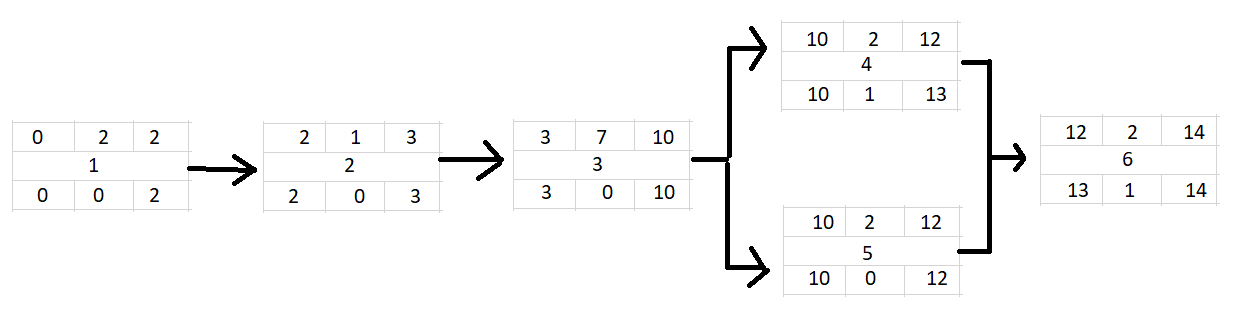
* **The method in the Level manager that retrieves audio files will be tested by being called using every possible file path to the resource files**
* **The maximum size of audio file that can be stored in the Sound Manager object will be investigated and discovered.**

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 2 | - |
| 2. Report Design | 1 | 1 |
| 3. User Documentation | 2 | 5 |
| 4. Programming | 5 | 1,2 |
| 5. Testing | 2 | 1,2 |
| 6. Installation | 2 | 5 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |